

Kerkko Maukonen

**KOHTI LEANIN MUKAISTA
OHJELMISTOTUOTANTOA**



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2015

TIIVISTELMÄ

Maukonen, Kerkko
Kohti Leanin mukaista ohjelmistotuotantoa
Jyväskylä: Jyväskylän yliopisto, 2015, 86 s.
Tietojärjestelmätiede, pro gradu -tutkielma
Ohjaaja: Leppänen, Mauri

Kilpailu globaaleilla markkinoilla ja Japanin toisen maailmansodan jälkeinen talouskriisi johtivat Toyotalla täysin uudenlaisen tuotantomenetelmän kehittämiseen. Tämä Toyotan tuotantojärjestelmä (TPS), joka sittemmin on tullut paremmin tunnetuksi Lean-ajatteluna, on otettu käyttöön useilla aloilla. Viime vuosina Lean-periaatteita on alettu soveltamaan myös ohjelmistotuotannossa. Kirjallisuudessa esiintyy kuitenkin erilaisia näkemyksiä siitä, mitä Lean-ohjelmistotuotannolla tarkoitetaan, millä tavalla Lean-periaatteita tulisi soveltaa ja miten Leanin mukaista toimintaa voidaan arvioida.

Tämän kirjallisuuskatsauksen tarkoituksena on esitellä teollisuustuotannon ja ohjelmistotuotannon näkökulmista Lean-ajattelun taustaa, pääperiaatteita, Lean-transformaatiomalleja ja -ohjeistoja sekä Leanin mukaisen toiminnan arviointia.

Tutkimustuloksina todetaan, että Lean on perinteisen teollisuuden ohella todettu hyväksi myös ohjelmistotuotannossa. Leanin avulla on mahdollista pienentää prosessien läpimenoaikoja ja todettua hukkaa. Näiden seurauksena yrityksen tuotantokustannukset laskevat ja tehokkuus kasvaa. Koska yrityksen toimintatapojen ja prosessien muuttaminen voi aiheuttaa yrityksessä huolta ja epätietoisuuteen liittyviä ongelmia, on Lean-transformaation tueksi kehitetty malleja sekä arviointityökaluja transformaation ohjaamiseksi ja arvioimiseksi. Tutkimuksessa esitellään tunnetuimpia tapoja Leanin mukaisen toiminnan ja Leanin mukaisen ohjelmistotuotannon arviointiin.

Asiasanat: Lean, Lean-ajattelu, ohjelmistotuotanto, ohjelmistokehitys, Kanban, Lean-transformaatio, Lean-arviointi

ABSTRACT

Maukonen, Kerkko
Towards Lean Software Engineering
Jyväskylä: University of Jyväskylä, 2015, 86 p.
Information Systems Science, Master's Thesis
Supervisor: Leppänen, Mauri

Competition on the global markets and the financial crisis in Japan after the Second World War led to a situation where Toyota needed a totally new production system in order to avoid bankruptcy. This Toyota Production System (TPS) later known as Lean thinking has been implemented to several different types of industry. In recent years, Lean principles have been applied in software engineering as well. There are various conceptions in the literature about what Lean software engineering actually means, how to apply Lean principles in software engineering, and how to assess Lean software engineering.

This literature review describes the background of Lean thinking, its main principles, Lean transformation and assessment from the viewpoints of industrial and software engineering.

This study shows that Lean can accelerate production in industrial production as well as in software engineering. By applying Lean it is possible to shorten the cycle times of the production processes and lower the amount of waste. These two lead to lower production costs and better performance. Changing the processes and the way of working in the company might cause some concerns and problems. To avoid these, some frameworks and guidelines have been developed to support Lean transformation and the assessment of Lean software engineering. This research describes some of the most well-known frameworks and ways to perform the assessments in Lean production and Lean software engineering.

Keywords: Lean, Lean-thinking, software engineering, software development, Kanban, Lean transformation, Lean assessment

KUVIOT

KUVIO 1 Lean-transformaatiomalli.....	17
KUVIO 2 Kanban-taulu	31
KUVIO 3 Kumulatiivinen virtauskaavio	32
KUVIO 4 Lean-transformaatiomalli ohjelmistotuotantoon.....	36
KUVIO 5 Lean-organisaation prosessiarkkitehtuuri.....	47
KUVIO 6 LESAT for software -mallin arviointikohtien määrät	56
KUVIO 7 Lean Amplifier -mallin arviointikohtien määrät	57
KUVIO 8 Arvovirtakartoituksen notaatio.....	64

TAULUKOT

TAULUKKO 1 Yhteenveto viidestä tutkitusta Lean-kyselytutkimuksesta	44
TAULUKKO 2 Yhteenveto seitsemästä tutkitusta teollisuuden Lean-arviointityökalusta	45

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT

TAULUKOT

1	JOHDANTO	7
2	LEAN-AJATTELU	10
2.1	Leanin syntyhistoria	10
2.2	Periaatteet	11
2.2.1	Viisi avainperiaatetta	11
2.2.2	Lisäarvoa tuottamaton hukka	12
2.2.3	Kanban	14
2.3	Transformaatio	15
2.3.1	Transformaatiohaasteet	15
2.3.2	Transformaatiomalleja	16
2.4	Yhteenveto	21
3	LEAN OHJELMISTOTUOTANNOSSA	23
3.1	Tausta	23
3.2	Lean-periaatteita	25
3.3	Kanban Leanin mukaisessa ohjelmistokehityksessä	28
3.3.1	Kanban-periaatteita	28
3.3.2	Kanbanin hyödyt ohjelmistotuotannossa	33
3.3.3	Kanbanin haasteet ohjelmistokehityksessä	34
3.4	Lean-transformaatio ohjelmistotuotannossa	35
3.5	Leanin hyötyjä	38
3.6	Leanin kustannuksia	39
3.7	Yhteenveto	40
4	LEANIN MUKAISEN TOIMINNAN ARVIOINTI	42
4.1	Tausta	42
4.2	Lean-organisaation arviointia koskevia tutkimuksia	43
4.3	LESAT-työkalu	46
4.3.1	Rakenne	46
4.3.2	LESAT-arviointiprosessi fasilitoijan näkökulmasta	48
4.3.3	Yleisimmät virheet LESAT:in käytössä	51
4.4	Yhteenveto	52
5	LEAN-OHJELMISTOTUOTANNON ARVIOINTI	53

5.1	LESAT ohjelmistotuotannossa	53
5.1.1	LESAT for Software	54
5.1.2	LESAT for Softwaren vertailu muihin arviointimenetelmiin	55
5.1.3	LESAT for softwaren soveltuvuus Lean-ohjelmistotuotannon arviointiin	58
5.2	Lean-ohjelmistokehityksen arviointi.....	59
5.2.1	SPI-LEAM.....	59
5.2.2	Muita arvioinnin mittareita	62
	Jätteen minimointi	62
	Virtauksen nopeus.....	62
	Arvonmuodostus	63
	Koodirivien määrä.....	65
	Virheiden tunnistus ja seuranta.....	65
	Leanin mukaisen ohjelmistojen ylläpidon arviointi	67
5.3	Yhteenveto.....	67
6	POHDINTA	69
6.1	Lean teollisuudessa	69
6.2	Lean-ohjelmistotuotanto	71
6.3	Lean-ohjelmistotuotannon arviointi.....	74
7	YHTEENVETO.....	77
	LÄHTEET	81

1 JOHDANTO

Ohjelmistotuotanto on ollut viime vuosina paljon otsikoissa, useimmiten epäonnistuneiden ja budjetit reilusti ylittäneiden projektien takia. Ala on moniin muihin, etenkin teollisuustuotantoon verrattuna varsin nuori ja nopeasti kehittynyt. 1990-luvun puolivälin jälkeen alkoi ilmaantua uudenlaisia kehitysmenetelmiä kuten Scrum (Schwaber, 1995) ja XP (Beck, 1999), jotka sittemmin 2000-luvun alussa tulivat yleisesti paremmin tunnetuiksi ketterinä kehittämismenetelminä (Beck ym., 2001). Ketterät menetelmät keskittyvät pääasiassa vain tiimi- ja projektitasolle (Salo, 2006, s.24). Parannukset tuotannon tehokkuudessa ja laadussa edellyttävät kuitenkin ohjelmistoyrityksen toiminnan kattavampaa huomioinnin ottamista.

Lean on teollisuudessa vuosikymmeniä käytössä ollut tuotantomenetelmä, joka on saanut alkunsa jo 1940-luvun autoteollisuudesta (Ohno, 1988, s. 3; Womack, Jones & Roos, 1990; Womack & Jones, 2003). Leania on tämän jälkeen sovellettu onnistuneesti eri alojen teollisuustuotannossa sekä palvelualoilla. Leanin perimmäisenä ajatuksena on tuottaa mahdollisimman vähällä mahdollisimman paljon. Vähyys ilmenee esimerkiksi resursseissa, varastossa, työntekijöiden määrässä ja tuotantoon tarvittavassa ajassa. (Womack & Jones, 2003, s. 15.) Lean perustuu viiteen avainperiaatteeseen, joita ovat arvo, arvovirta, virtaus, imu ja täydellisyys (Womack ym., 1990; Womack & Jones, 1996).

Siirtyminen perinteisestä toimintatavasta Leanin mukaiseen toimintaan ei ole kuitenkaan yksinkertaista (Bozdogan ym., 2000; Liker, 2004; Kuusela ym., 2011). Liker (2004, s. 7) mukaan yleisiä syitä Lean-transformaation epäonnistumiselle ovat muun muassa liikaa pelkkiin prosessityökaluihin keskittyminen, liian pintapuolinen implementointi ja ylimmän johdon passiivinen osallistuminen päivittäisiin operaatioihin ja jatkuvaan parantamiseen. Teollisuutta varten on kehitetty useampia transformaatiomalleja (Bozdogan ym., 2000; Liker, 2004) siirtymisen tukemiseksi. Bozdoganin ym. (2000) malli sisältää kolmesta syklistä koostuvan organisaatiotason yleisluontoisen kuvauksen vaiheista, jotka panevat alulle, ylläpitävät ja jatkavasti parantavat organisaation transformaatiota Lean-periaatteiden ja -käytänteiden mukaiseksi. Likerin (2004) malli on enemmän omaa ajattelua korostava ohjeisto joka painottaa enemmän omaa ajattelua,

omaa oppimista, Lean-filosofiaa ja oman tavan kehittämistä Toyotan ideoiden ja peruseriaatteiden päälle kertomatta suoraan mitä tulee tehdä. Osin näistä johdettuina on esitetty malleja ja ohjeistoja myös ohjelmistotuotantoa harjoittavien yritysten transformaatiota tukemaan. Esimerkkinä tällaisista lähteistä ovat Kuusela & Kouvuluoma (2011) ja Middleton & Sutton (2005).

Siirryttäessä kohti Leanin mukaista toimintaa on arvioinnin todettu olevan olennaista (Kuusela & Kouvuluoma, 2011; Karvonen ym., 2012). Leanin mukaisen toiminnan ja Lean-ohjelmistotuotannon arvioinnin on koettu kuitenkin olevan haasteellista (Jonsson ym., 2013; Karvonen ym., 2012). Jonssonin ym. (2013) mukaan Lean-ohjelmistotuotannon arviointi on haasteellista, sillä vaikka Leania on käytetty ohjelmistotuotannossa, niin Lean-ohjelmistotuotanto on vielä ilman tarkkaa määritelmää. Ilman tarkkaa määritelmää on arviointi hankala suorittaa ja verrata arviointimenetelmiä toisiinsa (Jonsson, ym. 2013). Arviointia helpottamaan on kehitetty erilaisia malleja ja työkaluja (MIT, 2001; LAI, 2012; Karvonen ym., 2012), jotka auttavat organisaatioita havainnoimaan paremmin nykytilaansa ja ohjaamaan toimintaa kohti haluttua tulevaisuuden tilaa.

Lean-ajattelusta ja sen soveltamisesta teollisuustuotantoon on olemassa hyvin paljon tutkimusta. Sen sijaan Leanin sovittamisesta ohjelmistotuotantoon tutkimuksia on huomattavasti vähemmän. Tyypillistä näille tutkimuksille on, että ne tarkastelevat aihetta usein vain tietyistä näkökulmista, jolloin tuloksena on jossain määrin pirstaleinen kokonaiskuva.

Tämän tutkielman tarkoituksena on muodostaa tiivis ja yhtenäinen esitys siitä, mitä Lean-ajattelulla tarkoitetaan teollisuudessa ja ohjelmistotuotannossa sekä mitä on Lean-ohjelmistotuotanto, miten transformointi Lean-periaatteiden mukaiseen ohjelmistotuotantoon voidaan toteuttaa ja kuinka Leanin mukaista toimintaa ja Lean-ohjelmistotuotantoa voidaan arvioida.

Tutkielma pyrkii vastaamaan seuraavaan tutkimusongelmaan: *Miten Lean-ajattelua voidaan soveltaa ohjelmistotuotannossa?* Tutkimusongelma on jaettu seuraaviin tutkimuskysymyksiin:

- Mikä on Lean-ajattelun tausta ja sisältö?
- Mitä tarkoitetaan Lean-ajattelulla ja -periaatteilla ohjelmistotuotannossa?
- Millä tavoin voidaan tukea Leanin mukaiseen toimintaan siirtymistä?
- Miten Leanin mukaista ohjelmistotuotantoa voidaan arvioida?

Tässä työssä tarkastellaan ohjelmistotuotantoa useammalla tasolla. Jotta voidaan tehdä selväksi, minkä tasoista asioista on kulloinkin kysymys, määritellään seuraavaksi termit ohjelmistotuotanto ja ohjelmistokehitys. *Ohjelmistotuotannolla* (software engineering) tarkoitetaan tässä tutkimuksessa kaikkea sellaista toimintaa, joka kattaa ohjelmistojen kehittämisen, laadunvarmistuksen ja prosessinhallinnan. Tähän kuuluvat myös ohjelmistoja tuottavan organisaation kulttuuri ja johtaminen kaikilla organisaatiotasoilla. Ohjelmistoja kehitetään projekteissa. Ohjelmistotuotanto sisältää useita projekteja, joita laajassa organisaatiossa toteutetaan useita samanaikaisesti. *Ohjelmistokehityksellä* (software development) tarkoitetaan yksittäisen projektin yhteydessä tapahtuvaa ohjelmis-

tojen tekemistä. Tällöin keskiössä ovat projektinhallinta sekä ohjelmistokehityksen menetelmät ja käytänteet. Tätä jakoa käytetään jatkossa puhuttaessa Lean-ajattelun soveltamisesta ohjelmistotuotantoon.

Tutkielma on käsitteellis-teoreettinen, ja se toteutetaan kirjallisuuskatsauksena nojautuen aihetta käsittelevään tutkimukseen. Lähdemateriaalissa pyritään kiinnittämään erityistä huomiota julkaisun tunnettavuuteen, julkaisukanavan tasokkuuteen ja lähteiden ajantasaisuuteen. Tässä tutkielmassa on käytetty lähdemateriaalina aineistoa vuosilta 1985 - 2014 aiheeseen liittyvistä artikkeleista ja kirjallisuudesta. Lähdemateriaali on kerätty pääosin sähköisistä tietokannoista kuten IEEE-Xplore, ACM Digital Library, SciVerse ja Google Scholar. Hakusanoina lähteiden hankinnassa on käytetty muiden muassa: "Lean production", "Lean thinking", "Lean software development", "Lean assessment" ja "Lean software assessment". Näiden lisäksi tietoa on kerätty myös saatavilla olevista kirjoista.

Tutkimustulokset on pyritty saamaan siihen muotoon, että niitä voisi hyödyntää kuka tahansa Lean-ajattelun periaatteista ja Lean-ajattelun mukaisesta ohjelmistotuotannosta, Lean-transformaatiosta sekä Leanin mukaisen toiminnan ja Leanin mukaisen ohjelmistotuotannon arvioinnista kiinnostunut.

Tutkielma koostuu johdannon ja yhteenvedon lisäksi viidestä luvusta. Luvussa 2 tarkastellaan Lean-ajattelun syntyhistoriaa, periaatteita ja liiketoiminnan transformointia Leanin mukaiseksi teollisuuden näkökulmasta. Jotta Leanin taustasta ja perusteista saataisiin riittävän hyvä kuva, tarkastellaan sitä aluksi perinteisen teollisuuden näkökulmasta syntyhistorian, periaatteiden ja transformoimisen kautta. Luvussa 3 esitellään Leanin syntyhistoriaa, periaatteita, Kanbanin käyttöä ja Lean-transformaatiota ohjelmistokehityksen näkökulmasta. Luvussa 4 tarkastellaan Leanin mukaisen toiminnan arviointia, sen pääperiaatteita sekä esitellään LESAT-arviointityökalu. Luvussa 5 tarkastellaan Lean-ohjelmistotuotannon ja -kehityksen arviointia sekä esitellään muutamia yleisimpiä arviointimenetelmiä. Luvussa 6 esitetään tutkielmaa koskeva pohdinta ja esitetään muutamia kriittisesti Leaniin, Lean-ohjelmistotuotantoon ja sen arviointiin suhtautuvia lähteitä. Luvussa 7 annetaan koko tutkielmasta tiivis yhteenvedo.

2 LEAN-AJATTELU

Tässä luvussa esitellään Lean-ajattelun pääperiaatteita ja sen syntyyn johtaneita syitä sekä transformaatiokeinoja, eli kuinka toiminta saataisiin muutettua Lean-periaatteiden mukaiseksi. Luvussa käydään läpi Lean-ajattelua ja Lean-periaatteiden mukaista toimintaa perinteisen tuotantoteollisuuden näkökulmasta, josta se on alun perin lähtöisin.

2.1 Leanin syntyhistoria

Teollisuustuotannossa tunnetun Lean-tyyppisen ajattelun kehityksen aloitushetkenä voidaan pitää 15. elokuuta 1945. Japani oli tuolloin juuri hävinnyt sodan ja maan autoteollisuus oli hyvin vakavassa kriisissä. Silloinen Toyota Motor Companyn pääjohtaja Toyoda Kiichiro antoi johtamalleen yritykselle kolme vuotta aikaa saada kiinni amerikkalaiset kilpailijat, silloiset maailman suurimmat autovalmistajat, Ford ja General Motors. Toyotan tekemien laskelmien mukaan eurooppalaisiin ja amerikkalaisiin kilpailijoihin verrattuna tarvittiin sata japanilaista työntekijää suorittamaan sama työmäärä kuin mikä näissä maissa saatiin aikaiseksi kymmenellä työntekijällä. Tämän toimeksiannon tuloksena kehitettiin *Toyotan tuotantojärjestelmänä* (Toyota Production System, TPS) tunnettu teollisuuden tuotantomenetelmä. (Ohno, 1988, s. 3.)

Likerin (2004, s. 24) mukaan 1960-luvulla TPS oli Toyotalla omaksuttu toimintaa ohjaavana filosofiana niin laajasti, että sitä alettiin levittää myös heidän tärkeimmille tavarantoimittajilleen. Näin saatiin TPS:n tarjoamat hyödyt käyttöön koko toimitusketjun matkalla.

Parin kymmenen vuoden kehityksen jälkeen TPS ei kuitenkaan ollut vielä levinnyt juurikaan Toyotan ja sen tavarantoimittajien ulkopuolelle. Vaikka Toyotalla TPS oli jo laaja-alaisessa käytössä, ei se herättänyt kovinkaan laajaa mielenkiintoa muissa yrityksissä tai aloilla. (Ohno, 1988, s. 1.) Syksyllä 1973 ensimmäisen öljykriisin iskiessä maailmanlaajuisesti oli Japanin talous nollakasvun pisteessä ja yritykset eri aloilla alkoivat kärsiä hätää. Toyotalla asiat olivat

toisin. Vaikka sekin kärsi öljykriisin vaikutuksista tuottojen osalta, niin vuosina 1975-1977 Toyotan onnistui kasvattamaan kuilua oman tuloksensa ja maan yleiseen tasoon verrattuna. Tässä vaiheessa muiden alojen yritysten katseet alkoivat siirtyä Toyotan suuntaan, mitä siellä tehtiin erilailla, mikä mahdollisti poikkeaman yleisestä trendistä. (Ohno, 1988, s. 1.)

Tässä vaiheessa Japanin hallitus teki tilanteen korjaamiseksi aloitteen TPS-seminaarien käynnistämisestä, vaikka se ymmärsi vain murto-osan Toyotan tavan mukaisesta toiminnasta. Puutteellisin tiedoin järjestetyt seminaarit eivät tarjonneet parasta mahdollista käsitystä TPS:stä muille. Mitä kauemmaksi Toyota Citystä ja Toyotan tavarantoimittajista edettiin, sitä laimeampaa ja heikompaa TPS-periaatteiden noudattaminen oli. (Liker, 2004, s. 24.)

TPS tuli maailmanlaajuisesti tunnetuksi, ja nimettiin Leaniksi, kuitenkin vasta vuonna 1990 James Womackin, Daniel Rossin ja Daniel Jonesin teoksessa "The machine that changed the world" (Womack ym., 1990). Toinen merkittävä teos oli vuonna 1996 ilmestynyt Womackin ja Jonesin kirja "Lean thinking" (Womack & Jones, 1996). Nämä teokset johtivat maailmanlaajuiseen autoteollisuuden tuotantomenetelmien muutokseen, joka sittemmin on levinnyt myös muille alueille.

2.2 Periaatteet

Lean kehitettiin alun alkaen nimenomaan tehostamaan tuotantoa. Leanin perimmäisenä ajatuksena onkin tuottaa mahdollisimman vähällä mahdollisimman paljon. Vähyys ilmenee esimerkiksi resursseissa, varastossa, työntekijöiden määrässä ja tuotantoon tarvittavassa ajassa. (Womack & Jones, 2003, s. 15.) Seuraavassa kuvataan Leanin tärkeimpiä periaatteita.

2.2.1 Viisi avainperiaatetta

Lean perustuu viiteen avainperiaatteeseen. Nämä kaikki viisi periaatetta tulisi omaksua käyttöön, jotta saataisiin luotua Leanin mukaisia prosesseja ja lopulta saavutettaisiin jopa niin kutsuttu Lean-organisaatio. Nämä viisi periaatetta ovat arvo, arvovirta, virtaus, imu ja täydellisyys. (Womack ym., 1990; Womack & Jones, 1996.)

Jokaisen tuotteen päätarkoitus on tuottaa asiakkaalle lisäarvoa. Tuotteen *arvon* (value) määrittäminen asiakkaalle on sidottu tiiviisti tuotteen kehittämisen visioon ja strategiaan. Mikäli yritys ei tunne omaa toimialaansa ja arvoa, jonka sen tuotteet tarjoavat asiakkaille, johtaa tämä helposti tilanteeseen, jossa yritys keskittyy vain lyhyen tähtäimen ongelmiin ja pieniin kustannusten leikkauksiin. (Maglyas, Nikula ja Smolander, 2012, s. 41.)

Kun arvo on tunnistettu, niin seuraavana vaiheena on tärkeää tehdä *arvovirrasta* (value stream) kuvaus, joka sisältää vaihe vaiheelta kaikki askeleet ideasta toimitettavaan tuotteeseen asti. Kuvauksen tarkoituksena on tunnistaa ja

poistaa kaikki turhat vaiheet, jotka eivät luo arvoa. Tällä tavoin yritys saa eliminoitua turhat vaiheet ja pienennettyä sykli-aikoja. (Maglyas ym., 2012, s. 41.) Tässä tutkimuksessa sykliajoilla tarkoitetaan yhden tehtävän tai selkeän prosessin vaiheen suoritus-aikaa, ei siis koko prosessin läpimenoaikaa.

Arvoa tuottamattomien vaiheiden eliminoinnin jälkeen yrityksen tulisi ta-soittaa sen arvon tuottamisprosessia. Tämän saavuttamiseksi Lean ehdottaa keskittymistä koko arvon luomisprosessiin yksittäisten ja eristettyjen osien sijasta. Tuotannon järjestäminen yhtenäiseksi virtaukseksi lisää tuottavuutta ja vähentää virheiden määrää sekä tarvittavia resursseja. Tämä periaate tunnetaan nimellä *virtaus* (flow). (Maglyas ym., 2012, s. 41.)

Imulla (pull) tarkoitetaan sitä, kun asiakas lähettää pyynnön tuotteesta, niin yritys tuottaa tuotteen täyttäen asiakkaan tarpeet mahdollisimman nopeasti. Yrityksen jo omaksuttua virtausprosessin toimintaansa lyhenee aika ajatuksesta tuotteen toimitukseen. Tätäkin tärkeämpänä kohtana tulisi huomioida imuohjauksen edut yrityksen näkökulmasta; vältetään ylimääräistä tuotantoa ja turhien tuotteiden varastointia. (Maglyas ym., 2012, s. 41.)

Lehtisen (2011, s. 28) mukaan Womack ym. (1990) määrittelee imuohjauksen siten, että tehdään juuri se mitä asiakas tarvitsee juuri silloin, kun hän sitä tarvitsee. Toisin sanoen ei tehdä mitään, ennen kuin asiakas sitä tarvitsee. Tästä mallista käytetään myös nimitystä Just in time (JIT), eli juuri oikeaan aikaan.

Viides avainperiaate, eli *täydellisyys* (perfection) on tulosten analysointia ja mahdollisten muutosten toteutuksen suunnittelua. Siinä missä neljä edeltävää periaatetta keskittyvät siihen mitä tehdä, niin täydellisyys keskittyy siihen, kuinka itse asiassa tehdä toiminnot. (Maglyas ym., 2012, s. 41.)

Näiden viiden avainperiaatteen lisäksi eräs keskeinen Leaniin liittyvä termi on hukka (waste), joka liittyy erittäin läheisesti toiseen Leanin keskeiseen käsitteeseen, arvoon (Womack ym., 2003).

2.2.2 Lisäarvoa tuottamaton hukka

Helpoin ja eniten Leanin yhteydessä käytetty keino arvon ja tuottavuuden lisäämiseksi on Ohnon (1988), Womackin ym. (1990), Womackin ja Jonesin (1996, 2003) sekä Likerin (2004) mukaan hukan eliminointi.

Hukkaa on kolmen tyyppistä: Muda, Mura ja Muri. *Mudalla* tarkoitetaan lisäarvoa tuottamatonta toimintaa, jota on tunnustettu seitsemän perustyyppiä: ylituotanto, odottelu, tarpeeton kuljetus, ylikäsittely, liiallinen varasto, tarpeeton liike ja viat. *Mura* on epätasaisuudesta tai varianssista johtuvaa hukkaa. *Murilla* tarkoitetaan ihmisten, laitteiden tai järjestelmien ylikuormittamisesta johtuvaa hukkaa. (Hines, Found, Griffiths & Harrison, 2011, s. 5-9.)

Likerin (2004, s. 27) mukaan on tärkeää oppia kartoittamaan toiminnot, jotka lisäävät tuotteen arvoa ja auttavat pääsemään eroon lisäarvoa tuottamattomista toiminnoista jo senkin takia, että monet TPS:n työkaluista ja Toyotan periaatteista juontavat juurensa tästä.

Poppendieckin ja Poppendieckin (2003, s. 4) mukaan jos jokin ei suoraan lisää arvoa kuluttajalle, se on hukkaa, tai vastaavasti jos on olemassa jokin tapa

tehdä ilman, se on hukkaa. Poppendieckin ja Cusumanon (2012, s. 28) mukaan Lean-ajattelun mukainen hukka on mitä tahansa, mikä ei lisää asiakkaan kokemaa arvoa suoraan, tai vastaavasti ei lisää tietoa siitä, kuinka toimittaa arvoa tehokkaammin.

Ensimmäinen kysymys TPS:ssä on aina: ”Mitä asiakas haluaa tästä prosessista?” Asiakkaalla tarkoitetaan niin sisäistä (tuotantolinjan seuraavat vaiheet) kuin ulkoista eli lopullista asiakasta, ja vastaus tähän kysymykseen määrittää arvon. Näin prosessia tutkimalla voidaan erottaa lisäarvoa tuottavat vaiheet lisäarvoa tuottamattomista. (Liker, 2004, s.27.)

Tämän tutkielman kannalta mielenkiintoisena kohtana Likerin (2004, s. 27) mukaan tätä valmistusprosessin tutkimista asiakkaan silmin voidaan soveltaa mihin tahansa prosessiin valmistuksessa, tiedotuksessa ja palveluissa. Toisin sanoen hukan tunnistamista voidaan soveltaa missä tahansa prosessissa myös teollisuuden tuotantoprosessin ulkopuolella.

Likerin (2004) mukaan Toyota on tunnistanut seitsemän lisäarvoa tuottamattoman hukan päätyyppiä liiketoiminta- ja valmistusprosesseissa. Näitä periaatteita voidaan soveltaa tuotantolinjan lisäksi tuotekehityksessä, tilausten vastaanottamisessa sekä toimistoissa. Liker (2004, s. 28) on itse lisännyt mukaan kahdeksannen hukkatyyppin, jota ei ole alkuperäisessä Toyotan dokumentaatiossa kirjattu omana kohtanaan. Päätyypit ovat (Liker, 2004, s.28):

1. *Ylituotanto*. Tilaamattomien osien valmistaminen, mikä aiheuttaa tarpeetonta henkilökunnan palkkaamista ja varasto- ja kuljetuskustannuksia liiallisen varastoinnin vuoksi.
2. *Odottelu*. Työntekijät joutuvat vain seuraamaan automatisoitua konetta tai seisoskelemaan odotellen seuraavaa käsittelyvaihetta, työkalua, toimitusta, komponenttia jne. tai heillä ei yksinkertaisesti ole mitään tekemistä varaston loppumisen, käsittelyviiveiden, välineistön sammuttamisen ja kapasiteetin pullonkaulojen vuoksi.
3. *Tarpeeton kuljettelu*. Keskeneräisen työn kuljettaminen pitkiä matkoja, tehoton kuljetuksen luominen tai materiaalien, osien tai valmiiden hyödykkeiden siirtely varastoon, varastosta tai prosessista toiseen.
4. *Ylikäsittely tai virheellinen käsittely*. Tarpeettomien vaiheiden suorittaminen osien käsittelyssä. Tehoton käsittely kehnon työkalun tai tuotesuunnittelun vuoksi, mistä aiheutuu tarpeetonta liikkumista ja virheitä tuotteeseen. Hukkaa syntyy myös, jos tuotetaan laadukkaampia tuotteita kuin välttämätöntä.
5. *Tarpeettomat varastot*. Liikaa raaka-aineita, keskeneräisiä tuotteita tai valmiita hyödykkeitä, mistä seuraa pidempiä läpimenoaikoja, vanhentuneisuutta, vahingoittuneita hyödykkeitä, kuljetus- ja varastokustannuksia ja viivettä. Lisäksi liian suuret varastot kätkevät sellaisia ongelmia kuin tuotannon epätasapainon, myöhästyneet toimitukset alihankkijoilta, viat, välineistön alhaallaoloajan ja pitkät asennusajat.

6. *Tarpeeton liikkuminen.* Kaikki turha liike, mitä työntekijöiden täytyy suorittaa työaikana, kuten osien, työkalujen jne. etsiminen, kurkottelu ja pinoaminen. Myös kävely on hukkaa.
7. *Viat.* Viallisten osien tuottaminen tai korjaaminen. Korjaaminen tai uudelleen työstäminen, pois heittäminen, täydennysosan tuottaminen ja tarkastus tarkoittavat tarpeetonta käsittelyä, hukattua aikaa ja turhaa työtä.
8. *Työntekijän luovuuden käyttämättä jättäminen.* Ajan, ideoiden, taitojen, parannusten ja oppimismahdollisuuksien hukkaaminen, kun työntekijää ei sitouteta tai kuunnella.

Ohno (1988) pitää ylituotantoa tärkeimpänä hukkana, koska se aiheuttaa suurimman osan muusta tuhlauksesta. Missä tahansa prosessin vaiheessa ylituotanto johtaa varaston kertymiseen ja tämän seurauksena materiaalia lojuu odottamassa käsittelyä seuraavassa vaiheessa. Ongelmaksi tässä muodostuu prosessien välisten varastopuskureiden koon kasvaminen, mikä johtaa pitkällä ajanteella epäoptimaaliseen toimintaan, kuten heikentyvään motivaatioon toimintojen jatkuvan parantamisen suhteen. (Liker, 2004, s.29.)

2.2.3 Kanban

Japanin kielessä kan tarkoittaa korttia ja ban signaalia. Näin ollen kanban voidaan vapaasti kääntää signaalikortiksi. Jokainen Kanban-kortti sisältää pienen määrän tietoa tehtävästä, eli ne ovat ikään kuin kehittyvä tarina. Kortit voidaan helposti järjestää sekä uudelleen järjestää siten, että seuraava työntekijä tietää ottaa päällimmäisen kortin ja käynnistää työt heti kyseisen kortin sisältämän informaation perusteella. (Poppendieck & Poppendieck, 2006, s. 138.)

Kortit voivat olla joko fyysisiä kortteja taululla tai elektronisia kortteja tietojärjestelmässä. Valinta näiden kahden tavan välillä riippuu usein siitä, työskenteleekö kortteja käyttävä tiimi fyysisesti samassa paikassa vai esimerkiksi maantieteellisesti hajallaan. (Poppendieck & Poppendieck (2006, s. 138.)

Kanbanilla on omat etunsa ja haasteensa. Kanbanin etuina pidetään muun muassa sitä, että ihmiset tietävät helposti mitä tekevät seuraavaksi, kun kortteja on jonossa jo valmiina odottamassa. Kortteihin kuitenkin liittyy haaste sen suhteen, onko kortti ylipäätään ollut oikeassa paikassa ja onko sen sisältö kaikilta osin täysin validia. (Poppendieck & Poppendieckin 2006, s. 138.)

Kortin sisältämän informaation tilanteesta ja tehtävästä vaiheesta ei tarvitse olla kovin yksityiskohtaista. Poppendieckin ja Poppendieckin (2006, s. 138) mukaan kortti itsessään ei ole tehtävän spesifikaatio, vaan ainoastaan signaali siitä, että seuraavana tehtävänä on kerätä yhteen oikeat henkilöt luomaan yksityiskohtaiset suunnitelmat, verifikaatio ja kortilla olevan tehtävän toteutus.

2.3 Transformaatio

Siirtyminen perinteisestä toiminnasta Lean-ajattelun mukaiseen toimintaan ei ole ongelmaton. Kuuselan, Huomon ja Korkalan (2012) mukaan Lean vaatii huomattavaa panostusta ajallisesti koko organisaation henkilöstön kouluttamiseen, jotta toiminta saadaan muutettua Lean-tyyppiseksi. Muina ongelmina mainitaan muun muassa ympäristöllinen näkökulma, sillä Lean-tuotanto voi lisätä raaka-aineiden kuljetusten määrää. (Kuusela ym., 2012, s. 12.)

Useat eri alojen yritykset, joissa on yritetty omaksua käyttöön Lean-periaatteita, ovat epäonnistuneet, koska implementointi on jäänyt liian pintapuoliseksi. Näissä yrityksissä on keskitytty liikaa pelkkiin prosessityökaluihin ymmärtämättä, että Lean on kokonainen järjestelmä, jonka täytyy ulottua koko organisaation kattavaksi. Usein näissä yrityksissä on unohdettu erityisesti ylemmän johdon osallistuminen päivittäisiin operaatioihin ja jatkuvaan parantamiseen. (Liker, 2004, s. 7.)

Seuraavassa kerrotaan ensin transformaation haasteista ja sen jälkeen keinoista.

2.3.1 Transformaatiohaasteet

Kuuselan ja Koivuluoman (2001) mukaan useat aiemmat tutkimukset ovat tulleet siihen tulokseen, että yrityksen muuntautuminen Leanin mukaiseksi ei onnistu kehitysstrategialla, joka kulkee pelkästään ylhäältä alaspäin, vaan tämän lisäksi tarvitaan alhaalta ylöspäin suuntautuvaa lähestymistä. Myös Hinesin, Holwegin ja Richin (2004) mukaan Leania on olemassa kahdella eri tasolla, strategisella ja operatiivisella tasolla. Asiakaskeskeinen strateginen ajattelu on sovellettavissa missä vain, kun taas "lattiataason" työkalut eivät. Tämä on usein johtanut sekaannuksiin tai väärinkäsityksiin siitä, missä soveltaa Leania. (Hines ym., 2004.)

Bozdoganin, Milauskasin, Mizen, Nightingalen, Tanejan ja Tonaszuckin (2000, s. 51) mukaan onnistuneen Lean-transformaation saavuttaminen on valtaisa tehtävä, jonka aikana kohdataan monia suuria haasteita. Lean-oppaassaan he käsittelevät joitain merkittävimpiä Leanin käyttöönottoon liittyviä haasteita, joita käsitellään seuraavaksi tarkemmin.

Tietoisuus ja ymmärrys Leanista (awareness and understanding of Lean). Lean-käsitteet ovat verrattain uusia, eivätkä laajalti tunnettuja tai täysin ymmärrettyjä. Jotkin Lean-käsitteet ovat epäintuitiivisia verrattuna perinteiseen tietouteen. Menetelmät Leanin mukaiseen toimintaan eivät ole vielä täysin kypsiä. Osaavista Lean-asiantuntijoista on puutetta. Lean vaatii myös syvää ymmärtämistä, joka saavutetaan vain Lean-periaatteiden jatkuvalla toistamisella. Kokonaisvaltainen hyväksyminen ja implementointi jäävät usein puolitiehen, vaikka johtajat olisivatkin halukkaita pilotoimaan Leania. Lean sekoitetaan muihin johtamisstrategioihin. Lean-aloitteet kohtaavat johdon ja työntekijöiden taholta kyynisyyttä. Ne koetaan vain yhtenä uutena villityksenä. Joskus Leanin

implementointi vaatii yritykseltä hyppyä tuntemattomaan, tai omalta mukavuusalueelta poistumista. (Bozdogan ym., 2000, s. 51.)

Syvälle ajatusmaailmaan juurtunut massatuotanto mentaliteetti (deeply engrained mass-production mentality). Massatuotantomentaliteetille tyypillisesti pyritään minimoimaan koneiden tyhjäkäyttö ja alhaallaoloaika, työntekijöiden toimeton aika, tyhjät varastot sekä työvoimakustannukset. Mikäli työntekijöitä motivoidaan saavuttamaan nämä tehokkuusmittarit, lopputulos johtaa usein Lean-periaatteiden vastaiseen tulokseen, kuten: pitkiin tuotantoaikoihin erissä, suuriin varastoihin (raaka-ainevarastoissa sekä valmiissa tuotteissa), suuriin virhemääriin (vialliset ja vanhentuneet tuotteet), kattaviin uudelleentyöstö- ja korjausalueisiin, pitkiin tuotantojonoihin laitteilla ja itse aiheutettuihin vaihteluihin kysynnässä. Näiden kaltainen massatuotantoajattelu ulottuu negatiivisilla vaikutuksillaan myös itse tuotannon ulkopuolelle ja vaikuttaa väistämättä koko organisaatioon. (Bozdogan ym., 2000, s. 51-52.)

Johdon vastarinta (management resistance). Niin kauan kun yrityksestä puuttuu syvälinen ymmärrys Lean-periaatteista ja -ajattelusta, yritykset voivat odottaa vastarintaa erityisesti keskitason johdolta. Vastarintaa ruokkivia uskomuksia ovat muun muassa seuraavat. Lean johtaa voiman, arvovallan ja auktoriteetin heikkenemiseen. Lean edustaa suuren luokan kaavamutosta, joka sotii perinteistä ja tuttua tietoutta vastaan. Leanin tiimien johtamiseen perustuva johtamismalli koetaan uhkaavana. Johtajat kokevat uhkana Leanin mukanaan tuoman niin sanotun turvaverkkojen poistamisen, jossa luovutaan ylijäämävarastoista, liikkapasiteetista ja kaunistelluista ennusteista. (Bozdogan ym., 2000, s. 52.)

Työntekijöiden kokemat huolet (employee concerns). Johtoportaan tapaan monet työntekijät voivat kokea Leanin epämuukavana ja pelottavana. Siirtyminen pois vanhasta tiukasti osastoidusta ja järjestelmällisestä työstä koetaan uhkaavana. Lean korostaa jatkuvaa oman työn tarkkailua ja parantamista. Työturvallisuuden pelätään heikentyvän. Työntekijät pelkäävät intensiteetin kasvua, päättymättömiä tehokkuusvaatimuksia ja ylityötä. Pienemmän työtehtävien yhteislukumäärän koetaan heikentävän uralla etenemistä. Pelätään työtapaturmien kasvun lisääntymistä ympäristössä, jossa on vain vähän varaa tapaturmille. Pelätään standardoidun työn tuovan mukanaan uutta jäykkyyttä ja tähän liittyviä jatkuvasti lyheneviä työsyklejä. Ylijäämävaraston ja muiden pus-kureiden poisto koetaan kasvavina paineina. Työntekijän ja yrityksen välisen suhteen muuttuessa kyseenalaistetaan lojaliteetti. (Bozdogan ym., 2000, s. 52.)

2.3.2 Transformaatiomalleja

Koska Leanin mukaiseen toimintaan siirtymisen on havaittu tuottavan yrityksille ongelmia ja jotkut pilotoidut Lean-kokeilut eivät ole päättyneet toivotulla tavalla, on yritysten avuksi kehitetty erilaisia transformaatiomalleja ja ohjeistusta. Hinesin ym. (2004) mukaan Donaldson (1996, s. 51) toteaa, ettei ole olemassa vain yhtä täydellisen lähestymistavan mallia, joka olisi sovellettavissa tehokkaasti kaikkiin organisaatioihin, vaan malli täytyy valita tilannekohtaisesti. Seu-

raavassa kuvataan tarkemmin kahta transformaatiomallia, joista toinen on Bozdoganin ym. (2000) esittämä ja toinen Likerin (2004) malli.

Bozdogan ym. (2000, s. 55) esittelevät yleisen mallin organisaation avuksi sen siirtyessä Lean-periaatteiden mukaiseksi. Ohjeistus sisältää organisaatiotasolla yleisluotoisen kuvauksen vaiheista, jotka panevat alulle, ylläpitävät ja jatkuvasti parantavat organisaation transformaatiota Lean-periaatteiden ja -käytänteiden mukaiseksi.

Malli rakentuu kolmesta syklistä. Ensimmäinen sykli on sisäänkäynti- / paluusykli, joka määrittelee Leanin käyttöönottopäätökseen liittyvät toimet. Tämä ensimmäinen sykli on hyvin läheisesti liitoksissa organisaation strategiseen suunnitteluun. Toinen sykli on pitkän aikavälin sykli, jossa ympäristö ja olosuhteet valmistellaan transformaatiota varten. Tässä vaiheessa organisaatio valmistellaan tarkempaa transformaatiosuunnittelua ja implementointia varten. Kolmas sykli on lyhyen aikavälin sykli, jossa transformaation yksityiskohtainen toteutus suunnitellaan, suoritetaan ja sen toteutumista seurataan. (Bozdogan ym., 2000, s. 55.) Kuviossa 1 on esitetty Bozdoganin ym. (2000, s. 55) määrittelemät kolme sykliä ja transformaatiovaiheet.



KUVIO 1 Lean-transformaatiomalli (Bozdogan ym., 2000, 55)

Bozdoganin ym. (2000) määrittelemät transformointivaiheet ovat:

- *Omaksu Lean-paradigma* (Adopt Lean paradigm). Paradigman toteuttaminen vaatii jokaisen asiakaskontaktin, tuotesuunnittelun, tuotannon, laadunvarmistuksen, henkilöstöresurssien, työnvalvonnan, organisaatiokenteen, järjestelmien ja hankkijasuhteiden kanssa liitoksissa olevien ole-tusten, käytänteiden ja prosessien uudelleen läpikäymistä. Organisaation

täytyy oppia tekemään liiketoimintaa, käyttäytymään ja näkemään arvoa perinteisestä täysin poikkeavalla tavalla.

- *Keskity arvovirtaan* (Focus on the value stream). Lean-ajattelun ensisijainen käsite on se, että kaikkien organisaation toimintojen ja resurssien tulisi olla keskittyneitä arvon luomiseen. Jokainen toiminto tai resurssikulu, joka ei liity arvon luomiseen on hukkaa ja tulisi näin ollen eliminoida mahdollisimman hyvin pois.
- *Kehitä Lean-rakenne ja -käyttäytyminen* (Develop Lean structure and behavior) Lean-organisaatio eroaa perinteisestä massatuotanto-organisaatiosta sekä rakenteeltaan että käyttäytymiseltään. Tässä transformaatiomallin vaiheessa luodaan ajatusmaailma ja olosuhteet valmiiksi Lean-käytänteiden ja -periaatteiden implementointia varten.
- *Luo ja hio implementointisuunnitelma* (Create and refine implementation plan). Siirryttäessä lyhyen aikavälin sykliin ja halutunlaisen transformaa-tion saavuttamiseksi täytyy tunnistaa ja priorisoida ne Lean-aloitteet, jotka yhdessä muodostavat organisaatiotason suunnitelman. Tämä vaatii resursointia suunnitelmaan sekä mittavissa määrin koulutusta ja harjoittelua.
- *Implementoi Lean-aloitteet* (Implement Lean initiatives). Tässä vaiheessa muutokset käytänteissä ja menettelyissä varsinaisesti implementoidaan käyttöön. Määritellään spesifit toiminnot, ohjelmat ja projektit, jotka toteutetaan jokaisella organisaation osa-alueella, ja päätetään, kuinka ne implementoidaan järjestelmätasolla. Näitä yksityiskohtaisia toimintasuunnitelmia toteutetaan, tarkkaillaan ja muokataan vaatimusten mukaisesti.
- *Keskity jatkuvaan parantamiseen* (Focus on continuous improvement). Samanaikaisesti kun yksityiskohtaisilla suunnitelmilla saadaan aikaan muutosta, niin muutoksia mitataan ja arvioidaan. Korjaavat toiminnot ovat osa jatkuvan parantamisen prosessia. Pyritään tunnistamaan jatkokehittämismahdollisuuksia. Yksityiskohtaiset korjaavat toiminnot palautetaan lyhyen aikavälin syklin alkuun syötteenä ”Luo ja hio implementointisuunnitelma”-kohtaan, josta ne voidaan ottaa käyttöön seuraavan organisaatiotason suunnitelman iteroinnissa. Tätä perusteellisemmän tason muutokset palautetaan pitkän aikavälin sykliin, jossa tehdään tarvittavia muutoksia organisaation rakenteeseen ja käyttäytymiseen.

Bozdoganin ym. (2000) transformaatiomalli ei ole ainoa kirjallisuudessa esitetty. Vastineeksi Liker (2004, s. 302) määrittelee 13 ohjetta, joita voi käyttää yrityksen muuntamisessa Leanin mukaiseksi. Ohjeet kuvataan seuraavassa lyhyesti.

Muutos kannattaa aloittaa teknisestä järjestelmästä ja täydentää sitä nopeasti kulttuurin muutoksella. TPS:n sosiaaliset ja tekniset järjestelmät ovat toisiinsa läheisesti liittyviä, ja yrityksen halutessa muuttaa kulttuuriaan tulee sen kehittää myös aitoja Lean-johtajia, jotka kykenevät vahvistamaan ja johtamaan kulttuurin muutosta. Johtajilta vaaditaan sitoutumista arvovirran kartoitukseen

sekä lattiatason muutokseen, jotta he oppivat havaitsemaan hukkaa. (Liker, 2004, s. 302.)

Likerin (2004) mukaan yksi yleinen virhe Lean hankkeiden aloittamisessa on tehdä opetuksesta liian teoreettista ja PowerPoint-esityksiin perustuvaa. Toyotan tapa tarkoittaa oppimista tekemisen kautta, ja Likerin mukaan erityisesti alkuvaiheessa opetuksen tulisi olla vähintään 80 % tekemistä ja 20 % muuta teoreettisempaa koulutusta ja informointia. Toyotan tavan mukainen lähestyminen asettaa ihmiset erilaisiin ongelmatilanteisiin ja antaa heidän itsensä löytää omat ratkaisunsa.

Liker (2004, s. 303) kehottaa ottamaan käyttöön aluksi arvovirtapilottiohjelman, jolla voidaan havainnollistaa Leania järjestelmänä ja tarjota mahdollisuus nähdä hyödyt käytännössä. Tämä tarkoittaa sitä, että tietyn tuoteperheen määrittämän arvovirran sisällä luodaan malli. Tämä malli sisältää kokonaisen työkalujärjestelmän ja lopulta henkilöstökäytäntöjen toteuttamista siten, että se tarjoaa myös muille yrityksen työntekijöille mahdollisuuden nähdä Leanin käytännössä ilman, että heidän tarvitsee mennä tutustumaan siihen johonkin toiseen yritykseen. Tehtaissa malli rakennetaan yleensä yhden Lean-tuotelinjan ympärille alkaen raaka-aineiden vastaanottamisesta ja päättyen valmiisiin hyödykkeisiin. Palveluorganisaatiossa taas malli on joku kokonainen liiketoiminta-prosessi alusta loppuun organisaation rajojen sisällä. (Liker, 2004, s. 303.)

Arvovirran kartoitusta tulisi soveltaa tulevan tilan visioita kehitettäessä. Arvovirran kartoitus on menetelmä, joka näyttää selkeästi kaavion muodossa materiaalin ja informaation etenemisen. Nykyisen tilan karttaa, tulevan tilan karttaa ja toteutussuunnitelmaa kehitettäessä tulisi käyttää laaja-alaista ryhmää, joka sisältää sekä prosessiin liittyviä johtajia että työntekijöitä. Näin tiimi oppii parhaiten, kun se oppii yhdessä näkemään hukkaa nykyisessä tilassa. Tulevan tilan karttaa selvittäessä he pääsevät yhdessä soveltamaan Lean-työkaluja ja -filosofiaa. (Liker, 2004, s. 303.)

Huomioitavaa tässä vaiheessa on se, että arvovirran kartoitus tulisi tehdä vain tiettyihin tuoteperheisiin kerralla, jotka muutetaan välittömästi. Useimmat koko tehtaan ja sen kaikkien tuotteiden osalta toteutetut kartoitukset päätyvät kaaokseen ja valtavaan määrään dataa. (Liker, 2004, s. 303.)

Kaizen-työpajoja (inkrementaalinen jatkuva parantaminen) käytettäessä on mahdollista kouluttaa ihmisiä ja tehdä nopeita muutoksia toimintaan. Saalessaan johtajatasolta aikaa ja tukea oikeanlaiset ja motivoituneet ihmiset sisältävä Kaizen-työpaja vapauttaa monialaisen tiimin tekemään viikossa muutokset, jotka voisivat muutoin viedä kuukausia. Kaizen-työpajat toimivat parhaiten, kun niitä käytetään yhtenä työkaluna tulevan tilan arvovirtakartoituksen osoittamien parannusten toteuttamisessa, eikä niin kutsutussa ”piste-kaizenissa” eli yksittäisten ongelmien korjaamisessa, ilman perustavan laatuisen arvovirran suoristamista. (Liker, 2004, s. 304.)

Useimmat organisaatiot ovat organisoineet johtoportaanensa tietyn prosessin tai toiminnon mukaan. Tehdasympäristössä esimerkkejä tällaisesta ovat maalausosaston johtaja, kokoonpano-osaston johtaja ja huolto-osaston johtaja. Näin ollen johtajilla on vastuullaan vain yksi vaihe prosessista, jossa luodaan

arvoa asiakkaille, eikä kukaan ole vastuussa koko arvovirrasta. (Liker, 2004, s. 304.) Likerin (2004, s. 304) mukaan Womack ja Jones (2003) suosittelivat erillisten arvovirtajohtajien luomista, joilla on täysi vastuu arvovirrasta ja jotka voivat vastata suoraan asiakkaalle. Yhdellä arvovirtajohtajalla on täysi käskyvalta kaikkiin resursseihin, joita valmistuksessa tai palvelussa tarvitaan.

Liker (2004) nostaa esille myös Womackin ja Jonesin (2003) toisen tyyppisen toteutuksen, jossa käytössä on matriisiorganisaatio. Organisaatiossa on edelleen osastopäälliköitä, mutta heidän lisäksi myös arvovirtapäälliköitä, jotka vastaavat Toyotan tavan mukaisia pääinsinöörejä. Molemmissa toteutuksissa viesti on sama; jonkun jolla on kunnolliset johtajataidot ja syvälinen ymmärrys tuotteesta ja prosessista täytyy olla tilivelvollinen asiakkaille ja vastuussa prosessista, jossa luodaan arvoa asiakkaille. (Liker, 2004, s. 304.)

Likerin (2004, s. 304) mukaan muutoksen tulisi olla pakollista. Lean-transformaatio ei tule tapahtumaan lainkaan tai ainakaan kunnolla, jos yritys suhtautuu siihen mukavana vapaa-ajan harrastuksena tai vapaaehtoistyönä. Kriisi voi helposti laukaista Lean-hankkeen, mutta se ei ole välttämätöntä yrityksen muuttamiseksi. Kriisitilanteessa johdon ja työntekijöiden suhtautuminen Leaniin tulee varmasti olemaan vakavaa. Kuitenkin on useita onnistuneesti Lean-transformaation läpikäyneitä yrityksiä, jotka eivät ole olleet ongelmissa ja joissa yrityksen johto on tukenut parannusta jo etukäteen. Tärkeänä tulisi muistaa se, että Lean-johtajuus keskittyy nimenomaan pitkän tähtäimen oppimiseen. (Liker, 2004, s. 304.)

Suuria taloudellisia mahdollisuuksia selvitetessä tulisi olla opportunistinen. Liker (2004) painottaa koko ajan sitä, että Toyotan tapauksessa he keskittyvät prosessien parantamiseen luottaen siihen, että tämä parantaa taloudellisia tuloksia. Kun yritys ei vielä usko Lean-filosofiaan sydämensä pohjasta, on sille erityisen tärkeää saavuttaa joitakin suuria voittoja. Kun valitaan oikea tuoteperhe ja kun käytössä on kokenutta Lean-asiantuntemusta, ja asian eteen tehdään vakavia ponnisteluja, on noin 100 % mahdollisuus tehdä valtavia ja näkyviä parannuksia, jotka vakuuttavat kenet tahansa johtajista. (Liker, 2004, s. 304-305.)

Mittarit tulisi kohdistaa uudelleen arvovirran ympärille. Toyotan tapaa käyttää mittareita poikkeaa monista muista yrityksistä huomattavasti. Toyotalla mittarit ovat kokonaisvaltainen työkalu, joilla seurataan yrityksen edistymistä, ja ne ovat avaintekijä jatkuvaan parantamiseen. Useimmissa muissa yrityksissä mittareita käytetään lähinnä lyhyen tähtäimen kustannustenhallintatyökaluina johtajille, jotka eivät syvälinen ymmärrä johtamiaan asioita. Esimerkkinä tällaisesta on yritysten seuranta epäsuoran ja suoran työpanoksen suhteesta. Suhde saadaan näyttämään hyvältä teettämällä runsaasti suoraa työtä ja pitämällä ihmiset kiireisinä, vaikka he tuottaisivat ylituotantoa tai tekisivät turhaa työtä. Mikäli tähän tuodaan mukaan Toyotan rakenteen mukainen tiiminvetäjän apurooli, suhde vahingoittuu ja väärillä (ei-Leanin mukaisilla) mittareilla mitattuna tämä voi johtaa jopa irtisanomisiin. Tämän takia tulisikin poistaa kaikki mittarit, jotka eivät ole Leanin mukaisia ja jotka pilaavat niiden työn, jotka vakavissaan pyrkivät operatiivisen tason parantamiseen. Tämän jälkeen tulisi tehdä

arvovirtamittauksia läpimenoajasta varastoasteisiin ja kerralla saatuun laatuun. Näihin mittauksiin tulisi suhtautua yhtä vakavasti kuin työvoiman tuottavuuteen ja muihin lyhyen tähtäimen kustannuslukuihin. (Liker, 2004, s. 305.)

Jokaisen yrityksen tulisi kehittää oma tapansa perustan päälle rakentaen. Toyotan itsensäkin opettaessa muille yrityksille TPS:ää se vaatii näitä yrityksiä kehittämään oman järjestelmänsä. Oivalluksien ja peruseriaatteiden lainaaminen suoraan TPS:stä on hyödyllistä, mutta ne tulisi sovittaa tuotantojärjestelmään tavalla, joka sopii yritykseen ja sen olosuhteisiin. (Liker, 2004, s. 305.)

Organisaation tulisi palkata tai kehittää Lean-johtajia ja kehittää seuraaja-järjestelmä. Johtajien tulisi tuntea perusteellisesti yrityksen tapa sekä uskoa siihen ja elää sen mukaisesti. Kaikkien johtajien tulee ymmärtää työ yksityiskohdaisesti ja tietää, miten ihmisiä sitoutetaan. Mikäli ylin johto ei aja muutosta, sitä ei tapahdu. (Liker, 2004, s. 305-306.)

Yrityksen tulisi käyttää asiantuntijoita koulutukseen ja nopeiden tulosten saamiseen. Yritys tarvitsee "sensein", eräänlaisen opettajan, joka auttaa teknisissä asioissa ja antaa johtajille neuvoja, kun he tekevät jotain ensimmäistä kertaa. Tämä opettaja auttaa muutoksen teossa, hankkii tuloksia nopeasti ja pitää yrityksen jatkuvassa liikkeessä. Tämä ei kuitenkaan aina auta, jos halutaan Lean-organisaatio. Täytyy hankkia Lean-tietämystä yrityksen sisältä tai palkata asiantuntijoita joilla on vähintään viiden vuoden Lean-kokemus, tai vastaavasti palkata ulkopuolisia asiantuntijoita konsulteiksi. Sisäinen tai ulkoinen asiantuntija voi käynnistää pikaisesti prosessin kouluttamalla henkilöstöä ja johtajia toiminnan kautta, mutta oppivan Lean-organisaation kehittämiseksi täytyy yrityksen kehittää sisäistä asiantuntemustaan johtajien, parannusasiantuntijoiden ja ryhmänvetäjien kautta, jotka uskovat Leanin filosofiaan ja ovat valmiita levittämään sitä koko organisaatioon. (Liker, 2004, s. 306.)

Kuten yllä olevasta nähdään, Bozdoganin ym. (2000) ja Likerin (2004) esitykset ovat lähestymistavoiltaan erilaisia, mutta sisältävät silti myös paljon yhtäläisyyksiä. Bozdoganin ym. (2000) malli on lähestymistavaltaan teknisempi ja se on kuvattu vaihe vaiheelta etenevänä prosessina, kun taas Liker (2004) korostaa enemmän omaa ajattelua, omaa oppimista, Lean-filosofiaa ja oman tavan kehittämistä Toyotan ideoiden ja peruseriaatteiden päälle.

2.4 Yhteenveto

Tämän luvun tarkoituksena oli kertoa, mitä Lean-ajattelulla ja sen keskeisillä periaatteilla tarkoitetaan teollisuustuotannon näkökulmasta, mistä Lean on lähöisin ja kuinka yrityksen toiminta voitaisiin transformoida Lean-periaatteiden mukaiseksi.

Luvussa esiteltiin Leanin alkuperää Ohnon (1988), Likerin (2004), Womackin ym. (1990) sekä Womackin ja Jonesin (1996, 2003) mukaan. Kirjallisuudessa Lean-ajattelun periaatteet esitetään hyvin samantyyppisesti viisi avainperiaatetta ja hukka keskeisimpinä kohtina.

Luvussa esiteltiin myös toiminnan transformointia Lean-periaatteiden mukaiseksi ja siihen liittyviä haasteita sekä transformaatio-ohjeistoa ja -malleja. Bozdoganin ym. (2000) mukaan transformaatiohaasteissa yleisenä ajatuksena nousevat esiin epävarmuuden, epätietoisuuden ja muutoksen mukaan tuomat ajatukset. Transformaatiomalleista esiteltiin Bozdogan ym. (2000) ja Liker (2004). Molempien mallien tarkoituksena on ohjata yritystä muuttamaan toimintaansa Lean-periaatteiden mukaiseksi, mutta mallien lähestymistapa on hieman erityyppinen. Karkeasti yleistäen Bozdogan ym. (2000) on suoraviivaisempi malli, joka etenee järjestelmällisesti vaihe vaiheelta. Liker (2004) on puolestaan enemmän kokoelma erilaisia ajatuksia ja ohjeistoa, jonka ympärille yrityksen tulisi itse rakentaa oma transformaatiosuunnitelmansa.

3 LEAN OHJELMISTOTUOTANNOSSA

Tässä luvussa kerrotaan, mitä tarkoitetaan Leanilla ohjelmistotuotannossa ja miten Leanin mukaiseen ohjelmistotuotantoon voidaan siirtyä. Aluksi kuvataan Lean-ajattelun muotoutumista ohjelmistotuotannon alueella. Toiseksi esitetään ohjelmistotuotantoa varten määriteltyjä Lean-periaatteita. Kolmanneksi kavennetaan näkökulmaa ohjelmistokehitykseen ja esitetään, millä tavalla Kanbanin muodossa Lean-periaatteita sovelletaan yksittäisessä ohjelmistoprojektissa. Neljänneksi kerrotaan, miten transformaatio nykyisestä ohjelmistotuotannosta voidaan tehdä Leanin mukaiseksi ja mitä hyötyjä ja kustannuksia Leanin käytössä on havaittu.

3.1 Tausta

Lean-periaatteet ovat kiehtoneet yrityksiä ja tutkijoita ympäri maailman siitä saakka, kun niitä esiteltiin ensimmäisiä kertoja TPS:n yhteydessä (Karvonen ym., 2012). Ohjelmistotuotanto ei ole poikkeus, vaan kiinnostus Leanin käyttömahdollisuuksia ja periaatteita kohtaan on kasvanut muiden alojen kanssa nopeaa tahtia (Jonsson, Larsson & Punnekkat, 2013). Vaikka Leania on käytetty ohjelmistotuotannossa 2000-luvun alusta saakka, yksiselitteinen vastaus siihen, miten Lean-ohjelmistotuotanto määritellään, on vielä vailla tarkkaa vastausta. Niin pitkään kuin Lean-ohjelmistotuotannolla ei ole yhtenäistä ja tarkkaa määritelmää, on sitä hankalaa tutkia ja verrata ketteriin menetelmiin ja ohjelmistotuotannon prosessikehitysmenetelmiin. Leania tutkittaessa on olennaista huomioda myös se, millä organisaatiotasolla Lean on otettu käyttöön organisaatiossa. Useissa organisaatioissa Leania ei sovelleta välttämättä jokaisella organisaatiotasolla. Tästä syystä mittamuksen ja arvioinnin helpottamiseksi ja yhtenäistämiseksi tulisikin määritellä tarkalleen, mitä Lean tarkoittaa ohjelmistotuotannon näkökulmasta. (Jonsson ym., 2013.)

Monille ohjelmistoyrityksille on vielä varsin epäselvää, kuinka siirtyä kohti Lean-ajattelua ja sen mukaista toimintaa. Myöskään yhtään arviointiviiteke-

hystä, joka tukisi transformaatiota kohti Lean-ohjelmistotuotantoa, ei ole tois-
taiseksi ollut saatavilla ainakaan yleisessä tiedossa. (Karvonen ym., 2012.)

Ensimmäisiä kattavampia ohjelmistotuotantoon tehtyjä tulkintoja Lean-
ajattelusta sisältyi Mary ja Tom Poppendieckin (2003) kirjaan. Tästä on tehty
myös uudistettu painos (Poppendieck & Poppendieck, 2006). Muita tunnettuja
Leania ohjelmistotuotannon näkökulmasta käsitteleviä teoksia ovat muun mu-
assa Middleton ja Joyce (2010) sekä Middleton ja Sutton (2005)

Ohjelmistotuotanto eroaa monilta osin autojen valmistuksesta, ja siksi on-
kin huomattava, ettei teollisuuden Lean-käytänteitä voi suoraan implementoida
ohjelmistotuotantoon. Useat yritykset Lean-tuotannon käytänteiden implemen-
toinnista suoraan ohjelmistotuotantoon ovat epäonnistuneet näiden alojen eri-
laisen luonteen takia. (Poppendieck & Poppendieck, 2003, s. 15.)

Poppendieckin ja Poppendieckin (2003, s. 15) mukaan ohjelmistotuotanto
ei ole tuotantoprosessi vaan kehitysprosessi. Näiden kahden eroa voidaan ku-
vata vertaamalla ruuan valmistukseen. Kehitys vastaa ruokareseptin tekemistä.
Tuotanto on puolestaan jo valmiin reseptin ohjeiden seuraamista, kuten teolli-
suudessaakin jo valmiin tuotteen monistamista. Reseptin tai ohjelmiston kehit-
täminen on oppimisprosessi, johon sisältyy kokeilua ja virheitä, mutta näin te-
kemällä pyritään saavuttamaan lopputuloksena paras mahdollinen ateria tai
ohjelmistotuote. (Poppendieck & Poppendieck, 2003, s. 15.)

Aivan kuten teollisuustuotannossakin, niin myös ohjelmistotuotannossa
esiintyy hukkaa. Kuusela ym. (2011) määrittelevät Poppendieckin ja Poppen-
dieckin (2003, 2006) esitykseen perustuen ohjelmistotuotannon seitsemän eri-
tyyppistä hukkaa seuraavasti:

- *Osittain tehty työ* (Partially done work): Osittain tehty työ on jotain mikä
ei ole tehty vielä valmiiksi. Esimerkiksi testaamaton koodi, dokumen-
toimattomat tai ei-ylläpidetyt liiketoiminta päätökset.
- *Ylimääräiset ominaisuudet* (Extra features): Jotain mitä ei ole oikeasti vaa-
dittu tai mitä ei lopulta tarvita.
- *Uudelleen oppiminen tai tiedon hukka* (Relearning / a waste of knowledge):
Esimerkiksi aiempien päätösten unohtaminen tai jo aiemmin yritettyjen
ja epäonnistuneiden ratkaisujen uudelleen kokeileminen. Tiedon hukkaa
on myös kyvyttömyys hyödyntää muiden ihmisten tietoa.
- *Tuotteen luovutukset* (Handoffs): Tiedon / työn siirtäminen toiselle henki-
lölle joko kesken prosessin tai heidän siirtyessään prosessista toiseen.
Luovutus voi aiheuttaa hiljaisen tiedon katoamista ja tiedon vääristymis-
tä.
- *Tehtävien vaihdot* (Task switching): Tehtävien vaihdot kesken prosessin
esimerkiksi useaa projektia yhtäaikaaisesti toteutettaessa.
- *Viivästykset* (Delays): Jonkin ei-valmiin tai toimimattoman tuotoksen
odottaminen johtaa viivästyksiin.
- *Viat* (Defects): Jokin jo kehitetty ominaisuus tai toiminto joka ei täytä an-
nettua tavoitteita tai joka on jotain muuta kuin sen piti olla. Esimerkkei-

nä tällaisesta ovat ohjelmavirheet ja väärin toteutetut liiketoimintavaatimukset.

Petersen ja Wohlin (2010b) näkevät Middletonin (2001) esitykseen perustuen osittain tehdyn työn eli varaston olevan erityisen kriittiseksi. Petersenin ja Wohlinin (2010b) mukaan syynä tähän ei ole niinkään se, että ohjelmistotuotteet veisivät paljon varastotilaa, vaan seuraavat syyt:

- Varasto piilottaa viat, jotka löytyvät vasta prosessin myöhemmissä vaiheissa (Middleton, 2001).
- Varastossa oleviin tuotteisiin on käytetty aikaa esimerkiksi vaatimusten määrittelyssä ja arvioinnissa ja kontekstin muuttuessa nämä vaatimukset muuttuvat turhiksi, jolloin niihin käytetty aika on täysin hyödytöntä (Petersen, Wohlin, Baca, 2009).
- Varasto vaikuttaa muihin jätteisiin. Esimerkiksi suuri varasto aiheuttaa odotusaikoja. Tämä esiintyy esimerkiksi vesiputousmallin mukaisessa kehittämisessä, jossa suunnittelijat joutuvat odottamaan vaatimusedokumentaation hyväksyntää (Petersen ym., 2009). Pitkät odotusajat sisältävät myös riskin siitä, että jo valmiiksi saatu työ muuttuu tarpeettomaksi. Lisäksi suuri varasto vaatimusmäärittelyssä voi johtua siitä, että vaatimukseen on sisällytetty suuri määrä ylimääräisiä ominaisuuksia.
- Varasto hidastaa koko kehitysprosessia moottoritien tavoin. Moottoritien nopeudet tippuvat sitä alemmas, mitä enemmän liikennettä on.
- Suuri varasto aiheuttaa räsitystä organisaatiossa (Morgan, 1998).

3.2 Lean-periaatteita

Hibbsin, Jewettin ja Sullivanin (2009, s.16) mukaan Poppendieckien kirjat (2003, 2006) olivat ensimmäiset kattavammat perusteokset aiheesta, joita useat kirjoittajat ovat käyttäneet pohjana omissa teoksissaan. Koska näitä teoksia pidetään kattavina perusteoksina aiheesta, tarkastelemme tässä Lean-ohjelmistokehityksen periaatteita heidän mukaansa. Poppendieck ja Poppendieck (2006, s. 23-24) määrittelevät seitsemän Lean-ohjelmistotuotantoon soveltuva perusperiaatetta, jotka kuvataan alla tarkemmin.

Poista hukka (Eliminate waste). Aivan kuten teollisuuden puolella, kaikki mikä ei tuota asiakkaalle arvoa, tulisi pystyä poistamaan mahdollisimman tarkkaan myös ohjelmistotuotannossa. Ohjelmistotuotannossa hukkaa ilmenee muun muassa osittain tehdyssä työssä ja ylimääräisten ominaisuuksien kehittämisessä. Laskelmien mukaan normaali käyttäjä käyttää vain noin kahtakymmentä prosenttia ohjelmiston ominaisuuksista ja toiminnosta säännöllisesti. Arviolta kaksi kolmasosaa ohjelmistojen ominaisuuksista ja toiminnoista on todella harvoin käytettyjä. Näistä osin jopa täysin turhista ominaisuuksista ja toiminnoista aiheutuu koodin kompleksisuuden kasvua ja kertyy tuotannolle valtavat suhteelliset kulut niin ajassa kuin rahassakin. Nyrkkisääntönä voidaan

pitää, että 20 % koodista tulisi tuottaa 80 % asiakkaan kokemasta arvosta ja vasta tämän osuuden toteuttamisen jälkeen tulisi siirtyä seuraavaksi tärkeimpiin ominaisuuksiin. (Poppendieck & Poppendieck, 2006, s. 23-24.)

Rakenna laatu ohjelman sisälle jo alussa (Build quality in). Usein muissa kehitysmenetelmissä virheiden jäljitys ja niiden korjaus toteutetaan vasta testauksen yhteydessä. Leanin mukaisessa ohjelmistotuotannossa virheiden tunnistus ja korjaus tulisi toteuttaa välittömästi jo kehitysvaiheessa, ilman että niihin tarvitsi puuttua enää myöhemmin testauksen yhteydessä. Perimmäisenä ideana on siis sitoa kehitys ja testaus yhteen siten, että kehitettyä koodia testataan usein, vaikka tunnin tai parin välein, ja edetään vasta kun tämän osalta testaus on läpäisty. Viikon lopussa aikaansaatu ja kehityksen yhteydessä tehdyt testit läpäissyt koodi siirretään tarkempaan testaukseen. (Poppendieck & Poppendieck, 2006, s. 25-26.)

Luo tietoa (Create knowledge). Ohjelmistotuotanto on tietoa luova prosessi, mikä aiheuttaa omat haasteensa suunnittelun ja varsinaisen kehityksen välille. Vaikka yleinen arkkitehtuurikonsepti ohjelmistolle luodaan etukäteen, sen virallinen hyväksyntä tapahtuu vasta koodaamisen yhteydessä. Käytännössä yksityiskohtainen ohjelmiston suunnittelu tapahtuu vasta koodaamisen yhteydessä, vaikka se olisi dokumentoitu jo etukäteen. Etukäteen tehty tarkka suunnittelu ei voi ottaa huomioon kompleksisuuden mukanaan tuomia haasteita tai itse ohjelmistokehityksen mukanaan tuomaa palautetta. Lean-organisaatio tietää, että sen tulee jatkuvasti kehittää prosessejaan, koska se toimii kompleksisessa ympäristössä, jossa ongelmia tulee aina. Kehitysprosessi, joka on keskittynyt tiedon luomiseen, osaa odottaa suunnitelmien muuttuvan, eikä tuhlaa aikaa turhiin oletuksiin ja ennustuksiin, jotka eivät välttämättä tule toteutumaan. (Poppendieck & Poppendieck, 2006, s. 29-30.)

Lykkää sitoutumista (Defer commitment). Ensiaputyöntekijät koulutetaan toimimaan paineen alla haastavissa, arvaamattomissa ja usein vaarallisissakin tilanteissa. He toimivat tiettyjen aikaikkunoiden sisällä tehdessään päätöksiä, siten että kriittiset päätökset pyritään lykkäämään mahdollisimman lähelle aikaikkunan loppua, jotta heillä olisi tässä vaiheessa käytössään mahdollisimman paljon ja mahdollisimman tarkkaa tietoa. Lean-ohjelmistotuotannossa tulisi peruuttamattomien päätösten osalta soveltaa samaa ideaa. Huomattavaa on kuitenkin, että samaa menetelmää ei tule käyttää kaikissa päätöksissä, ja ensisijaisista olisikin pyrkiä tekemään mahdollisimman monista päätöksistä peruutettavia. Täten tilanteen muuttuessa siihen pystytään vastaamaan tarvittavilla muutoksilla mahdollisimman nopeasti. Esimerkkeinä tällaisista päätöksistä ovat muun muassa kehitettävän järjestelmän aikaisessa vaiheessa määritetyt ominaisuudet. Niiden yhteydessä tulisi välttää päätöksiä, jotka lukitsevat järjestelmän kriittisten osien suunnitelmat, joita on hankalaa ja sekä ajallisesti että rahallisesti kallista korjata myöhemmin. (Poppendieck & Poppendieck, 2006, s. 32-33.)

Toimita nopeasti (Deliver fast). Yritykset, jotka pystyvät tuottamaan nopeasti, ovat onnistuneet poistamaan suuren määrän hukkaa prosesseistaan. Hukka taas maksaa yrityksille rahaa. Nopealla tuotannolla on etunsa myös siinä mielessä, että mikäli asiakkaan tilaama ominaisuus saadaan tuotettua nopeasti,

hän ei ehdi muuttaa mieltään kesken tuotantoprosessin. Tällainen muutos johdaisi tuotannon osalta turhaan työhön ja kasvaneisiin kuluihin. (Poppendieck & Poppendieck, 2006, s. 34.)

Kunnioita ihmisiä (Respect people). Ihmiset pitävät menestyvien tuotteiden parissa työskentelystä, ja menestyneiden tuotteiden taustalta löytyy usein loistavia johtajia. Yritys, joka kunnioittaa ihmisiä, kehittää hyviä johtajia, jotka saavat tiimensä jäsenet sitoutumaan, ajattelemaan ja keskittymään loistavan tuotteen luomiseen. Jollakin erityisalalla toimivan yrityksen tulisi itse kehittää teknistä asiantuntemustaan parantaakseen kilpailuetuaan muihin nähden. Yrityksen ostaessa osaamista ulkopuolelta tähän samaan ostettuun osaamiseen pääsevät käsiksi myös kilpailijat. Yritykset, jotka eivät näe tarvetta osaamisen kartuttamiselle, ovat näistä kolmesta kaikkein heikoimmassa asemassa ilman kilpailuetua osaamisen kohdalla. Ihmisten kunnioittaminen tarkoittaa myös sitä, että tiimeille annetaan yleissuunnitelmat ja järkevät tavoitteet ja luotetaan siihen, että he itseohjautuvasti saavuttavat tavoitteet. Suorien ja tarkkojen käskyjen sijaan kehitetään refleksiivinen organisaatio, jossa ihmiset ajattelevat ja toimivat itse. (Poppendieck & Poppendieck, 2006, s. 36-37.)

Optimoi kokonaisuus (Optimize the whole). Ohjelmistotuotannolla on pitkään ollut taipumus osaoptimointiin (suboptimization), jossa keskitytään vain pieniin osiin pilkottuihin palasiin ja niiden seuraamiseen. Ongelmana tässä on se, että usein näin toimimalla kokonaisuus jää vaille huomiota ja lähes takuuvarmasti koko arvovirta kärsii. Lean keskittyy optimoimaan kokonaisuutta, jossa huomioidaan koko arvovirta. Esimerkkinä tällaisesta on kahden osaston välinen vastuunsiirto tuotannossa. Molemmat osastot voivat pelkästään omaa tekemistään mittaamalla näyttää mittareiden mukaan hyviltä ja tehokkailta. Kuitenkin tuotannon siirtyessä osastolta toiselle, voi juuri tämä osastojen välinen siirto olla ongelmakohta. Ellei kukaan ole vastuussa siirrosta osastojen välillä tai vastaamassa asiakkaalle, eli optimoimassa kokonaisuutta, ei tässä siirron yhteydessä syntyvää hukkaa välttämättä huomata. (Poppendieck & Poppendieck, 2006, s. 38-39.)

Luvussa 2 on kuvattu teollisuuden osalta Lean-ajattelua ja Lean-tuotantoa. Verrattaessa Poppendieckin ja Poppendieckin (2003, 2006) teoksien Lean-ohjelmistotuotannon seitsemää peruseriaatetta Womackin ja Jonesin (2003) ja Likerin (2004) näkemyksiin ja periaatteisiin on nähtävissä selkeitä yhtäläisyyksiä. Poppendieck ja Poppendieck (2003, 2006) on ikään kuin ohjelmistotuotannon näkökulmasta muunneltu yhdistelmä Womackin ja Jonesin (2003) länsimaisempaa, teknisempää ja prosessikeskeisempää näkemystä sekä Likerin (2004), Ohnon (1988) ja Toyotan alkuperäistä japanilaisempaa ajattelua, oppimista ja filosofiaa korostavaa näkemystä.

Leania voidaan lähestyä ohjelmistotuotannossa myös suoraan Womackin ja Jonesin (2003) viiden Lean-ajattelun peruseriaatteen kautta. Middleton ja Sutton (2005) käyttävät tätä tapaa Lean-ohjelmistokehitystä käsittelevässä teokseensa *Lean Software Strategies*.

3.3 Kanban Leanin mukaisessa ohjelmistokehityksessä

Edellä on tarkasteltu Leania kokonaisvaltaisena ajattelutapana, jota voidaan soveltaa koko ohjelmistotuotantoon. Seuraavaksi kerrotaan, miten Lean-periaatteita voidaan soveltaa ohjelmistokehityksessä yksittäisen ohjelmiston yhteydessä. Nämä periaatteet on koottu ja konkretisoitu Kanban-nimiseen toimintatapaan. Seuraavassa kerrotaan ensin Kanbanin periaatteista ja sen jälkeen hyödyistä ja haasteista.

3.3.1 Kanban-periaatteita

Kanbania ja sen käyttöä ohjelmistotuotannossa käsittelevistä teoksista tunnetuimpia ovat Anderson (2010) sekä Kniberg ja Skarin (2010). Andersonin (2010) mukaan Kanban on monimutkainen ja mukautuva järjestelmä, joka on tarkoitettu kiihdyttämään organisaation muutosta Leanin mukaiseksi. Tämän kaltaiset monimutkaiset ja mukautuvat järjestelmät edellyttävät alkuehtoja ja yksinkertaisia sääntöjä, joiden avulla voidaan saavuttaa monimutkaisempaa, mukautuvampaa ja muotoutuvaa käyttäytymistä. (Anderson, 2010, s.15.)

Andersonin (2010, s. 15) mukaan Kanban käyttää viittä ydinominaisuutta, joiden avulla voidaan saavuttaa Lean-ajattelun mukainen käyttäytyminen organisaatiossa. Nämä viisi ydinominaisuutta ovat olleet mukana kaikissa onnistuneissa Kanban-toteutuksissa (Anderson, 2010, s.15):

1. Visualisoi työn virtaus
2. Rajoita käynnissä olevien töiden määrää
3. Mittaa ja johda virtausta
4. Tee prosessikäytännöistä täsmällisiä
5. Käytä kehitysmahdollisuuksien tunnistamiseen olemassa olevia malleja

Andersonin (2010, s. 16) mukaan Kanban ei ole ohjelmistokehityksen elinkaarimenetelmä tai projektinhallinnallinen lähestymistapa. Kanban vaatii että organisaatiossa on jo olemassa oleva prosessi johon Kanbania voidaan soveltaa. Kanban ei pakota tiimiä omaksumaan käyttöönsä mitään valmiiksi määritettyä menetelmää tai prosessimallia, vaan se toimii niin sanottuna luvan antajana (permission giver) sille, että tiimit kehittäisivät omia prosessiratkaisuja ja työkaluja. Tämän kaltainen lähestyminen inkrementaaliseen muutokseen on aiheuttanut kiistanalaista keskustelua ketterien ohjelmistokehitysmenetelmien yhteisössä. (Anderson, 2010, s. 16.)

Knibergin ja Skarinin (2010, s. 4-5) mukaan Kanban sisältää kolme sääntöä. Ensimmäinen sääntö on *"Visualisoi työn virtaus"*, jonka mukaan työ tulisi jakaa pienempiin osiin ja kirjoittaa jokaiselle työn osalle kortti joka asetetaan seinälle nähtäväksi. Tässä yhteydessä tulisi käyttää työvaiheiden mukaan nimettyjä sarakkeita kuvaamaan vaihetta jossa työn osio on sillä hetkellä. Toinen sääntö on *"Käynnissä olevan työmäärän rajoittaminen"*, jossa määritetään selkeät rajat

sille, miten monta työn osiota on yhtäaikaisesti toteutuksessa yhtä työvaihetta kohti. Kolmas sääntö on ”*Työn osan läpimenoajan mittaaminen*”. Mittauksen kohteena on keskimääräinen aika, joka yhdellä työn osalla menee koko prosessin alusta loppuun saakka. Läpimenoaikojen mittaamisella saadaan optimoituja koko prosessia siten, että läpimenoaika pienenee ja työnkulun ennustaminen helpottuu. (Kniberg & Skarin, 2010, s. 4-5.)

Knibergin ja Skarinin (2010, s. 7) mukaan Kanban on prosessityökalu, kuten Scrumkin, sillä se auttaa työskentelemään tehokkaammin tiettyyn pisteeseen saakka kertomalla tekijälle mitä tehdä. Kniberg ja Skarin (2010, s. 8) painottavat kuitenkin sitä, että mikään työkalu ei ole täydellinen. Työkalut eivät kerro kaikkea, mitä tulee tehdä, vaan ne antavat tiettyjä rajoitteita ja ohjesääntöjä. Kanban sallii käyttäjälleen vapauden toimia mieleisellään tavalla niin pitkään, kunhan työn virtaus on visualisoitu ja käynnissä olevien töiden määrä on rajoitettu. Työkaluja tulisi rohkeasti kokeilla ja täten selvittää ne parhaimmat käytänteet ja työkalujen yhdistelmät, mitkä toimivat tiimissä parhaiten. Esimerkkinä tästä monissa Kanbania käyttävissä tiimeissä pidetään päivittäisiä kokouksia (Stand-up meetings), jotka ovat Scrumista tuttu käytäntö. Kanban ei rajoita muiden työkalujen käyttöä. (Kniberg & Skarin, 2010, s. 7-10.)

Andersonin (2010) ja Knibergin ja Skarinin (2010) näkemykset Kanbanista eroavat selkeästi jonkin verran toisistaan. Anderssonin (2010) näkemyksen mukaan Kanban on monimutkainen ja mukautuva järjestelmä, jolla organisaatio voi päästä lähemmäksi Leanin mukaista toimintaa. Hänen mukaansa Kanban ei ole projektinhallinnallinen lähestymistapa tai menetelmä, vaan pikemminkin kannustin siihen, että tiimit voivat kehittää prosessiratkaisuja ja työkaluja tarpeidensa mukaan. Knibergin ja Skarinin (2010) mukaan Kanban on selkeästi prosessityökalu, joka auttaa kertomalla mitä tehdä. Knibergin ja Skarinin (2010) näkemyksessä Kanbanin ainoat rajoitteet ovat käynnissä olevan työn määrä sekä työn kulun visualisointi.

Andersonin (2010) viittä ydinominaisuutta sekä Knibergin ja Skarinin (2010) kolmea pääsääntöä verrattaessa molemmista voidaan nähdä selkeinä yhtäläisyyksinä *työn virtauksen visualisointi* sekä *käynnissä olevan työn määrän rajoittaminen*. Andersonin (2010) näkemyksessä kolmantena kohtana on mainittu *mittaa ja johda työn virtausta* sekä Knibergin ja Skarinin (2010) näkemyksessä *työn osan läpimenoajan mittaaminen*, joissa voidaan nähdä ainakin osittaisia yhtäläisyyksiä prosessin optimoinnissa työn virtauksen mittaamiseen kautta.

Kanban korostaa visualisointia pullonkaulojen ja muiden asioiden havaitsemiseksi ja virtauksen maksimoimiseksi. Yksi Kanbanin avainkäsitteitä on se, että koko kehityksen tulee olla optimoituja. Itse prosessi perustuu useisiin parametreihin ja näin ollen kokonaisvaltainen optimointi on haasteellista. Tämän lisäksi parametrit eivät ole kiinteitä, vaan riippuvat useista tekijöistä, kuten kehittäjien määrästä sekä heidän kokemuksestaan, tiimin sisäisistä käytänteistä sekä implementoitavien ominaisuuksien määrästä ja niiden vaatimasta työmäärästä. Nämä tekijät eivät eroa ainoastaan eri projektien välillä, vaan voivat vaihdella myös yksittäisen projektin aikana. Näiden syiden takia etukäteen suunniteltua ”yksi malli sopii kaikkiin” -tyyppistä lähestymistä ei voida käyttää

tutkimuksissa tai prosessiparametrien parantamisessa. Tämän kaltaisessa ongelmassa ohjelmistoprosessin simulointi voi olla hyödyllistä. (Anderson, Concas, Lunesu & Marchesi, 2011.)

Jätteen eliminointiin ja arvon lisäämiseen liittyvä imujärjestelmä (pull system), joka kertoo, miten työ virtaa, on oleellinen käsite, jossa prosessit perustuvat asiakkaan esittämiin vaatimuksiin ominaisuuksista ja toiminnallisuuksista. Työn alla oleva työ (Work in process / WIP) kuvataan yleensä tiimille ja sidosryhmille Kanban-taulua käyttäen. Yleisesti voidaan määritellä Kanban ohjelmistoprosessin olevan WIP-sidonnainen imujärjestelmä, joka on kuvattu Kanban-taululla. (Anderson ym., 2011.)

Kanban-taulun oikeanlainen käyttö helpottaa minimoimaan WIP-määrää, korostaa rajoitteita ja auttaa koordinoimaan tiimityötä. Tulee kuitenkin huomata, että Lean on enemmän kuin pelkkä Kanban, ja tästä syystä tulisikin käyttää useampia Lean-käytänteitä yhdessä Kanbanin kanssa, jotta saavutettaisiin täysi hyöty Leanin soveltamisesta ohjelmistotuotannossa. (Anderson ym., 2011.)

Kanban on organisaation johdon osallistumista edellyttävä lähestymistapa. Tällä tarkoitetaan sitä, että organisaation johdon tulee osallistua keskusteluun siitä, kuinka työtä suoritetaan ja miten sitä seurataan. Tämä on tärkeää siksi, että näin toimimalla organisaation johto ei voi vaatia kehittäjiä vain tekemään enemmän. Tämän sijaan organisaation johto sitoutuu noudattamaan tiimin työhönsä valitsemia käytänteitä ja menetelmiä. Kanbanin avulla saadaan koko tiimin prosesseista läpinäkyviä ja näin ollen organisaation johto voi osallistua prosessien parantamiseen yhdessä tiimin kanssa. (Shalloway ym., 2009, 98-99.)

Kanban-taulu on joko fyysinen tai elektroninen taulu, jossa esitetään koko kehitysprojekti vaiheittain, työnvirtauksen suunta sekä ilmoitetaan vaiheittaiset WIP-rajoitteet. Kuviossa 2 olevat vaiheet ovat rästissä, kehitys, testaus, julkaisu ja valmis. Nämä vaiheet ovat esimerkkejä vaiheiden jaosta ja vaiheet tulisikin valita aina tiimin tarpeen mukaisesti. Vaiheita on tarpeen vaatiessa mahdollista jakaa myös omiin alivaiheisiin, esimerkiksi kehitysvaihe voi sisältää alivaiheet meneillään ja valmis. (Anderson, 2010; Kniberg & Skarin, 2010.)

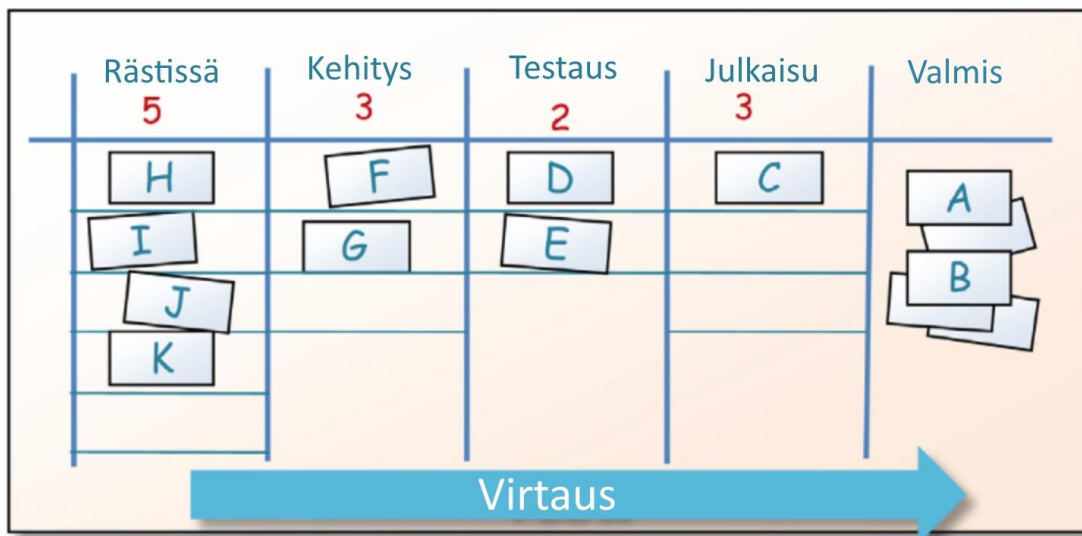
Työn osat Kanban-taulussa esitetään pienillä Kanban-korteilla. Mikäli käytössä on fyysinen taulu, voidaan Kanban-korttina käyttää esimerkiksi helposti tauluun liimattavia Post-it viestilappuja. Jokaisella työn osalla on oma Kanban-korttinsa, joita kuviossa 2 kuvataan aakkosin merkatuilla lapuilla. Työn osat etenevät taulussa virtauksen esittämään suuntaan. Kuviossa 2 virtaus esitetään yleisimmin esitetystä muodosta eli vasemmalta oikealle. Alkuvaiheessa työn osat kirjataan tehtävälisalle eli rästilistalle (backlog), josta ne aina WIP-rajoitteiden ja vapaan kapasiteetin salliessa "imetään", ja sitä kuvastava Kanban-kortti siirretään taululla, ensimmäiselle vaiheelle joka kuviossa 2 on kehitys. Työn osan valmistuttua kehitysvaiheesta se "imetään" seuraavaan vaiheeseen joka kuviossa 2 on testaus. Työn osan valmistuttua viimeisessä vaiheessa, se "imetään" sieltä automaattisesti valmiiden (live) työn osien listalle.

Kehitystiimin jokaisella kehittäjällä on taidot liittyen useisiin työn vaatimiin vaiheisiin. Kehittäjän saatua työn osan valmiiksi hän katsoo uuden vaadi-

tun työn osan Kanban-työkalulta kehitettäväksi. Kanban-järjestelmä keskittyy jatkuvaan työn virtaukseen, eikä yleensä sisällä kiinteitä iteraatioita. Työ toimitetaan heti, kun se on valmis, ja tiimi työskentelee ainoastaan muutamien työn osien parissa kerrallaan. Näin saadaan taattua WIP-määrän pysyminen rajoissa ja saavutetaan jatkuva valmiiden työn osien virtaus koko kehitysprosessin läpi. (Anderson ym., 2011.)

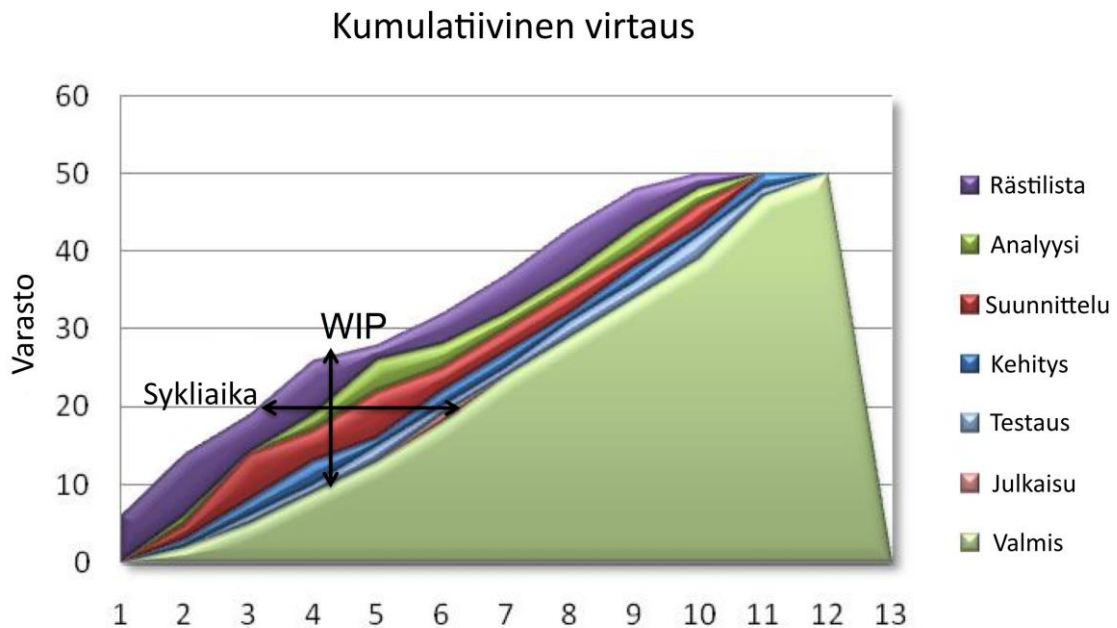
Kanban-työkalun voidaan nähdä pyrkivän täyttämään Andersonin (2010) ja Knibergin ja Skarinin (2010) näkemyksissä esittämät kolme yhtenevää Kanban-periaatetta, joita ovat *työn virtauksen visualisointi, käynnissä olevan työn määrän rajoittaminen ja työn osan läpimenoajan mittaaminen / mittaa ja johda virtausta*.

Kanban-työkalun oikeanlainen käyttö helpottaa minimoimaan WIP-määrää, korostaa rajoitteita ja auttaa koordinoimaan tiimityötä. Tulee kuitenkin huomata, että Lean on enemmän kuin pelkkä Kanban, ja tästä syystä tulisikin käyttää useampia Lean-käytänteitä yhdessä Kanbanin kanssa, jotta saavutettaisiin täysi hyöty Leanin soveltamisesta ohjelmistotuotannossa. (Andersson ym., 2011.) Seuravassa esitetään esimerkki Kanban-työkalusta (kuvio 2).



KUVIO 2 Kanban-työkalu (Kniberg & Skarin, 2010, 5)

Toinen kuvaaja, jota käytetään usein Kanbanin yhteydessä on kumulatiivinen virtauskaavio (CFD), joka kuvaa kokonaisvirtausta Kanban-järjestelmän läpi. CFD tarjoaa mittarit jokaiselle merkittävälle vaiheelle työvirtauksessa. Näitä vaiheita voivat olla esimerkiksi rästilista, analyysi, suunnittelu, kehitys, testaus, julkaisu ja valmis kuten kuviossa 3. CFD:ssä horisontaalisella akselilla kuvataan sykliä ja vertikaalisella akselilla varastomääriä. Jokaista vaihetta kohti CFD näyttää työn osien määrän sillä hetkellä. CFD:ssä paksut linjat, jotka kertovat suuresta yhtäaikaista työn osien määrästä vaihetta kohti, kuvaavat esteitä tai tukoksia virtauksessa. Vastaavasti ohuet linjat CFD:ssä osoittavat, että WIP-määrä on liian pieni, eikä työn osia pakkaudu vaiheeseen ja vaihetta hoitavilla henkilöillä olisi vapaata kapasiteettia useammankin työn osan yhtäaikaiseen käsittelyyn. (Shalloway ym., 2009, s.99.) Kuviossa 3 esitetään esimerkki kumulatiivisesta virtauskaaviosta.



KUVIO 3 Kumulatiivinen virtauskaavio (Scotland, 2009, 57)

Andersonin (2010) sekä Knibergin ja Skarinin (2010) Kanban-näkemyksissä kolmas yhtenäinen kohta on *mittaa ja johda työn virtausta / työn osan läpimenoajan mittaaminen*. Tähän liittyen Kanban tarjoaa käyttöön mittareita toiminnan mitaamiseen. Tilauksesta toimitukseen kestävä aika eli sykli aika (cycle time) on tärkeä mittari, joka kuvastaa kokonaisaikaa siitä hetkestä, kun päätetään kehittää ominaisuus, aina siihen hetkeen, kun se on saatu julkaistua. Tämä mittari kuvastaa hyvin sitä, miten nopeasti, toistuvasti ja luotettavasti ohjelmistoa toimitetaan asiakkaille. (Kerzazi & Robillard, 2013.)

Koska yhtä tiettyä kaikille sopivaa Kanban-mallia ei ole tarjolla, myös sen yhteydessä käytetyt mittarit tulisi sovittaa aina projektin tai prosessin mukaiseksi. Organisaatiossa mahdollisesti jo käytössä olevat mittarit tulisi arvioida uudelleen Kanbanin käyttöönotossa ja miettiä tukevatko ne jatkuvaa parantamista ja muita Kanbanin ja Leanin mukaisia periaatteita. (Seikola, Loisa & Jago, 2011.)

Seikolan ym. (2011) mukaan sykli ajan mittaamisen lisäksi muita Kanbanin yhteydessä käytettäviä mittareita voivat olla esimerkiksi seuraavat:

- Jonotusaika rästilistalla (työn osien prioriteetteja, keskiarvoista määrää ja levikkiä kohti)
- Sykli aika tilauksesta toimitukseen (työn osien prioriteetteja, keskiarvoista määrää ja levikkiä kohti)
- Myöhästymisprosentti (työn osien prioriteettia kohti)
- Sisään- ja ulosvirtaukset
- Määrä ja prosentuaalinen osuus tapauksia jokaisessa taulun sarakkeessa

3.3.2 Kanbanin hyödyt ohjelmistotuotannossa

Useat ketterien kehittämisen tiimit käyttävät kymmenestä kahteenkymmeneen prosenttiin ajastaan haluttujen ominaisuuksien muokkaamiseen tarinoiksi ja niiden työmäärän estimointiin. Joskus tätä voidaan pitää hyödyllisenä riittävän ymmärryksen hankkimiseksi, mutta jos tarinan kuvausta tarvitaan ainoastaan tarinaa vastaavan tehtävän mahduttamiseen keinotekoiseen aikarajaan (time-box), voidaan tehtyä työtä pitää turhana ajan hukkana. Kun tämä yhdistetään vielä pienempien tarinoiden estimoinnin aikaansaamiin kustannuksiin, on saatu aikaan jo paljon kustannuksia ja tehtyä työtä ilman ylimääräistä arvoa. Kanban eliminoi tämän tyyppisen jätteen hallitsemalla virtausta aikalaatikoiden sijaan. (Shalloway ym., 2009, s. 100.)

Kanbanissa ei oleteta, että kaikki etukäteen tehtävä arviointi on täysin tarpeetonta, mutta se ehdottaa kiinnittämään huomiota saavutettavaan arvoon suhteessa tehtävään sijoitettuun aikaan. Mikäli ominaisuuksiin tarvittavan ajan arviointi toimii organisaatiossa jo hyvin, voidaan tulkita, että yksityiskohtaiset arviot eivät ole kovinkaan välttämättömiä. (Shalloway ym., 2009, s.100.)

Kanbanin todellinen arvo perustuu sen vaatimukseen siitä, että tiimi luo työvirtauksen selvästi määritellyillä säännöillä ja rajoilla. Tämä mahdollistaa tiimin jäsenten keskustella siitä, mikä toimii ja mikä ei. Näin ollen Kanban auttaa tiimiä keskittymään prosessiin ennemmin kuin syyttämään yksittäisiä henkilöitä. Yksittäinen henkilö voi tehdä virheitä, mutta syyttämisen sijaan tärkeämpää on keskittyä siihen, mikä prosessissa mahdollisti virheen tapahtumisen tai sen, ettei sitä havaittu. Edellä mainitussa tilanteessa tulee tutkia tapahtunutta ja korjata prosessia vastaavan virheen välttämiseksi tulevaisuudessa. (Shalloway ym., 2009, s.100.)

Shalloway ym. (2009, s.100) mukaan aiemmat tutkimukset todistavat, että tiimit oppivat jatkuvaa prosessien parantamista nopeammin Kanbanin avulla muiden muassa seuraavista syistä:

- Kanban vähentää sitoutumispelkoa tarinakohtaisiin arvioihin, joka on merkittävä riski monille tiimeille. Pelko heikentää aina oppimista.
- Kanban on selkeästi tiimiprosessi, sen sijaan että se olisi yksilöille suunnattu. Kanban korostaa koko tiimin suorituskykyä ennemmin kuin yksilötason suorituksia sekä auttaa vähentämään häpeän pelkoa.
- Kanban keskittyy siihen, kuinka työvirtausprosessia voidaan parantaa sen sijaan, että syytettäisiin yhtä yksittäistä työntekijää.
- Kanban sallii konkreettisten mittareiden reflektointien, kuten ”Pitäisikö WIP-määrän olla 4 vai 5?” Alussa konkreettisten asioiden reflektointi on yleensä helpompaa kuin abstraktien tai henkilökohtaisten asioiden reflektointi.
- Läpinäkyvä prosessi sallii johdon osallistumisen prosessin kehittämiseen.

Seikola ym. (2011) raportoivat kahdessa Ericssonin tutkimus- ja tuotekehitysyksikössä tekemässään tapaustutkimuksessa samantyyppisiä tuloksia Kanbanin käyttöönoton aikaansaamista hyödyistä. Keskeisimpinä positiivisina tuloksina voidaan nähdä seuraavat kohdat: selkeää parannusta tiimityössä, voimaantumisen ja vastuunottamisessa, työssä on siirrytty työntöohjauksesta imuohjaukseen, organisaatiossa haastetaan mukavuusalueelta poistumista, osastojen välinen yhteistyö on parantunut ja tiimeissä on omaksuttu käyttöön ristikkäistoiminnallisuudet, jotka mahdollistavat esimerkiksi viankorjaamisen ja analysoinnin samanaikaisesti. Lisäksi koko organisaation ajattelutapa on suuntautunut kohti jatkuvaa parantamista, mikä mahdollistaa kenen tahansa organisaation jäsenen ilmaista huoliaan ja etsiä parannuksia toimintaan. Toiminnasta on tehty myös huomattavasti läpinäkyvämpää kaikille sidosryhmille, jolloin esimerkiksi kaikki ovat tietoisia korkeimman prioriteetin tehtävistä ja yleisimmistä ongelmista asiakasverkostoissa. Lisäksi kehityksen pullonkaulat ovat helposti havaittavissa Kanban-tauluilta. (Seikola ym., 2011.)

3.3.3 Kanbanin haasteet ohjelmistokehityksessä

Kanbanin käyttöönotossa ja sen käytössä on havaittu positiivisten vaikutusten ohella myös haasteita. Esimerkiksi muutos yksilöiden ryhmästä oikeaksi tiimiksi on aikaa vievä prosessi. Sosiaalisen käyttäytymisen ja työmäärän tasapainottamisen lisäksi tulee huomata, että käytössä olevat ohjelmistotyökalut voivat olla alun perin suunniteltu tukemaan yksilönä työskentelyä. (Seikola ym., 2011.)

Yhtenä suurena haasteena on vakavien ja kiireisten muutostarpeiden ennalta arvaamaton ilmaantuminen. Näin ollen tehtävien vaihtoja ei voida aina kokonaan välttää. On myös huomattava, että joskus osa tiimin jäsenistä voi joutua matkustamaan paikan päälle. Näin ollen lopun tiimin on kyettävä tukemaan toisiaan esimerkiksi rästilistalta seuraavien tehtävien valinnassa. (Seikola ym., 2011.)

Organisaatiossa on mahdollisesti aiemmin ollut käytössä johdon määräämä työntöohjaukseen (push) ja jatkotoimenpiteisiin perustuva prosessimalli. Uudessa Kanbanin mukaisessa, imuohjaukseen perustuvassa mallissa tiimillä on vastuu suoritettavista työtehtävistä. Muutos kohti tiimityöskentelyä ja sen vaatimaa ajatusmaailmaa ja vastuunottamista on usein aikaa vievää. (Seikola ym., 2011.)

Työntöohjauksesta imuohjaukseen siirtyminen on haastavaa myös organisaation johdolle. Varsinkin Kanbanin käyttöönoton alkuvaiheissa kasvava rästilista voi herättää johdossa huolta. Ensimmäisten rästilistapiikkien jälkeen usko konseptin toimivuuteen helpottaa. Samalla kasvaa luottamus siitä, että rästilistalle merkatut tehtävät tulevat hoidettua kunnolla, vaikka ne viipyisivätkin listalla hetken odottamassa vuoroaan. (Seikola ym., 2011.)

Aiemmin organisaation johto on pysynyt ajan tasalla tasaisin väliajoin järjestettävien tilannepalaverien avulla. Kanbanin mukaan toimittaessa johdon täytyy pystyä luottamaan siihen, että työntekijät ottavat vastuuta tehtävistä myös ilman virallisia palavereita. Johdolle voi olla myös haastavaa luottaa sii-

hen, onko työntekijöillä riittävän hyvä kokonaiskäsitys tilanteesta ja pystyvätkö he priorisoimaan tarvittavia tehtäviä oikeilla perusteilla sen mukaan ovatko tehtävät kriittisiä vai eivät. (Seikola ym., 2011.)

Valittu Kanbanin implementointi voi nojata kasvokkain keskusteluihin, fyysisiin Kanban-tauluihin sekä seisaaltaan tapahtuviin pikapalavereihin. Nämä voivat muodostaa omat haasteensa, mikäli tiimit on maantieteellisesti hajautettu useampaan paikkaan. Hajautetussa organisaatiossa tulee kiinnittää erityistä huomiota siihen, että saatavilla oleva tieto ja käytössä olevat Kanban-taulut on synkronoitu sijainnista huolimatta. (Seikola ym., 2011.)

Hajautettujen tiimien kompetenssit voivat poiketa toimipisteiden välillä. Vaikka tiimit omaisivat riittävän osaamisen tehtäviinsä, on mahdollista, että niiden erityisosaamiset painottuvat epätasaisesti toimipisteiden välillä. Tämä aiheuttaa haasteita rästilistan tehtävien valikoinnissa. Eräs ratkaisu tähän on luoda vahtirooli (Component guardian) kompetenssien kehittämistä varten. Vahdin tehtävään valitaan kokenut kehittäjä, jolla on hyvää kokemusta vaaditulta tekniseltä osa-alueelta. Vahdille on allokoitu aikaa muiden kehittäjien tukemiseen ja opettamiseen tällä osa-alueella. Tämän kaltainen jatkuvan oppimisen malli voidaan kokea myös uuvuttavana ja turhauttavana, koska työntekijät eivät pääse koko ajan työskentelemään heille tutuimmalla alueella ja tuttujen tehtävien parissa. (Seikola ym., 2011.)

Mikäli organisaation kaikilla osastoilla ei ole käytössä ketteriä kehitysmenetelmiä, voi työn läpinäkyvyys kärsiä, eikä tehtäville töille voida välttämättä määrittellä WIP-rajoitteita. Näin ollen pullonkaulojen havaitseminen ei-ketterissä prosesseissa kärsii ja hankaloituu. (Seikola ym., 2011.)

3.4 Lean-transformaatio ohjelmistotuotannossa

Siirtyminen perinteisestä toiminnasta Lean-ajattelun mukaiseen toimintaan ei ole ongelmattonta myöskään ohjelmistotuotannossa. Kuuselan ja Koivuluoman (2011) mukaan Middleton ja Sutton (2005) pitävät ohjelmistotuotantoa yhtenä sopivimmista liiketoiminnan aloista, joka voidaan transformoida Lean-periaatteiden mukaiseksi. Kuuselan ja Koivuluoman (2011) mukaan tässä transformoinnissa voidaan käyttää Poppendieckin ja Poppendieckin (2003, 2006) Lean-periaatteita.

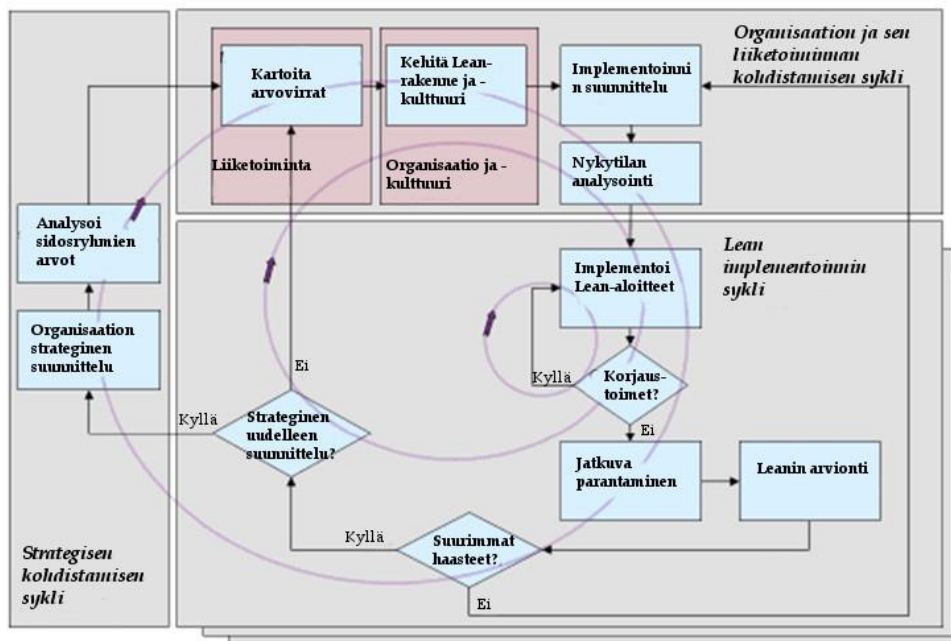
Mikäli yritys aikoo omaksua Lean-ajattelun, tarvitaan kokonaisvaltaista lähestymistä. Vain tämän seurauksena syntyy Lean-organisaatio (Kuusela & Koivuluoma, 2011). Kuuselan ym. (2011) mukaan Lean-transformaation toteuttaminen valmiiksi ja koko organisaation kattavaksi on yleensä vuosia kestävä prosessi. Tähän liittyen Hines ym. (2004) esittelee Womackin ja Jonesin (1996) kehittämän aikaikkunan globaalille Lean-organisaatiolle, jonka transformatio-prosessi koko organisaation kattavaksi kestää jopa viisi vuotta. (Kuusela ym., 2011.)

Lean-transformaatiota voidaan lähestyä kahdesta näkökulmasta. Transformaatioon on olemassa askel askeleelta eteneviä ohjeistuksia sekä enemmän

ajattelua painottavia ohjeistuksia. Bhasinin ja Burcherin (2006) mukaan Lean on enemmän filosofia kuin strategia ja tästä syystä Lean tulisi nähdä enemmän matkana ja ajattelutapana siitä, kuinka nähdä liiketoiminta ja sen prosessit. Vastakohtana tälle on se, että transformaatio toteutettaisiin tiukasti tiettyjä ennalta määrättyjä ohjeita käyttööön tuomalla. (Kuusela ym., 2001.) Tässä mielessä Bhasinin ja Burcherin (2006) näkemys Leanista on enemmän japanilainen ja filosofinen näkökulma, ja siinä on selviä viitteitä Likerin (2004), Ohnon (1988) ja Toyotan alkuperäiseen näkemykseen Leanista.

Kuuselan ym. (2011) mukaan aiemmin esiteltyä Bozdoganin ym. (2000) transformaatiomallia voidaan soveltaa hyvin myös ohjelmistotuotannossa, kunhan huomioidaan, että kyseinen malli on tehty organisaation näkökulmasta, eikä spesifisti vain ohjelmistoliiketoimintaa ajatellen. Tämä transformaatiomalli antaa Lean-transformaatiosta kiinnostuneille yrityksille hyvät vaihteittaiset ohjeet siitä, kuinka aloittaa, ylläpitää ja jatkuvasti kehittää toimintaansa Lean-periaatteiden ja -käytänteiden mukaiseksi (Kuusela & Koivuluoma, 2011).

Kuusela ja Koivuluoma (2011) ovat kehittäneet oman transformaatiomallinsa ohjelmistotuotantoa varten Bozdoganin ym. (2000) transformaatiomallin pohjalta. Malli koostuu kolmesta eri syklistä, joita ovat strategisen kohdistamisen sykli (strategic alignment cycle), organisaation ja liiketoiminnan kohdistamisen sykli (organisational & business alignment cycle) ja Lean-implemtoinnin sykli (Lean implementation cycle) (Kuvio 4).



KUVIO 4 Lean-transformaatiomalli ohjelmistotuotantoon (Kuusela ym., 2011, 380)

Eniten aikaa vievä sykli on *strategisen kohdistamisen sykli*, joka vastaa hyvin pitkälti Bozdoganin ym. (2000) transformaatiomallin sisäänkäynti- / paluusykliä. Sykli koskee organisaation korkeinta tasoa, jossa strateginen suunnittelu tapah-

tuu. Leanin koskiessa koko yritystä sen kaikilla tasoilla tulisi se huomioida jo suunnitteluvaiheessa, eikä vain liittää sitä erillisenä kokonaisuutena aiemman ja Leania huomioimattoman strategisen suunnitelman päälle. (Kuusela & Koivuluoma, 2011.) Kuusela ja Koivuluoma (2011) painottavat tässä kohti myös päämäärien asettamista kaikille sisäisille tai ulkoisille sidosryhmille.

Toinen sykli, eli *organisaation ja sen liiketoiminnan kohdistamisen sykli* toteutetaan, kun yrityksen strategiset päämäärät on asetettu ja hyväksytty ja syklillä on näin ollen sen tarvitsemat syötteet aloitusta varten. Verrattuna Bozdoganin ym. (2000) transformaatiomalliin tämä sykli vastaa sen pitkän aikajänteen sykliä. Tässä syklissä sekä liiketoiminnan että organisaation kehittämiseen liittyvät asiat suunnitellaan ja toteutetaan käytännössä. (Kuusela & Koivuluoma, 2011.) Bozdoganin ym. (2000) malli painottaa tässä syklin vaiheessa arvovirran tärkeyttä ja kehottaa toteuttamaan kartoituksen siitä, kuinka asiakkaat saavat arvoa yrityksen arvovirrasta. Kuusela ja Koivuluoma (2011) vievät tämän ajatuksen hieman pidemmälle ja ehdottavat, että nykypäivän jatkuvassa muutoksessa olevassa yritysmaailmassa tulisikin tutkia tarkasti eri asiakasarvoja, arvoverkostoja, liiketoiminta-analyyseja ja tuottomalleja. Kuuselan ja Koivuluoman (2011) mukaan arvovirtojen kartoituksen lisäksi tässä syklin toisessa vaiheessa tulisi suorittaa Leanin rakenteellinen suunnittelu organisaation näkökulmasta sekä Leanin huomiointi yrityskulttuurissa.

Erona Bozdoganin ym. (2000) malliin, jossa Leanin implementointisuunnitelman kehittäminen on sijoitettu kolmanteen, eli lyhyen aikajänteen sykliin, Kuusela ja Koivuluoma (2011) ovat sijoittaneet tämän, sekä nykyisten prosessien ja toimintojen analysoinnin, kokonaan oman mallinsa toiseen sykliin. Kuuselan ja Koivuluoman (2011) mukaan Bozdoganin ym. (2000) transformaatiomalli on vaihe vaiheelta etenevä malli, jonka tulisi edetä lineaarisesti, mutta he kuitenkin kertovat, että syklien välisissä siirtymissä on mahdollista toteuttaa joitain vaiheita, kuten Lean-implementointisuunnitelma, ikään kuin kahden eri syklin risteyksessä.

Kolmas sykli, eli *Lean-implementoinnin sykli* voidaan aloittaa, kun päästrategian, organisaation ja liiketoiminnan kohdistamista koskevat päätökset on tehty ja operaatioiden nykytila on määritelty. Tämä sykli on verrattavissa Bozdoganin ym. (2000) transformaatiomallin lyhyen aikajänteen sykliin. Lean-implementointi koostuu lyhyen tähtäimen toiminnoista ja keskittyy jatkuvaan toimintojen parantamiseen tiimin tai yksikön sisällä. Tässä kolmannessa syklissä tulisi myös huomioida, onko suurille tai pienille korjaaville toimenpiteille tarvetta. (Kuusela & Koivuluoma 2011.)

Vaikka strateginen suunnitelma on luotu nimenomaan pitkän tähtäimen muutoksia varten, tulisi transformaatiomallin kolmannessa syklissä ajoittain pohtia, onko jatkossa tarvetta päivittää strategista suunnitelmaa nykyhetken tietojä vastaavaksi. Useissa, etenkin suurissa yrityksissä eri toiminnoille on kokonaan omat yksikkönsä tai osastonsa. Mikäli yritys rakenne koostuu eri osastoista tai yksiköistä, kolmannen syklin toiminnot tulisi toteuttaa näissä yksiköissä yhtäaikaaisesti rinnakkain. (Kuusela & Koivuluoman, 2011.)

Kuuselan ja Koivuluoman (2011) transformaatiomallin kolmas sykli on hyvin pitkälti Bozdoganin ym. (2000) transformaatiomallia vastaava, pois luki-en Kuuselan ja Koivuluoman (2011) mallissa toiseen sykliin siirrettyjä implementointisuunnitelmaa ja nykytilan analysointia.

3.5 Leanin hyötyjä

Kuuselan ym. (2011) mukaan useat lähteet kuten Middleton (2001) ja Fujitsu (2010), ovat sitä mieltä, että oikein toteutetulla Leanilla on monia etuja verrattuna perinteisempiin lähestymistapoihin. Middletonin (2001) mukaan Lean-ajattelu liiketoiminnassa voi vähentää virheiden määrää jopa yhteen per miljoona tuotettua osaa kohti sekä vähintään kaksinkertaistaa tehokkuuden niin teollisessa tuotannossa kuin palvelutuotannossakin sekä vähentää tuotantoaika ja täten johtaa kustannussäästöihin. Yleisellä tasolla Lean johtaa selkeään ja kokonaisvaltaiseen toiminnan paranemiseen. Fujitsun (2010) raportoidut tulokset antavat samansuuntaisia tuloksia. Heilläkin Leanin käyttö on nostanut tuotavuutta 30 %. (Kuusela ym., 2011)

Middleton ym. (2005) tutkivat yritystä, joka käytti Leanin mukaisia käytänteitä päivittäisessä työssä kahden vuoden ajan. Tutkimuksessa havaittiin, että useat yrityksen prosessien vaiheet eivät tuottaneet arvoa lainkaan. Kysely henkilöstössä osoitti, että enemmistö kannatti Leanin mukaisia ideoita ja ajatteli, että ne olisi mahdollista sovittaa myös ohjelmistokehitykseen. Vain 10 % vastanneista ei ollut vakuuttunut Lean ohjelmistokehityksen tarjoamista hyödyistä. Yrityksestä kerätty tieto osoitti 25 % prosenttien kasvun tuottavuudessa sekä aikatauluista lipsuminen väheni aiemmasta kuukausien tai vuosien tasosta neljään viikkoon. Samalla virheiden korjaaminen väheni 65 % - 80 %. Loppuasiakaspalaute koskien Lean-kehittämisen mukaisia tuotejulkaisuja oli ylivoimaisen positiivista aiempaan verrattuna. (Middleton ym., 2005.)

Bozdoganin ym. (2000) mukaan ne yritykset, jotka ovat onnistuneet Lean-transformaatiossaan ovat niitä yrityksiä, joissa transformaatioon on otettu kokonaisvaltainen ote. Näissä transformaatio on nähty koko organisaation kattavana sekä toimintaa ja toimintoja perusteellisesti muuttavana toimenpiteenä, joka kattaa organisaatorakenteen, liiketoiminta- ja tietojärjestelmien, työvoimapolitiikan ja sidosryhmien välisten järjestelmien ja suhteiden muutoksen (Kuusela ym., 2011).

Kuuselan ym. (2011) mukaan Bozdogan ym. (2001) ovat sitä mieltä, että vaikka näistä yrityksistä ei ole tarkkoja mittaustuloksia, on niiden lopputuloksissa nähtävissä seuraavia selkeitä yhtäläisyyksiä, joita onnistuneelta transformaatiolta voidaan odottaa:

- Asiakkaan äänestä tulee yrityksen vetävä voima, joka johtaa selkeisiin parannuksiin tuotannon laadussa, organisaatorakenteessa, tuotantoprosesseissa, menettelytavoissa ja yleisessä käyttäytymisessä.
- Uusien tuotteiden kehitysaika pienenee merkittävästi.

- Suhteet toimittajiin muuttuvat täysin uudenaikaisiksi.
- Muuttuviin markkinaolosuhteisiin vastaaminen paranee.
- Organisaatorakenne muuttuu vertikaalisesta horisontaaliseksi, mikä johtaa arvoa lisääviin toimiin asiakkaan arvovirrassa.
- Työvoima on voimaantunutta.
- Parantuneet liikevoittomarginaalit ja kasvanut joustavuus tuo lisää uusia liiketoimintamahdollisuuksia nykyisillä ja uusilla markkinoilla.

Huomioitavaa on, että Middletonin (2005) tutkimusta lukuun ottamatta yllä olevat tulokset eivät ole erityisesti ohjelmistotuotantoalalta tutkittuja ja mitattuja, vaan yleisesti eri teollisuuden ja palvelutoiminta aloilta kerättyjä havaintoja. Tämän lisäksi Middleton ja Sutton (2005) ovat sitä mieltä, että ohjelmistotuotanto on yksi sopivimmista liiketoiminnan aloista, joka voidaan transformoida Lean-periaatteiden mukaiseksi. Näin ollen ei aiemmissa tutkimuksissa ole havaittavissa ainakaan yhtä selkeää syytä siitä, miksi edellä mainittuja tuloksia ei voitaisi odottaa myös ohjelmistotuotantoalan yrityksiltä. Tätä puoltaa erityisesti se, että Poppendieckin ja Poppendieckin (2003, 2006) mukaan perinteisissä ohjelmistotuotannon prosesseissa on havaittavissa selkeää hukkaa.

3.6 Leanin kustannuksia

Lean-implementointi ei yleensä vaadi juurikaan rahallisia lisäinvestointeja, vaan kustannukset syntyvät lähinnä investoinneista koulutukseen ja harjoitteluun. Joissain yrityksissä ja olosuhteissa joudutaan kuitenkin panostamaan myös tuotantovälineistöön, jotta saadaan käyttöön riittävän luotettava ja informaation virtausta yrityksen läpi edistävä välineistö. (Bozdogan ym., 2000.)

Cusumanon (1994) mukaan tietyissä olosuhteissa Lean-implementoinnilla voi olla myös negatiivisia vaikutuksia niin yrityksen sisällä kuin sen ulkopuolella. Esimerkeiksi tällaisista Cusumano (1994) nostaa juuri oikeaan aikaan varastoon tuotavat raaka-aineet, jotka saattavat johtaa kuljetusten kokonaisuuden kasvamiseen ja tätä kautta liikenteen ruuhkautumiseen ja suurempiin päästöihin. Samoin raaka-ainetoimitusten viivästyessä ja varaston ollessa minimissä voi tilanne johtaa siihen, että tuotantolaitoksella odotellaan toimitettomana kuljetuksen saapumista. Mikäli tätä ilmaantuu toistuvasti, voi lopputulos johtaa siihen, että tuotantoa joudutaan siirtämään ulkomaille asti, jotta löydetään riittävän luotettavat raaka-aineiden toimittajat. (Kuusela ym., 2011.)

Hines ym. (2004) toteavat Lean-kriittisyyttä olleen niin pitkään kuin Lean on ollut olemassa. Lean-ajattelu on kuitenkin kehittynyt aikojen saatossa ja siinä olleita puutteita on onnistuttu paikkaamaan. Usein kriittinen ajattelun pääkohdat ilmenevät puutteena ymmärtää epävarmuutta, ihmisenäkökulmaa, strategista näkökulmaa ja muutoksen kanssa toimeen tulemistä. (Kuusela ym., 2011.)

Leanin kustannuksiin liittyen tulee huomioda, että lähdemateriaaleista ei löytynyt erikseen ohjelmistokehityksen näkökulmasta mainittuja, teollisuuden

Lean-kustannuksista poikkeavia näkemyksiä. Lähdemateriaaleissa ei kuitenkaan mainittu yhtään selkeää syytä sille, miksi Lean-ohjelmistotuotannon kustannukset poikkeaisivat teollisuuden puolella tunnistetuista Leanin kustannuksista etenkin kun alasta riippumatta yleisimmiksi haasteiksi ja kustannuksiksi on tunnistettu koulutukseen ja harjoitteluun tarvittava aika sekä Lean-kriittisyys.

3.7 Yhteenveto

Tässä luvussa esitettiin, mitä Leanilla tarkoitetaan ohjelmistotuotannossa ja mistä se on saanut alkunsa. Luvussa kerrottiin myös, miten Lean ohjelmistotuotannossa eroaa teollisuuden yhteydessä käytetystä Leanista, kuinka ohjelmistotuotanto voitaisiin transformoida Lean-periaatteiden mukaiseksi ja mitkä ovat Lean-ohjelmistotuotannon hyödyt ja kustannukset. Luvussa esitettiin myös, mitä tarkoitetaan Kanbanilla Leanin mukaisessa ohjelmistokehityksessä.

Vaikka Leania on käytetty ohjelmistotuotannossa 2000-luvun alusta saakka, yksiselitteinen vastaus siihen, miten Lean-ohjelmistotuotanto määritellään, on vielä vailla tarkkaa vastausta (Jonsson ym., 2013). Poppendieckin ja Poppendieckin (2003, 2006) mukaan ohjelmistotuotannossa on seitsemän erityyppistä hukkaa. He ovat myös kehittäneet seitsemän peruseriaatetta Lean-ohjelmistokehitykselle. Poppendieckien tunnistamat hukat ja peruseriaatteet eivät eroa kuitenkaan teollisuuden puolen vastaavista kovinkaan merkittävästi, vaan ovat ikään kuin ohjelmistotuotannon näkökulmasta tehty tarkempi kuvaus Leanista. Poppendieckien näkemys Lean-ohjelmistotuotannosta ei kuitenkaan ole ainoa, vaan muun muassa Middleton ja Sutton (2005) lähestyvät Lean-ohjelmistotuotantoa suoraan Womackin ja Jonesin (2003) peruseriaatteiden kautta.

Kanban ohjelmistokehityksessä on ollut yksi viimeaikojen kuumimmista puheenaiheista, ja sen suosio on kasvanut viimeisten vuosien aikana. Tunnetuimmat teokset Kanbanin käytöstä ohjelmistokehityksessä ovat Anderson (2010) ja Kniberg ja Skarin (2010). Andersonin (2010) mukaan Kanban perustuu viiteen ydinominaisuuteen. Kniberg ja Skarin (2010) mukaan Kanban sisältää kolme perussääntöä. Andersonin (2010) ydinominaisuuksissa ja Knibergin ja Skarinin (2010) kolmessa säännössä yhtenevät kohdat ovat *työn virtauksen visualisointi, käynnissä olevan työn määrän rajoittaminen sekä mittaa ja johda työn virtausta / työn osan läpimenoajan mittaaminen*.

Kanbanin käyttöönnotolla on todettu olevan selkeitä hyötyjä ohjelmistokehityksessä, näistä keskeisimpinä suurempi saavutettu arvo suhteessa käytettyyn aikaan, tiimien nopeampi oppiminen jatkuvien prosessien parantamisessa, tiimityön parantuminen ja vastuunottamisen kasvaminen tiimin sisällä (Shalloway ym., 2009, s.100; Seikola ym., 2011).

Luvussa esiteltiin myös ohjelmistotuotannon transformointia Lean-periaatteiden mukaiseksi. Luvussa esitettiin myös Kuuselan ja Koivuluoman

transformaatiomallia, joka perustuu Bozdoganin ym. (2000) vastaavaan esitykseen.

Oikein toteutetulla Lean-lähestymistavalla on havaittu olevan monia hyötyjä. Kuusela ym. (2011) mainitsevat hyödyiksi muun muassa tehokkuuden ja tuottavuuden merkittävän kasvun sekä Leanin johtamisen selkeään ja kokonaisvaltaiseen paranemiseen. Hyötyjen lisäksi Leanilla on havaittu olevan kustannuksia. Kustannuksista Kuusela ja Koivuluoma (2011) sekä Bozdogan ym. (2000) nostavat esille muun muassa Lean-implementoinnin aiheuttamat investoinnit koulutukseen ja harjoitteluun.

4 LEANIN MUKAISEN TOIMINNAN ARVIOINTI

Tässä luvussa kerrotaan, mitä tarkoitetaan Leanin mukaisen toiminnan arvioinnilla ja mitkä ovat sen keskeisimmät pääkohdat. Luvussa esitellään myös tunnetuimpia arviointimenetelmiä ja -malleja.

4.1 Tausta

Karvosen, Rodriguezin, Kuvajan, Mikkosen ja Oivon (2012) mukaan Lean-periaatteet ovat kiehtoneet alasta riippumatta yrityksiä ja tutkijoita ympäri maailman siitä saakka, kun ne esiteltiin ensimmäisiä kertoja TPS:n (Toyota Production System) yhteydessä.

Karvonen ym. (2012) määrittelevät organisatorisen muutoksen Jonesin (2004) mukaan prosessiksi, jossa organisaatiot siirtyvät nykyisestä tilastaan haluttuun tulevaisuudentilaan kasvattaakseen tehokkuuttaan. Muutokset organisaatioissa voivat tapahtua useilla eri tasoilla, kuten manuaalisen työn automatisoinnilla tai uudelleen miettimällä koko organisaation luonne sekä määrittelemällä sille uudet liiketoimintamallit. Mitä laajemmalle alueelle transformaatio ulottuu, sitä suuremmaksi organisaation muutokseen liittyvät riskit kasvavat. (Karvonen ym., 2012.) Karvonen ym. (2012) määrittelevät Womackin ym. (1990) esitykseen perustuen *Lean-organisaation* organisaatioksi, joka ei käytä Leania vain joillain toiminnan aloillaan, vaan kaikessa mitä organisaatio tekee.

Arvioinnin on todettu olevan olennaista siirtyessä kohti Leanin mukaista toimintaa. Pintapuolinen, lopputulokseen keskittyvä arviointi (evaluation) organisaation nykytilan suorituskyvystä ja sen toiminnallisesta tehokkuudesta ei ole riittävän laaja ohjatakseen toimintaa tarpeeksi kohti Lean-transformaatiota. Syvällisempi kokonaisvaltainen arviointi (assessment) lähestyy nykytilan arviointia huomattavasti laajemmin peilaten organisaation sisäisiä mahdollisuuksia ja tuottaa samalla hyödyllistä informaatiota organisaation sisäisten prosessien parantamisen tueksi. (Karvonen ym., 2012.)

Arvioinnit ovat hyödyllinen tapa saada kuvaa siitä, kuinka Lean-transformaatio edistyy. Aika-ajoin on tärkeää tarkastaa, tarvitaanko organisaatiossa suuria tai pieniä korjaavia toimenpiteitä. Vaikka strateginen suunnittelu tähtää pitkän aikavälin suunnitelmiin, välillä olisi hyödyllistä pohtia, onko organisaation strategisia suunnitelmia tarvetta katsoa tarkemmin uudelleen tai päivittää niitä vastaamaan ennalta näkemättömiä muutoksia. (Kuusela & Koivuluoma, 2011.)

4.2 Lean-organisaation arviointia koskevia tutkimuksia

Lean-organisaation mukaiseen toimintaan siirtymisen ja arvioinnin avuksi on tehty tutkimusta sekä työkaluja. Tutkimukset ja työkalut on kohdistettu lähinnä yritysjohtoon käyttöön. Muun muassa Doolen ja Hacker (2005) tutkivat ja vertailivat muutamia tunnetuimpia Lean-organisaation arviointiin tehtyjä arviointikyselyitä ja työkaluja.

Taulukossa 1 on esitetty Doolenin ja Hackerin (2005) mukaisesti viisi tunnettua Lean-kyselytutkimusta. Taulukon perustella voidaan todeta useiden kyselytutkimusten keskittyvän arvioimaan Leania lähinnä Lean-käytänteiden näkökulmasta korkeamman tason käsitteiden kuten JIT:n (Just-in-time) tai TQM:n (Total Quality Management) perusteella.

Aiempaa tutkimusta ja materiaalia Leanin arvioinnista on myös erilaisten arviointityökalujen muodossa. Doolen ja Hacker (2005) kävivät tutkimuksessaan läpi myös seitsemän tunnettua, pääasiassa teollisuustuotannon käyttöön suunnattua Lean-arviointityökalua. Arviointityökalut sisältävät useita Lean-käytänteitä, mutta ne olivat epäjohtomukaisia arvioitavissa Lean-käytänteissä sekä Lean-implemmentaation tason arvioinnin tunnistamisessa (Doolen & Hacker, 2005). Taulukossa 2 on esitetty yhteenveto näistä työkaluja koskevista tutkimuksista.

Tutkimuksessa mainittujen kyselytutkimusten ja arviointityökalujen sisältö paljastaa, että ne painottavat arvioinnissa eri näkökulmia. Osassa painopiste pidetään selkeämmin Lean-työkalujen ja -käytänteiden piirissä "lattiatasossa". Vastaavasti osa korostaa kokonaisvaltaisempaa ja organisaation kaikkia tasoja huomioivaa suuntausta, ei siis pelkästään työkalujen tai käytänteiden vaan myös johtamisen ja koko organisaatiokulttuurin näkökulmista. (Doolen & Hacker, 2005.)

Seuraavaksi esitetään taulukoiden muodossa yhteenveto Doolenin ja Hackerin (2005) tutkimasta viidestä Lean-kyselytutkimuksesta (taulukko 1) ja seitsemästä teollisuuden käyttöön suunnitellusta Lean-arviointityökaluja koskevasta tutkimuksesta (taulukko 2).

TAULUKKO 1 Yhteenveto viidestä tutkitusta Lean-kyselytutkimuksesta

Viitteet	Kuvaus ja käsitellyt Lean-näkökulmat
Fullerton, McWatters & Fawson (2003)	Tämä tutkimus perustui kyselyyn, joka mittaa JIT (Just-in-time) implementoinnin tasoa organisaation sisällä. Kymmenen JIT-elementtiä määriteltiin tutkimusta varten ja yksitoista vastaavaa kyselyosiota kehitettiin arvioimaan JIT-implementoinnin tasoa.
Nightingale & Mize (2002)	Tämä tutkimus kuvailee Lean Aerospace Initiativen kehittämän arviointityökalun rakennetta. Työkalu käsittää kolme osiota: Lean-transformaation johtaminen, elinkaarisykli-prosessi ja infrastruktuurin mahdollistaminen. Työkalu sisältää yhteensä 54 Lean-käytännettä. Nämä käytännöt on valittu koska ne edustavat käytänneindikaattoreita, jotka Lean-organisaatioiden tulisi omaksua toimintaansa.
Perez & Sanchez (2000)	Tämä tutkimus perustuu kenttätutkimukseen autoteollisuuden myynnin parissa. Kerätty tieto sisälsi organisatorisen demografian (työntekijöiden määrä, organisaation ikä, myynti sekä omistajuus), teknologiainnovaatioiden lähteen, joustavien tuotantoteknologioiden käytön (JIT), työvoiman sekä työpaikan joustavuuden mittareita (tiimit, työnkierto, harjoittelu).
Panizzolo (1998)	Tämä tutkimus perustui kenttätutkimukseen useiden italialaisten valmistajien keskuudessa eri teollisuuden aloilta. Kyselyosiot kehitettiin ottamaan selvää kuuden eri osa-alueen (prosessit ja laitteet, tuotantosuunnittelu ja -ohjaus, työvoimaresurssit, tuotesuunnittelu, toimittajasuhteet ja asiakassuhteet) välisten Lean-käytänteiden implementoinnista. Esimerkkejä Lean-käytänteistä: asennusten redusointi, JIT, toimittajan kokonaiskustannusten arviointi, ajallaan toimitukset sekä asiakkaiden osallistuminen tuotekehitykseen.
Shah & Ward (2003)	Tämä tutkimus perustui vuosittaiseen kyselyyn tuotantojohtajille vuonna 1999 IndustryWeek-lehden toimesta. Kysely sisälsi kysymyksiä liittyen 22 eri Lean-käytännön implementoinnin tasoihin. Kyselyyn sisällytetyt käytännöt liittyivät sellaisiin käytänteisiin kuten JIT, TPM (Total Productive Maintenance), TQM (Total Quality management) sekä työvoimaresurssien johtaminen.

TAULUKKO 2 Yhteenvedo seitsemästä tutkitusta teollisuuden arviointityökalusta

Viitteet ja työkalut	Kuvaus ja käsitellyt Lean-näkökulmat
Jordan & Michel (2001), Survey of Perceptions of a Company's Leanness	Tämä työkalu sisältää 36 kysymystä Leanin mukaisen toiminnan mittaamisesta. Kyselystä on eri versioita eri sidosryhmille (johto, työntekijät, sijoittajat, toimittajat ja asiakkaat). Kyselyiden yhdistetty sarja osoittaa huomioitavia aiheita kuten jätteen tunnistaminen ja vähentäminen, jatkuva parantaminen, arvovirran johtaminen, virtaus sekä ihmisresursseihin liittyviä aiheita kuten työntekijöiden kehittäminen ja johtajuus.
Lean Enterprise Implementation Group (1999), The 360° Lean Audit	Tätä arviointityökalua käytetään käytänteiden, prosessien johtamisen, Lean-työkalujen ja -tekniikoiden sekä toimitusketjun integroinnin tason arviointiin. Lean-työkalut ja -tekniikat ovat organisaation järjestys, jäte, virtaus, imu, laatu, standardit, PDCA (plan-do-check-act-sykli), välineiden tehokkuus ja niiden luotettavuus sekä linjattu tuotanto.
Lean Learning Center (2003), The Lean Company Survey	Tämä suorituskykyarviointi etsii tietoa tehokkuuden muutoksista, jotka liittyvät Leanin tai jatkuvan parantamisen toimiiin, infrastruktuurin yksityiskohtiin (kuka on vastuussa Lean toiminnoista) sekä implementoinnin yksityiskohtiin (minkä tyyppisiä Lean-työkaluja implementoidaan).
Northwest High Performance Enterprise Consortium (2002), HPEC Assessment	Tämä työkalu on rakennettu tunnistamaan laaja-alaisesti useita mitattavia Lean-implementoinnin tuloksia sisältäen muutokset johtamiskäytänteissä, laatusaavutukset, työntekijöiden osallistumisen, joustavat valmistuskäytänteet, ylläpitokäytänteet, varaston johtamisprosessit sekä uusien tuotteiden kehitysprosessit.
Robert Abair Associates, Inc. (2002), Lean Checklist Self-Assessment	Tämä itsearviointityökalu antaa organisaatioille vertailukohtamittauksen Leanin mukaisen toiminnan edistymisestä. Työkalu sisältää laajan valikoiman johtamis- ja Lean-käytänteitä kuten Lean-koulutuksen, harjoittelun, tilastollisen prosessien hallinnan, JIT (Just-In-Time), kaizen, heijunka, 5S, SMED, poka-yoke, jäte, työvoiman joustavuuden, tehokkuusmittaukset sekä QFD.
Wisconsin Manufacturing Extension Partnership (2002), Lean Business Assessment	Tämä itsearviointityökalu auttaa organisaatioita tunnistamaan ja priorisoimaan parannustoimia. Työkalu käyttää kymmentä Lean-periaatetta ja laajaa kokoelmaa Lean-käytänteitä kuten virtaustuotantoa, nopeita tuotantoprosessien välisiä vaihtoja, ihmisiä huomioivaa automaatiota, imujärjestelmiä, itsenäistä kunnossapitoa sekä kaizenraportointia.
Wisconsin Manufacturing Extension Partnership (2001), How Lean is Your Culture?	Tämä suppea itsearviointityökalu on suunniteltu auttamaan johtajia tunnistamaan kulttuurisia tekijöitä, jotka voivat tukea tai estää Lean-tuotannon aloitteiden pitkäjänteisyyttä.

4.3 LESAT-työkalu

Eräs tunnetuimpia työkaluja Leanin mukaisen toiminnan arviointiin on MIT:n kehittämä Lean Enterprise Self-Assesment Tool (LESAT) (MIT, 2012a). LESAT on alun perin kehitetty yhteistyössä ilmailuteollisuuden kanssa. LESAT:in tarkoituksena on ohjeistaa yrityksiä implementoimaan Leanin mukaista toimintaa arvioimalla nykytilaansa Lean-kyvykkyyteen perustuen sekä asettamaan tavoitteita tuleville askelille. Ilmailualan lisäksi LESAT on sittemmin otettu käyttöön onnistuneesti useilla eri aloilla kuten terveydenhuollossa sekä erilaisissa valtion organisaatioissa. (Karvonen ym., 2012.) Seuraavassa kerrotaan ensin LESAT:in rakenteesta ja sen jälkeen LESAT-arviointiprosessin rakenteesta ja yleisimmistä virheistä, jotka liittyvät LESAT-arvioinnin käyttöön.

4.3.1 Rakenne

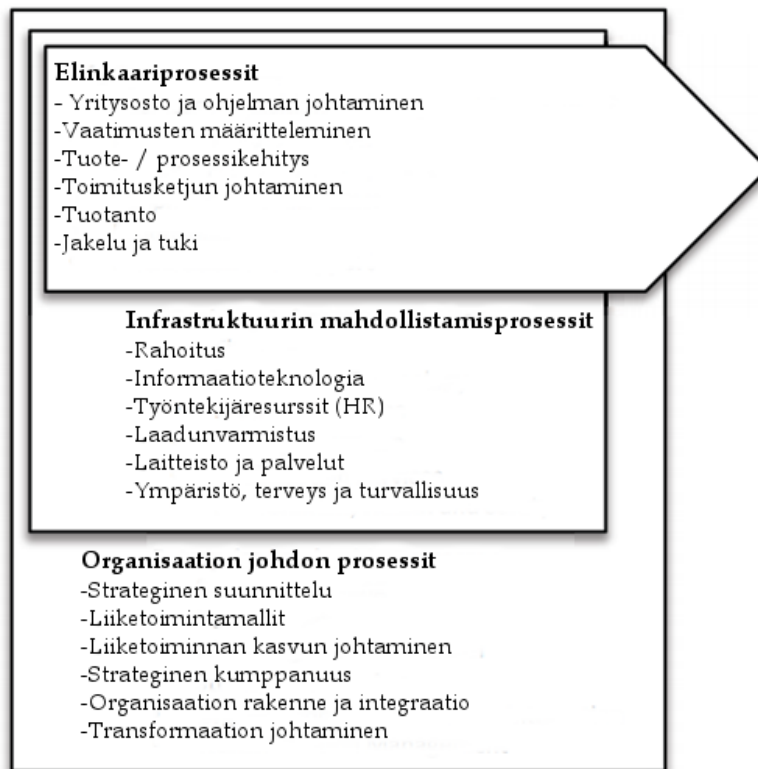
LESAT:in tarkoituksena on arvioida kokonaisvaltaisesti organisaation nykytilaa sen Lean-kyvykkyyden suhteen sekä asettaa selkeät tavoitteet halutulle tulevaisuuden tilalle. LESAT ei keskity ainoastaan Leanin ominaispiirteisiin vaan kiinnittää huomiota myös Lean-käyttäytymiseen sekä erityisesti johtamisen aloitteellisuuteen ja sen vaalimiseen transformaatioissa kohti Lean-organisaatiota. LESAT koostuu arviointimatriiseista, jotka sisältävät Lean-transformaatiota ajatellen tärkeimpiä näkökulmia. (MIT, 2012a.)

Karvosen ym. (2012) mukaan LESAT:in pohjana olevat oletukset perustuvat MIT:n laajaan tietämykseen ja kokemukseen Leanista ja ne voisivat olla soveliaita muokattaviksi myös ohjelmistoyritysten ohjaamiseksi kohti Leanin mukaista toimintaa.

Karvosen ym. (2012) mukaan LESAT-arviointi jakautuu kolmeen eri osioon, joita ovat:

1. *Lean-transformaatio-/johtajuusosio*, joka sisältää prosessit ja johtamisperusteet, jotka ohjaavat transformaatiota kohti Lean-periaatteita ja käytänteitä.
2. *Elinkaariprosessiosio*, joka keskittyy tuotekehitysprosesseihin konseptitason tuotekehityksestä aina toimituksen jälkeiseen ylläpitoon saakka.
3. *Infrastruktuurin mahdollistamisiosio*, joka arvioi niitä prosesseja, jotka hoitavat ja ohjaavat organisaation toimintojen mahdollistavia resursseja.

LESAT-arvioinnin osiot rakentuvat toistensa päälle niin, että organisaation johtamisprosessit luovat pohjan infrastruktuurin mahdollistamisprosesseille, joiden pohjalta lopulta elinkaariprosessit rakennetaan (Karvonen ym. 2012). Tarvemmin Karvonen ym. (2012) kuvaavat LESAT-prosessien rakentumisen ja rakenteen seuraavan (kuvio 5) mukaisesti.



KUVIO 5 Lean-organisaation prosessiarkkitehtuuri (Karvonen ym. 2012, 268)

MIT:n (2012a) mukaan LESAT-arviointi jakautuu myös kolmeen osioon, mutta osioiden tarkempi sisältö poikkeaa jonkin verran Karvosen ym. (2012) esittämästä. MIT:n (2012a) mukaan organisaation *transformaatio-* ja *johtajuusosio* jakautuu seuraavaan kahdeksaan kohtaan:

- *Määritä korkeimman prioriteetin strategiset tavoitteet* (Determine strategic imperative).
- *Sitouta organisaation johto transformatioon* (Engage enterprise leadership in transformation).
- *Ymmärrä organisaation nykytila* (Understand current enterprise state).
- *Visioi ja suunnittele tulevaisuuden organisaatio* (Envision and design future enterprise).
- *Kehitä organisaatorakenne ja käyttäytyminen* (Develop enterprise structure and behavior).
- *Luo transformatiosuunnitelma* (Create transformation plan).
- *Implementoi ja sovita transformatiosuunnitelma* (Implement and coordinate transformation plan).
- *Vaali transformatiota ja sulauta organisaatioajattelu* (Nurture transformation and embed enterprise thinking).

MIT:n (2012a) mukaan organisaation *elinkaariprosessiosion* sisältö jakautuu seuraaviin viiteen osaan:

- *Hanki, kehitä ja käytä vipuvoimana organisaation kyvykkyyttä* (Acquire, develop and leverage enterprise capabilities).
- *Optimoi koko verkoston laajuinen tehokkuus* (Optimize network-wide performance).
- *Sisällytä asiakasarvo organisaation arvoketjuun* (Incorporate downstream customer value into enterprise value chain).
- *Sitouta aktiivisesti sidosryhmiä maksimoimaan arvon luominen* (Actively engage upstream stakeholders to maximize value creation).
- *Hanki kyky tarkkailla ja hallita riskejä ja suorituskykyä* (Provide capability to monitor and manage risk and performance).

MIT:n (2012a) mukaan organisaation *infrastruktuurin mahdollistamisosio* jakautuu seuraaviin kahteen osaan:

- *Organisationaaliset mahdollistajat* (Organizational enablers)
- *Prosessien mahdollistajat* (Process enablers)

4.3.2 LESAT-arviointiprosessi fasilitoijan näkökulmasta

LESAT-arviointiprosessin toteutusta helpottamaan on luotu myös ohjeistusta *fasilitoijalle* eli henkilölle, joka johtaa LESAT-arviointia organisaation sisällä. Seuraavassa kuvataan LESAT-arviointiprosessia tarkemmin fasilitoijan näkökulmasta perustuen MIT:n (2012b) ohjeistukseen.

LESAT-arviointiprosessi sisältää viisi päävaihetta, joista jokainen on yhtä tärkeä. Vaiheet tulee suorittaa jaksoittain, sillä jokaisen vaiheen tulos toimii syötteenä seuraavalle vaiheelle. LESAT-arviointiprosessi on iteratiivinen, eli se toteutetaan pienemmissä osissa ja prosessia toistetaan. Kun arviointi on suoritettu ja tulokset analysoitu, prosessiin osallistujat arvioivat arviointiprosessin tehokkuutta ja saavutettuja hyötyjä sekä tunnistavat tarvittavat parannukset. Uusi arviointisykli alkaa arviolla siitä, kuinka nämä aiemmin tunnistetut parannukset on implementoitu arviointiprosessiin. (MIT, 2012b, s. 29.)

LESAT-arviointiprosessin ensimmäisen vaiheen, eli *arvioinnin esitietojen*, tarkoituksena on luoda organisaatioon ympäristö, joka varmistaa, että organisaatio saa mahdollisimman suuren hyödyn irti arvioinnin tuomista eduista. Tällaisessa ympäristössä organisaatio lähestyy arviointia sen tarjoamien hyötyjen motivoimana. Arviointi on pitkän aikavälin toimenpide, joka suoritetaan sykleissä ja se vaatii jatkuvaa sitoutumista. Ensimmäisen vaiheen tavoitteena on kasvattaa sitoutumista koko organisaation johtoportaan läpi perustuen ymmärrykseen arviointitavoitteista ja arvioinnin tarjoamista hyödyistä. Ensimmäisen vaiheen sidosryhmien roolit painottuvat yksinomaan organisaation johdon vastuulle, mutta toimintasuunnitelman implementoinnin katsauksen suorittavat yhdessä organisaation johto, arvioinnin tekijät ja LESAT-fasilitoija. (MIT, 2012b, s.30.)

MIT (2012b, s.31) määrittelee arviointiprosessin ensimmäisen vaiheen tehäväksi seuraavat kohdat:

- Sitouta organisaatio
- Määrittele organisaatio ja sen rajat
- Määrittele aikaikkuna tulevaisuuden organisaation tilalle
- Määrittele arvioinnin ajoitus
- Määrittele osallistujien roolit ja vastualueet
- Allokoi resurssit
- Tee tilannekatsaus toimintasuunnitelmien implementoinnin edistymisestä

LESAT-arviointiprosessin toisen vaiheen, *eli arvioinnin suunnittelun*, tarkoituksena on valmistautua arviointia varten ja varmistaa, että prosessi on käytettävissä ja tehokas. Prosessin tehokkuutta voidaan mitata esimerkiksi arviointitulosten käyttökelpoisuudella tai muiden tarkkojen ja pätevien tulosten perusteella, jotka johtavat priorisoitujen parannussuunnitelmien tunnistamiseen. Tästä syystä tulee kiinnittää huomiota vastaajien valintaan ja harjoitella kattavasti arviointityökalun käyttöä sekä organisaation periaatteita. Aikataulun laatiminen ja siihen sitoutuminen jo tässä vaiheessa parantaa arviointiprosessin tehokkuutta. Toisen vaiheen sidosryhmien roolit jakautuvat pääosin organisaation johdolle ja fasilitoijalle, jotka yhdessä valitsevat vastaajat sekä varmistavat, että vastaajat ovat käytettävissä ja valmiina osallistumaan arviointiin. Vastaajien vastuulla on ymmärtää aikavaatimukset ja sitoutua aikatauluihin ennen arvioinnin alkua. Toisen vaiheen lopputuloksena syntyy arviointisuunnitelma. (MIT, 2012b, s.35.)

MIT (2012b, s. 35-37) määrittelee arviointiprosessin toisen vaiheen tehtäviksi seuraavat kohdat:

- Tunnista osallistujat
- Kiinnitä arvioinnin aikataulu
- Tutustuta osallistujat työkaluun, arviointiprosessiin ja tulosten aiottu hyödyntämiseen
- Varmista ymmärrys LESAT-käytänteistä ja arviointijärjestelmästä

LESAT-arviointiprosessin kolmannen vaiheen, *eli arvioinnin suorittamisen*, tavoitteena on saada arviointitulokset jokaiselta vastaajalta, sisältäen nykytilan ja tulevaisuuden tilan pisteytykset sekä kommentit, todisteet ja mahdollisuudet jokaiselle käytänteelle. Kolmas vaihe on omistettu arvioinnille ja tulosten alustavalle analysoinnille. Vaiheen tuloksina syntyvät pätevät ja tarkat yksittäis- ja ryhmätulokset seuraavan vaiheen syötteenä. Tulosten perusteella tunnistetut ja priorisoidut parannukset tarvitaan, jotta organisaatio voi jatkaa omaa Leantransformaatioprosessiaan eteenpäin. Kolmannen vaiheen sidosryhmien roolit painottuvat arviointiin vastaajille, jotka vastaavat arviointiin fasilitoijan avulla joka tarvittaessa avustaa vastaajia. Vastaajat palauttavat vastauksensa fasilitoijalle annettuun aikarajaan mennessä. Fasilitoija kerää vastaukset ja suorittaa niiden perusteella alustavan analyysin. Analyysi toimii pohjana vastaajien kesken käytävälle keskustelulle, jota fasilitoija vetää rohkaisemalla vastaajia kes-

kustelemaan esille nousseista aiheista. Kolmannen arviointivaiheen lopputuloksena syntyvät analysoidut tulokset vastauksista. (MIT, 2012b, s. 38-47.)

MIT (2012b, s. 38-47) määrittelee arviointiprosessin kolmannen vaiheen tehtäviksi seuraavat kohdat:

- Toimeenpane arvioinnit, joko yksittäin tai ryhmämuotoisena
- Kerää ja prosessoi tulokset
- Keskustele ja analysoi tulokset

LESAT-arviointiprosessin neljännen vaiheen, *eli arviointitulosten ja prosessin arvioinnin*, tavoitteena tunnistaa LESAT-käytänteet, jotka vaativat parannustoimenpiteitä. Tätä neljättä vaihetta pidetään yhtenä merkityksellisimmistä vaiheista arviointiprosessissa, koska se tarjoaa kontekstuaalisen näkemyksen kvantitatiivisista tuloksista. Tämän vaiheen aikana arviointiin osallistuneet tunnistavat parannettavat alueet, jotka tulee nostaa esille transformaatioprosessin jatkoa ajatellen. Osallistujat arvioivat myös itse arviointiprosessia varmistaakseen, että se vastaa organisaation tavoitteita. Neljännen vaiheen sidosryhmien roolit painottuvat vastaajiin, jotka arvioivat arviointituloksia fasilitoijan avustuksella. Arviointiprosessin arviointiin ja sen kehittämiseen osallistuvat johto, vastaajat sekä fasilitoija. Neljännen vaiheen lopputuloksena syntyvät parannusehdotukset ja -toimet. Huomioitavaa tässä vaiheessa on tulosten perusteella tehtävät valinnat siitä, siirrytäänkö seuraavaksi vaiheeseen 5, vai palataanko takaisin vaiheeseen 2 suorittamaan arvioinnin suunnittelua. (MIT, 2012b, s. 48-52.)

MIT (2012b, s. 48-52) määrittelee arviointiprosessin neljännen vaiheen tehtäviksi seuraavat kohdat:

- Arvioi saadut tulokset havaitaksesi alueet, joita on varaa parantaa
- Arvioi arviointiprosessia

LESAT-arviointiprosessin viidennen vaiheen, *eli toimintasuunnitelman tekemisen ja resurssien priorisoinnin*, tavoitteena on toimeenpanna arvioinneista saadut tulokset toimintasuunnitelman muodossa sekä tukea sitä tarvittavilla resursseilla. Viidennen vaiheen sidosryhmien roolit painottuvat arvioinnin tekijöihin, jotka luovat fasilitoijan avulla toimintasuunnitelman. Viidennen vaiheen lopputuloksena implementoidaan toimintasuunnitelma käyttöön ja palataan takaisin arviointiprosessin alkuun vaiheeseen 1. (MIT, 2012b, s. 53.)

MIT (2012b, s. 53) määrittelee arviointiprosessin viidennen vaiheen tehtäviksi seuraavat kohdat:

- Priorisoi parannusta vaativat kohdat
- Tunnista tarvittavat tehtävät ja resurssit
- Priorisoi tarvittavat tehtävät ja resurssit

4.3.3 Yleisimmät virheet LESAT:in käytössä

LESAT:in käytön ei ole todettu olevan täysin ongelmaton organisaatioissa. LESAT-implementointia seurattaessa onkin havaittu muutamia yleisesti toistuvia, käyttöön liittyviä virheitä, jotka johtavat heikkoihin lopputuloksiin. (MIT, 2012b, s. 54.) Seuraavaksi kerrotaan lähemmin näistä virheistä.

Yksi suurimpia LESAT:in käyttöön liittyviä virheitä on suorittaa arviointi, mutta jättää sen tulokset tulkittamatta. Toimimalla tällä tavoin saadaan aikaan arviointi, jonka tuloksilla on vähän jos lainkaan vaikutusta transformaatio-suunnitelmaan. Tämä tilanne voidaan nähdä parina skenaariona. Ensimmäisessä skenaariossa tuloksia tarkastellaan pinnallisesti, yleensä katsomalla vain keskiarvoja. Toisessa skenaariossa tulokset analysoidaan, mutta niitä ei toimita oikeille tahoille tai niihin ei reagoida. Organisaatio voi saavuttaa näin toimimalla pieniä parannuksia organisaatietietämyksessä ja -sanastossa, mutta varsinaiset kasvunpaikat jäävät vähäisiksi, sillä tuloksia ei ole tulkittu oikein, eivätkä ne näin ollen ohjaa meneillään olevaa transformaatiota tai jatkuvaa parannusta. (MIT, 2012b, s.54.)

Joissakin organisaatioissa LESAT:in rinnalla käytetään myös muita toiminnanparannussuunnitelmia. Nämä organisaatiot saavuttavat hyötyjä LESAT:in tulkinnasta, mutta samanaikaisesti niiden tulisi kiinnittää erityistä huomiota vastakkaisten käytänteiden koordinointiin sekä rajallisten resurssien allokointiin eri toiminnanparannussuunnitelmien välillä. (MIT, 2012b, s.54.)

Toinen yleisesti havaittu virhe LESAT:in käytössä liittyy tulosten liialliseen yksinkertaistamiseen. Se tunnetaan laadun menettämisenä (quality loss). LESAT edustaa moniulotteista arviointia organisaation laaja-alaisesta toiminnasta, jonka jokainen käytänte on itsenäinen. Vaikka käytänteet voidaan järjestää osioihin ja alaosioiden, on väärin tutkia LESAT-tuloksia vain osioiden keskiarvojen perusteella. Samoin kaikkien käytänteiden keskiarvojen ynnääminen yhdelle yksittäiselle tuloskortille heikentää LESAT-tulosten arvoa merkittävästi. Yksittäinen tuloskortti voi tuntua houkuttelevalta johdon näkökulmasta, koska se antaa selkeän merkin parannustarpeesta, mutta samalla se heikentää tulosten yksityiskohtaisuutta ja täsmällisyyttä, jotka normaalisti saavutetaan monenlaisia LESAT-käytänteitä hyödyntämällä. Esimerkkinä tällaisesta tilanteesta voidaan pitää tilannetta, jossa pelkkiä keskiarvoja tarkkailemalla toisen joukon parannukset peittyvät toisen joukon heikkoihin tuloksiin. Ilman tämän tasoista ymmärrystä organisaatio ei voi muokata transformaatio-suunnitelmaansa, esitellä korjaavia toimia tai nähdä omaa menestystasoaan. (MIT, 2012b, s.54.)

Tulosten suorassa vertailussa ja keskiarvojen tarkkailussa on myös perustavanlaatuisia rajoituksia. LESAT-arvioinnin viisi tasoa on suunniteltu johdonmukaisiksi kaikille käytänteille, mutta tuloksia arvioitaessa tulee huomioida, ettei tilanne aina ole kuitenkaan tämän mukainen. Arviointi on taipuvainen epäjohdonmukaisuuksien skaalautumiseen, jotka voivat olla liioiteltuja, kun LESAT:ia katsotaan vastaajan tai organisaation näkökulmasta. Tämä johtuu siitä, että tulkinta skaalasta voi olla erilainen arviointiin vastanneen ja tuloksia analysoivan välillä. Voitaisiin sanoa, että tulosten suora vertailu ja keskiarvo-

hin keskittyminen on kuin eri mittayksiköiden vertailua. Näiden skaala voi vaihdella eri käytänteiden välillä. Tulosten keskiarvoilla on olemassa kuitenkin myös omat hyötynsä, koska niiden avulla voidaan valottaa osioiden välisiä yhteyksiä. (MIT, 2012b, s.54.)

Kolmas yleisesti havaittu virhe LESAT:in käytössä on ulkoisen ympäristön huomiotta jättäminen. Syvälinen ymmärrys organisationaalisesta kontekstista ja ympäristötekijöistä, joilla voi olla painoarvoa LESAT-tuloksiin, on äärimmäisen tärkeää. Lukuisat ympäristötekijät voivat vaikuttaa arviointiprosessin tuloksiin ja näiden huomioiminen on tärkeää tulosten ymmärtämiseksi. Esimerkkejä tällaisista ympäristötekijöistä ovat muun muassa vaihtuva johto, viimeaikaiset epäonnistumiset tai suuret onnistumiset, kasvava organisaatietietämys (ja tästä johtuva kysymysten tulkinnan vaihtuvuus) ja ennakoasenteet arvioinnin fasilitointia hoitavaa henkilöä kohtaan. On olemassa suuri joukko erilaisia ympäristötekijöitä, joilla on vaikutusta arvioinnin lopputuloksiin. LESAT-tulosten arvioinnin yhteydessä näiden ympäristötekijöiden huomiotta jättäminen johtaa arviointitulosten hyötyjen ja niistä saatavan arvon heikentymiseen. (MIT, 2012b, s.55.)

4.4 Yhteenveto

Tässä luvussa kerrottiin mitä tarkoitetaan Leanin mukaisen toiminnan arvioinnilla, ja esiteltiin aiempia Leanin mukaisen toiminnan arviointia käsitteleviä tutkimuksia sekä Leanin mukaisen toiminnan keskeisimpiä arviointitapoja ja työkaluja. Tarkempaan analyysiin valittiin yksi tunnetuimmista Lean-arviointityökaluista, MIT:n kehittämä Lean Self-Assessment Tool (LESAT). Luvussa kuvattiin LESAT:in rakenne, päävaiheet ja yleisimmät sen käytössä ilmevät virheet sekä arviointiprosessi fasilitoijan näkökulmasta.

Luvussa esitetyissä aiemmissä Leanin mukaisen toiminnan arviointia koskevat kyselytutkimukset ja välineet ovat sisällöltään vaihtelevia. Osassa painopiste on selkeämmin Lean-työkalujen ja -käytänteiden piirissä lattiatasossa. Vastaavasti osa korostaa kokonaisvaltaisempaa ja organisaation kaikkia tasoja huomioivaa suuntausta, ei siis pelkästään työkalujen tai käytänteiden vaan myös johtamisen ja koko organisaatiokulttuurin näkökulmista.

Arviointimatriiseista rakentuvan LESAT:in tarkoituksena on arvioida kokonaisvaltaisesti organisaation nykytilaa Lean-kyvykkyyden suhteen sekä asettaa tavoitteet halutulle tulevaisuuden tilalle. LESAT jakautuu kolmeen osioon, joita ovat elinkaari prosessiosio, infrastruktuurin mahdollistamisosio ja transformaatio- ja johtajuusosio.

LESAT:in käyttöön on todettu liittyvän monenlaisia ongelmia. Esimerkkeinä näistä ovat arviointitulosten tulkittamatta jättäminen, muiden toiminnanparannussuunnitelmien samanaikainen käyttö, tulosten liiallinen yksinkertaistaminen sekä ulkoisen ympäristön jättäminen huomiotta.

5 LEAN-OHJELMISTOTUOTANNON ARVIOINTI

Tässä luvussa kerrotaan, mitä Leanin mukaisen ohjelmistotuotannon arvioinnilla tarkoitetaan ja millaisia apuvälineitä on tarjolla. Esitys on jaettu kahteen osaan. Ensiksi tarkastelun kohteena on koko ohjelmistotuotanto. Siinä kerrotaan, miten edellisessä luvussa esitettyä LESAT-työkalua voidaan käyttää ohjelmistotuotannossa. Toiseksi käsitellään Leanin mukaisen toiminnan arviointia käyttämällä rajauksena ohjelmistokehitystä tietyssä projektissa. Luku päättyy yhteenvetoon.

5.1 LESAT ohjelmistotuotannossa

Ohjelmistotuotannossa on ollut pitkään käytössä prosessin parannus- ja arviointimalleja, tunnetuimpia niistä ovat CMMI-DEV (CMMI-Product-Team, 2006) sekä ISO/IEC 15504 (ISO/IEC, 2003). Ne ovat muodostuneet standarditavoiksi arvioida organisaatiota ja sen mahdollisuuksia. CMMI-DEV V1.3 (CMMI-Product-Team, 2006) sisältää tuen myös ketterille kehitysmenetelmille. Näitä malleja ei ole kuitenkaan alun perin suunniteltu ohjaamaan Lean-transformaatiota ja tästä syystä ne soveltuvat tähän tehtävään heikosti. Lean-ohjelmistotuotantoa läheisempiä arviointimenetelmiä, jotka keskittyvät ketterien kehitysmenetelmien transformaation arviointiin, on myös julkaistu viime vuosina. Näistä tunnetuimpia ovat esittäneet Sidky ja Arthur (2007), Qumer ja Hendersson-Sellers (2008), Sureshchandra ja Shinivasavadnahani (2008) sekä Kettunen (2012). Nämä mallit eivät kuitenkaan erityisesti huomioi arviointia Leanin näkökulmasta (Karvonen ym., 2012).

Edellä on kuvattu LESAT-arviointivälinettä ja mainittu sen olevan käytössä monilla teollisuuden aloilla. Koska LESAT on vielä varsin uusi ja vähän tutkittu, etenkin ohjelmistotuotannossa, Karvonen ym. (2012) ovat kehittäneet aiemmista alan tutkimuksista johtaen oman mallinsa LESAT:in käyttöön ohjelmistotuotannossa. Se pyrkii täyttämään aiempien mallien puutteelliseksi jättämät ohjelmistotuotannon ja Leanin erityispiirteet. Seuraavaksi kuvataan Karvo-

sen ym. (2012) esittämää LESAT-muunnelmaa, jota kutsutaan termillä LESAT for Software. Tämän jälkeen sitä verrataan muihin arviointimenetelmiin. Lopuksi tarkastellaan muunnelman soveltuvuutta Lean-ohjelmistotuotannon arviointiin.

5.1.1 LESAT for Software

LESAT for Software -malli kehitettiin prosessilla, jossa ensiksi analysoitiin Lean- ja LESAT-perusperiaatteet. Tämän jälkeen LESAT mukautettiin ohjelmistotuotannon mukaiseksi sekä lisättiin kommentit ja kuvaukset ISO/IEC 12207-prosesseille sekä Leanin ja ketteränkehityksen käytänteille ja pääperiaatteille. Viimeisenä ohjelmistotuotantoon suunnattu LESAT arvioitiin. (Karvonen ym., 2012.)

Alkuperäistä LESAT-mallia tutkittaessa nousi viittauksissa esille neljä avainkäsitettä: laajennettu organisaatio, arvovirta, sidosryhmä sekä tavaran-toimittajat (toimittajaketju, toimittajaverkosto ja avaintoimittajat). Näin ollen tutkittiin, voitaisiinko näitä avainkäsitteitä soveltaa myös ohjelmistotuotannon näkökulmasta. (Karvonen ym., 2012.)

Laajennetun organisaation hyödyntäminen painottaa kykyä laajentaa Leanin käyttö kattamaan koko arvovirta. Karvonen ym. (2012) määrittelevät MIT:n (2001) mukaisesti laajennetun organisaation tarkoittavan kaikkea liiketoimintaa arvovirrassa, joka on luomassa arvoa asiakkaalle. Vastaavasti Karvonen ym. (2012) määrittelevät MIT:n (2001) dokumentaation mukaisen arvovirran tarkoittavan niitä toimenpiteitä, joita vaaditaan jonkin tietyn tuotteen suunnitteluun, tilaamiseen ja toimittamiseen aina konseptista julkaisuun, tilauksesta toimittamiseen sekä raakamateriaaleista asiakkaan haltuun. Läheisesti laajennettuun organisaatioon liittyvä käsite ekosysteemi on ollut usein esillä yritysmaailmassa viimeisten vuosien aikana. Laajennettu organisaatio voidaan määrittellä myös koostuvan B2B (Business-to-Business) -toimitusketjuista sekä asiakkaista. (Karvonen ym., 2012.)

Arvovirralla on erityinen tarkoitus Lean-kirjallisuudessa. Tästä huolimatta tulee huomioida, että arvovirran implementointi on aina tilanteesta riippuvainen. Karvonen ym. (2012) määrittelee Messerschmittin ja Szyperskin (2003) mukaisesti ohjelmistotuotannon arvoketjun koostuvan seuraavista toimijoista: sovellusohjelmiston tuottaja (application software provider), sovelluspalvelun tuottaja (application service provider), infrastruktuuripalvelun tuottaja (infrastructure service provider), järjestelmäintegraattori (system integrator), liiketoimintakonsultti (business consultant), teollisuuskonsultti (industry consultant), infrastruktuuriohjelmiston toimittaja (infrastructure software supplier), loppukäyttäjäorganisaatio (end-user organization) ja informaatioisisällön toimittaja (information content supplier). Kaikki edellä mainitut organisaatiot voidaan nähdä mahdollisina organisaatioina laajennetussa organisaatiossa ohjelmistotalalla. (Karvonen ym., 2012.)

MIT:n (2001) dokumentaation mukaisesti *sidosryhmällä* tarkoitetaan kaikkia niitä, joilla on kiinnostusta organisaatiota, sen toimintoja tai tuloksia koh-

taan. Näihin voidaan sisällyttää esimerkiksi asiakkaat, partnerit, työntekijät, osakkeenomistajat, omistajat, valtio ja lainlaatijat. Sidosryhmän määritelmä voi vaihdella asiayhteydestä riippuen (Karvonen ym., 2012). Karvonen ym. (2012) määrittelevät Rajalan, Rossin ja Tuunaisen (2003) mukaisesti ohjelmistoliiketoiminnan tyypillisiksi sidosryhmiksi kehittämiskumppanit (development partners), jakelijat (distributors), työntekijät (employees), rahoittajat (financiers), johdon (management), osakkeenomistajat (shareholders), alihankkijat (subcontractors) sekä toimittajat (suppliers).

LESAT for Software -malli rakentuu kolmen arviointiosion ympärille. Arviointiosiot ovat MIT:n (2001) pohjalta ohjelmistotuotannon tarpeisiin räätälöidyt *Lean transformaatio / -johtajuus, elinkaari ja infrastruktuurin mahdollistaminen*. Nämä kolme eri osiota sisältävät yhteensä 54 arviointikohtaa. Seitsemän yhteensä 54:stä alkuperäisen LESAT-dokumentaation arviointikohdasta muokattiin vastaamaan ohjelmistoalan sanastoa ja prosesseja. Alkuperäisistä LESAT-arviointikohdista 87 % säilyi ennallaan, koska näiden katsottiin sopivan myös ohjelmistotuotantoon. Suurimmat muutokset toteutettiin LESAT:in elinkaari-prosessiosioon johtuen suurista eroista teollisuustuotannon (tuotanto- ja valmistusprosessien) ja ohjelmistotuotannon tuotekehityksen prosessien välillä. Lean-transformaatiota / johtajuutta sekä infrastruktuurin mahdollistamista koskevat osiot ovat miltei suoraan sovellettavissa ohjelmistoalan tarpeisiin. (Karvonen ym., 2012.)

Elinkaari-prosessiosio muokattiin ISO/IEC 12207 (ohjelmistoelinkaari-prosessien) -standardin (ISO/IEC, 2008) muotoon, koska sitä käytetään laajasti ohjelmistoalalla. Jokainen LESAT-arvioinnin osa uudelleen arvioitiin käyttäen ISO/IEC 12207-standardin mukaisia määritelmiä ja muokattiin tarvittaessa vastaamaan ohjelmistoprosessien terminologiaa. Näin ollen tuotanto- ja valmistusprosessit käännettiin ohjelmistointegrointiprosesseiksi. (Karvonen ym., 2012.)

Useita kommentteja lisättiin selventämään ohjelmistoalaa sekä Leanin ja ketterien kehitysmenetelmien mukaista toimintaa koskien niin Leanin mukaisen käytänteiden, mittareiden kuin kyvykkyytasojenkin arviointia ohjelmistokehityksen näkökulmasta. Esimerkiksi Scrum esitettiin valinnaisena tapana kehittää Lean-rakenteita ja -käyttäytymistä ohjelmistoyrityksessä. (Karvonen ym., 2012.)

5.1.2 LESAT for Softwarin vertailu muihin arviointimenetelmiin

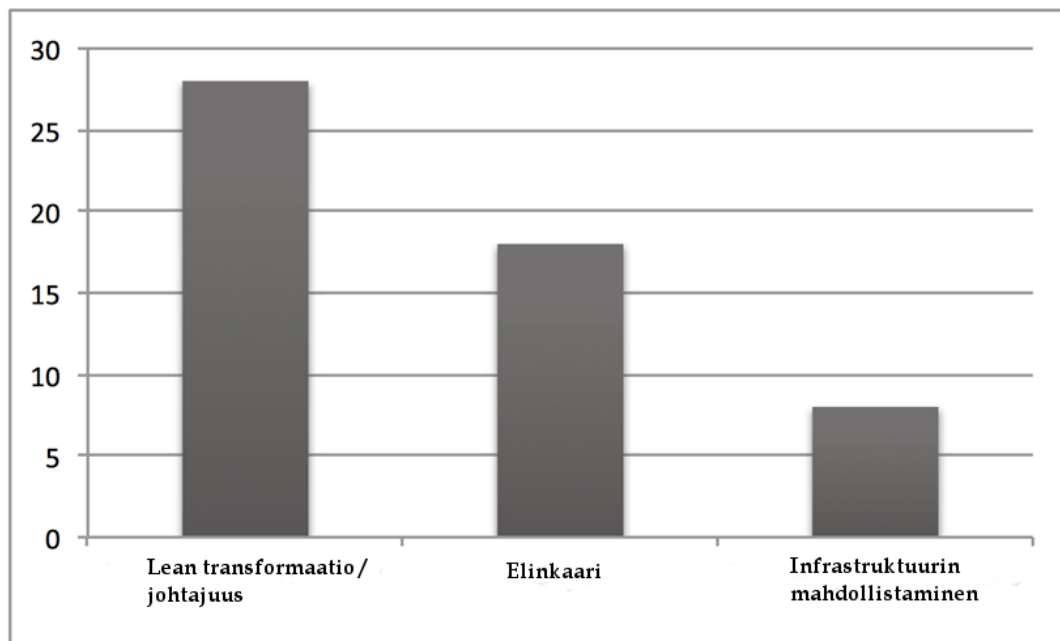
Jotta ohjelmistotuotannolle muokattua LESAT:ia (LESAT for Software) voitiin arvioida, Karvonen ym. (2012) vertasivat sitä muutamiin tunnetuimpiin arviointimalleihin. Ensisijaiseksi vertailun kohteeksi valittiin Ericssonin omiin tarpeisiinsa kehittämä Lean Amplifier -malli. LESAT for Software -mallia verrattiin ensin Lean Amplifier -malliin ja tämän jälkeen Womackin ja Jonesin (1996) viiteen Lean-periaatteeseen sekä Poppendieckin ja Poppendieckin (2006) Lean-ohjelmistotuotannon periaatteisiin. Ericsson on kansainvälinen tietoliikennealan yritys, joka oli tutkimuksen aikana keskellä muutosprosessia kohti Lean-

käytänteiden mukaista toimintaa. Näin ollen Ericssonia voidaankin pitää varhaisena Lean-ohjelmistokehityksen omaksujana. (Karvonen ym., 2012.)

Ericssonin Lean Amplifier-malli on yrityksen sisäisesti suunniteltu malli, jolla voidaan seurata tiimien edistymistä kohti ketteriä ja Leanin mukaisia työkentelytapoja. Vaikka Ericssonin Lean Amplifier on suunniteltu erityisesti heidän omiin tarpeisiinsa, se valittiin käytännölliseksi verrokiksi, koska Ericssonilla on vahva tausta Lean-käytänteiden soveltamisessa oikeassa ohjelmistotuotantokontekstissa. (Karvonen ym., 2012.)

Itse arviointi oli kaksivaiheinen, jossa arvioitiin sitä, miten LESAT voi tukea Lean-transformaatiota ohjelmistoyrityksissä, sekä kerättiin ideoita LESAT for software -mallin jatkokehitystä varten. Rakenteeltaan LESAT for Software ja Lean Amplifier ovat hieman erityyppiset ja painottavat arviointiosioita erilailla. LESAT:in 54:stä arviointikohdasta yhteensä 28 kappaletta arvioi Lean-transformaatio- ja johtamisprosesseja. Tämä edustaa yhteensä 52 % koko mallin kaikista kohdista. Ericssonin Lean Amplifier koostuu yhteensä 167:stä arviointikohdasta, jotka jakautuvat kahdeksaan eri osioon. Lean Amplifier -mallin arviointiosiot ovat *Scrum master tai team coach, tuotteen omistaja, kehittäjä, tiimi, agile coach, organisaatio, johtaja* sekä *julkaisun ja portfolion johtaminen*. (Karvonen ym., 2012.)

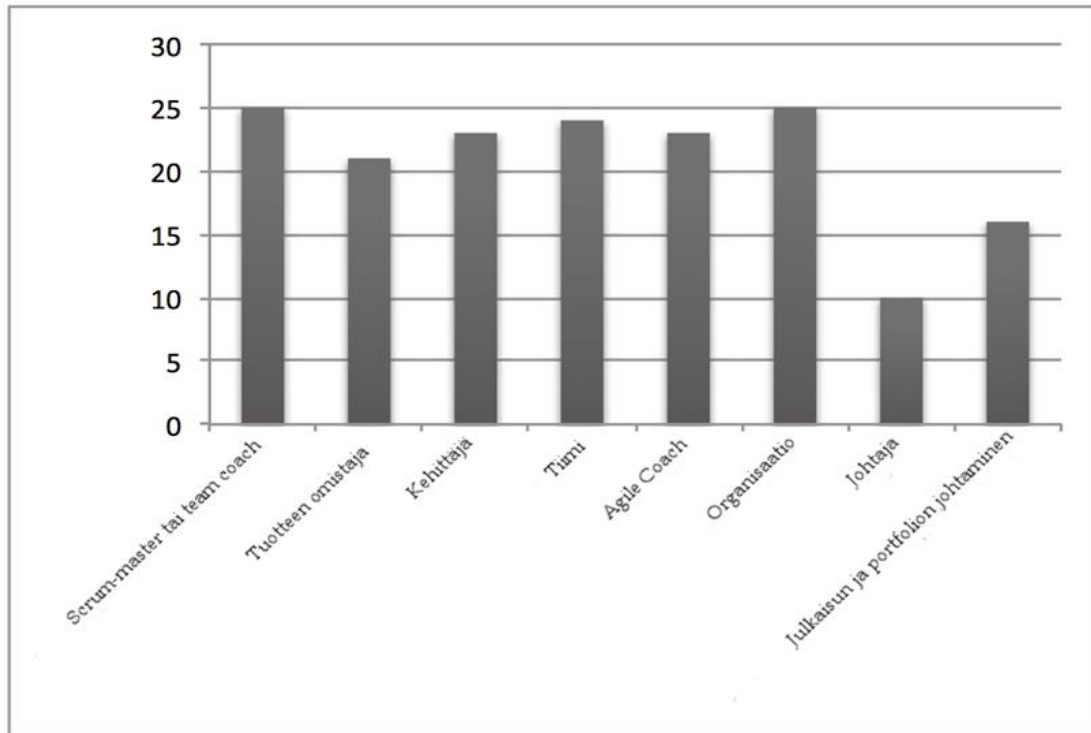
Tarkemmin kuvattuna Karvonen ym. (2012) esittävät LESAT for software ja Lean Amplifier -arviointimallien pääosiot ja niiden arviointimenetelmien painoarvot graafisessa muodossa kuvioiden 6 ja 7 mukaisina.



KUVIO 6 LESAT for software -mallin arviointikohtien määrät (Karvonen ym., 2012, 271)

Vaikka Ericssonin mallissa painotetaan tiimin ja organisaation arviointikohtia Leanin ja ketterien menetelmien käyttöönotossa, arviointitulokset kuitenkin osoittavat, että LESAT for software -malli kattaa paremmin koko organisaation.

Huomion arvoista on myös näiden kahden mallin ero johtaja- tai johtajuus-arviointikohtien osalta, sillä Lean Amplifier -mallissa johtajaa koskeva arviointiosio sisältää vähiten arviointikohtia, kun taas vastaavasti LESAT for software-mallin arviointikohtien enemmistö kuuluu johtajuus / Lean-transformaation alle. (Karvonen ym., 2012.)



KUVIO 7 Lean Amplifier -mallin arviointikohtien määrät (Karvonen ym., 2012, 271)

LESAT for Software ja Ericssonin Lean Amplifier -malleja verrattiin myös teollisuuden ja Lean-ohjelmistotuotannon perusteoksina pidettyjen Womackin ja Jonesin (1996) sekä Poppendieckin ja Poppendieckin (2006) nimeämiin Leanin ja Lean-ohjelmistotuotannon pääperiaatteisiin. (Karvonen ym., 2012.)

LESAT for software ja Lean Amplifier -mallien vertailu Womackin ja Jonesin (1996) viiteen Lean-periaatteeseen osoitti, että nämä eroavat arvon ja arvovirran painotuksessa toisistaan. Suurin osa Lean Amplifierin arviointikohdista tunnistettiin kuuluviksi virtauksen tai täydellisyyden periaatteiden alle. Nämä periaatteet voidaan nähdä joko yksilö- tai tiimityön menetelminä, jatkuvan parantamisen mentaliteetin rakentamisena tai nykyisen työtavan haastamisena. Molemmissa arviointimalleissa havaittiin vain harvoja arviointiperiaatteita, jotka voitiin lukea Womackin ja Jonesin (1996) imuperiaatteeseen liittyviksi.

Verrattaessa LESAT for software -mallia ja Lean Amplifier -mallia Poppendieckin ja Poppendieckin (2006) seitsemään Lean-ohjelmistotuotannon pääperiaatteeseen havaittiin, että useita periaatteita ei voitu tunnistaa lainkaan LESAT for Software -mallissa. Tämä tulos oli odotettavissa, koska alkuperäistä LESAT-mallia ei ole alun perin suunniteltu ohjelmistotuotantoa ajatellen, eikä vastaamaan Poppendieckin ja Poppendieckin (2006) näkemyksiä Lean-ohjelmistotuotannosta. Kuitenkin tämä tulos osoittaa selkeän tarpeen jatko-

analyysille ja LESAT for Software -mallin kehittämiseksi vastaamaan Poppendieckin ja Poppendieckin (2006) Lean-ohjelmistotuotannon periaatteita. (Karvonen ym., 2012.)

Kaiken kaikkiaan tehdyt vertailut osoittivat LESAT for Software -mallin vahvuudet ja heikkoudet. Heikkouksina voidaan nähdä Poppendieckin ja Poppendieckin (2006) näkemysten mukaisten Lean-ohjelmistotuotannon pääperiaatteiden vähäinen yhtäläisyys LESAT for Software -mallin kanssa, jonka vaikutukset näkyvät merkittävästi LESAT:in elinkaari-prosessiosiossa. Vahvuuksina voidaan sanoa olevan LESAT for Software -mallin keskittyminen arvon määrittämiseen ja koko organisaation arvovirran optimointiin. (Karvonen ym., 2012.)

5.1.3 LESAT for Softwaren soveltuvuus Lean-ohjelmistotuotannon arviointiin

Karvonen ym. (2012) määrittelevät Nightingalen ja Mizen (2002) mukaisesti LESAT:in suunnittelun perustaksi kolme vaatimusta: 1. arvioida Leanin mukaisen toiminnan tasoa organisaatiossa ja kaikissa sen ydinprosesseissa, 2. tuottaa palautetietoa ja ohjausta seuraavia parannusaskelaita varten ja 3. perustua dataan ja dokumentoitaviin todisteisiin. Karvonen ym. (2012) mukaan on täysin mahdollista todeta, että oikein käytettynä MIT:n LESAT voi tukea Lean-transformaatiota myös ohjelmistoalalla toimivissa organisaatioissa.

LESAT:in kolmeen osioon liittyviä tuloksia analysoitaessa on huomionarvoista, että infrastruktuurin mahdollistamiseen ja organisaation johtamiseen liittyvät osiot, joiden päälle elinkaari-prosessit rakentuvat, ovat miltei täysin alasta riippumattomia. Tämän seurauksena arviointikohdat, jotka oli suunniteltu arvioimaan strategista suunnittelua, strategista kumppanuutta, transformointi johtamista tai henkilöstöresursseja säilytettiin ennallaan, kuten ne olivat alkuperäisessä LESAT-dokumentaatioissa. Elinkaari-prosessiosio, joka sisältää prosessit, jotka määrittävät suoraan asiakkaille tuotettavan arvon, vaatii laajempaa muokkausta. Muokkaukset toteutettiin ISO/IEC 12207 (ISO/IEC, 2008) -standardin ja yleisen Lean-ohjelmistotuotannon tietämyksen perusteella. (Karvonen ym., 2012.)

LESAT painottaa arvioinnissa vahvasti arvon ja arvovirran johtamista. Tämä on tärkeää, koska Lean on pääasiassa keskittynyt siihen, kuinka saada organisaatiot tuottamaan mahdollisimman paljon asiakasarvoa. (Karvonen ym., 2012.)

LESAT korostaa koko arvovirran huomioimista transformoitaessa toimintaa kohti Leanin mukaista toimintaa ja sisältää tähän liittyen arviointikohtia liiketoiminnalle arvovirran koko matkalla. Näin ollen LESAT for Software kykenee ohjaamaan koko organisaation transformointia kohti Lean-organisaatiota, sen sijaan että se ohjaisi vain tiettyjä organisaation osia. (Karvonen ym., 2012.)

LESAT for Software korostaa myös johtajuuden roolia. Johtajuus on erittäin tärkeää, koska se mahdollistaa tarvittavat resurssit jätteen hävittämiseksi

sekä virtaavan arvovirran saavuttamiseksi. LESAT for software -mallin johtajuus / Lean-transformaatio -osio voi auttaa alkuun ja edistää transformaatiota kohti Lean-ohjelmistotuotanto-organisaatiota. (Karvonen ym., 2012.)

Karvosen ym. (2012) tuloksiin perustuen voidaan todeta, että LESAT for software voi selkeästi olla avuksi ja ohjata organisaation transformaatiota kohti Leanin mukaista ohjelmistotuotantoa. Tulee kuitenkin huomata, että mallissa on vielä puutteita ja jatkokehitystä tarvitaan erityisesti LESAT for Software -mallin elinkaari-prosessiosion sovittamiseksi nykyisten tunnettujen Lean-ohjelmistotuotantokäytänteiden, kuten Poppendieck ja Poppendieck (2006) periaatteiden, mukaiseksi.

5.2 Lean-ohjelmistokehityksen arviointi

Tässä aluvussa käsitellään Leanin mukaisen toiminnan arviointia ohjelmistokehityksen näkökulmasta tunnettujen menetelmien ja yksittäisten mittareiden avulla. Aluvussa esitellään myös ohjelmistojen ylläpidon arviointia Leanin mukaisesti.

5.2.1 SPI-LEAM

Lean-ohjelmistokehitys sisältää samankaltaisia periaatteita kuin ketterät menetelmät sellaisilla alueilla kuin ihmisten johtamisessa ja johtajuudessa, laatuun keskittymisessä, teknisessä erinomaisuudessa sekä tiheässä ja nopeassa arvon toimituksessa asiakkaalle. Lean kuitenkin erottuu ketteristä menetelmistä keskittymisellä päästä päähän (end-to-end) -näkökulmaan, eli arvon virtaamisessa koko kehityksen läpi ensimmäisistä konseptiluonnoksista ja ideoista aina ohjelmisto-ominaisuuksien toimittamiseen asiakkaalle. (Petersen & Wohlin, 2010a.)

Päästä päähän -näkökulman tueksi Lean esittää useita työkaluja arvovirran analysointia ja parantamista koskien. Näitä työkaluja ovat muun muassa arvovirran kartoitus, varastonhallinta sekä imujärjestelmät kuten Kanban. Arvovirran kartoitus kuvaa kehityselinkaaren ja osoittaa käsittely- sekä odotusajat. Varastonhallinta tähtää rajoittamaan keskeneräisen työn määrää, sillä osittain tehty työ ei tuota arvoa. Varastonhallintamenetelmiä ovat rajoitteiden teoria (theory of constraints) sekä jonotusteoria (queuing theory). Työntöjärjestelmät allokoivat aikaa vaatimuksille reilusti etukäteen ja usein ylikuormittavat kehitysprosessia työllä. Päinvastaisesti imujärjestelmät mahdollistavat sen, että ohjelmistotiimit ottavat uusia tehtäviä aina, kun heillä on vapaata kapasiteettia. Korkeamman prioriteetin tehtävät tulisi hoitaa ensin mikäli mahdollista. (Petersen & Wohlin, 2010a.)

Ohjelmistokehitysprosessien parantaminen tähtää ohjelmistoprosessien tehostamiseen ja tuotteen laadun kasvattamiseen jatkuvalla arvioinnilla ja prosessin säätämällä. Tätä varten on luotu useita prosessinkehitysmalleja, joista

tunnetuimpia ovat Capability Maturity Model Integration (CMMI) (CMMI-Product-Team, 2006) ja Quality Improvement Paradigm (QIP) (Basili, 1985; Basili & Green, 1994). Nämä ovat korkean tason malleja, jotka tarjoavat ohjausta siihen, mitä tehdä, mutta eivät kerro suoraan sitä, miltä varsinaisen toteutuksen tulisi näyttää. (Petersen & Wohlin, 2010b.)

Petersen ja Wohlin (2010b) ovat esitelleet SPI-LEAM (Software Process Improvement through the Lean measurement) -mallin, joka yhdistää ohjelmiston laadun parantamisen ajatusmallin Lean-ohjelmistokehityksen periaatteisiin. SPI-LEAM kuvaa tavan, miten implementoida Lean-periaatteet mittaamisen kautta siten, että saadaan aikaan parannusta ohjelmistoprosesseissa. (Petersen & Wohlin, 2010b.)

SPI-LEAM auttaa organisaatioita lähestymään Lean-ohjelmistokehitysprosessien parantamista jatkuvan parantamisen kautta. Malli perustuu erilaisten varastojen mittaamiseen sekä yhdistettyihin analyysihin varastojen mittaamisesta. Varastot osoittavat helposti Lean-käytänteiden puuttumista ja näin ollen toimivat hyvin myös perusteluina Lean-periaatteiden käyttöönotosta keskusteltaessa. Varastoja analysoitaessa voidaan käyttää apuna järjestelmäajattelumallia, koska Lean-ajattelu vaatii kokonaisvaltaisen näkemyksen oikeiden ongelmien löytämiseksi. Näin ollen mallissa ei katsota ainoastaan kehitysprosessin yksittäisiä osia, vaan pikemminkin ongelmien tai parannusideoiden vaikutusta kokonaistilannetta ja -prosessia ajatellen. (Petersen & Wohlin, 2010b.)

SPI-LEAM -malli perustuu Basilin (1995) QIP-malliin, mutta se on muokattu vastaamaan ohjelmistoprosessien tarpeita. SPI-LEAM -malli on iteratiivinen, joten paluu edelliseen kohtaan ja prosessin osion toistaminen on mahdollista aina tarvittaessa. (Petersen & Wohlin, 2010b.) Petersenin ja Wohlinin (2010b) mukaan malli koostuu seuraavista kuudesta vaiheesta. Ensimmäisenä vaiheena on *Karakterisoi nykyistä projektia*. Vaiheen yhteydessä kuvataan projektin luonne ja ympäristö sekä sovellusalue, prosessikokemus, prosessiosaaminen ja ongelmarajoitteet.

Toisena vaiheena on *Aseta kvantitatiiviset tavoitteet ja mittarit* (Set Quantifiable Goals and Measurements). Tavoitteita ja niihin liittyviä mittareita ajatellen on suositeltavaa käyttää Goal-Question-Metric (GQM) (Basili, 1985) -mallia. Mallin käytön tavoitteena mahdollistaa jatkuva ohjelmistoprosessien parantaminen, joka johtaa Lean-ohjelmistoprosesseihin. Sovellettaessa mallia käytäntöön on tärkeää, että mallin tavoitteet vastaavat alaa. Tavoitteiden saavuttamiseksi tulisi vastata kahteen avainkysymykseen: 1. mikä on kehitysprosessin suorituskyky varastojen näkökulmasta, ja 2. mikä aiheuttaa suorituskykyongelmat? Ensimmäisen kysymyksen vastauksen tulisi kuvata heikkoa suorituskykyä eri varastojen näkökulmasta. Suurien varastomäärien tunnistaminen auttaa toimeenpanemaan parannukset, joiden tavoitteena on varastomäärien pienentäminen ja tätä kautta niiden aiheuttamien ongelmien välttäminen. Toinen kysymys tähtää tunnistamaan syyt sille, miksi varastomäärät ovat korkealla. Tämä on tärkeää oikeiden parannusten toimeenpanemista ajatellen. Kehitysprosessin varastojen ymmärtämiseksi tulisi kulkea koko kehitysprosessin läpi

vaihe vaihteelta ja käyttää tässä yhteydessä apuna esimerkiksi arvovirtakartoitusta jätteen tunnistamiseksi asiakkaan näkökulmasta. Tähän vaiheeseen tulisi kiinnittää huomiota, sillä kaikilla yrityksillä varastot itsessään eivät ole samoja. Ohjelmistokehitysprosessin varastoja voivat olla esimerkiksi vaatimukset, testaustapaukset, korjauspyynnöt ja viat. Ainakin yhden varaston tulisi kuvata laatu-ulottuvuutta, jotta saadaan vähennettyä riskiä laadun kustannuksella optimoitavista mittaustuloksista. (Petersen & Wohlin, 2010b.)

Kolmas vaihe on *Valitse prosessimallit ja -menetelmät*. Tässä vaiheessa päätetään projektin ja ympäristön piirteiden perusteella, mitä prosessimallia (esim. vesiputous, XP, Scrum) käytetään. Organisaation ollessa erittäin dynaaminen tulisi valita ketterä malli. Prosessimallin käyttöä ei tarvitse kuitenkaan aina aloittaa täysin alusta, vaan tulisi huomata että useissa organisaatioissa on käytössä malleja, jotka kaipaavat jatkokehitystä. Käyttöön voidaan valita myös modifioitu prosessimalli tai -menetelmä ja työkalut, joilla tuetaan prosesseja. Esimerkiksi mikäli halutaan saada aikaan enemmän Leanin mukainen prosessi, eräs vaihtoehto voisi olla pilkkoa työ pienempiin osiin, jotta se läpäisisi kehitysprosessin nopeammin. (Petersen & Wohlin, 2010b.)

Neljäs vaihe on *Toteuta prosessit ja kerää sekä tarkasta kerätty data*. Prosessia suorittaessa tallennetaan mittaustulokset. SPI-LEAM:in yhteydessä varastomäärien tasoa tulee tarkkailla jatkuvasti. Tiedon keräämisen jälkeen se tulee tarkastaa, jotta voidaan varmistua sen eheydestä ja tarkkuudesta. (Petersen & Wohlin, 2010b.)

Viides vaihe on *Analysoi kerätty data ja suosittele parannuksia*. Varastomäärien suoritustehokkuuden parantamiseksi on tärkeää löytää taustasyyn korkeille varastomäärille. Yksittäisen tilanteen ja parannustoimien arvioimisen ymmärtämiseksi järjestelmäajattelumenetelmät (system thinking methods) ovat osoittautuneet hyviksi. Korkeiden varastomäärien aiheuttavan syyn löydyttyä tulee itse prosessiin tai käytössä oleviin menetelmiin tehdä korjaava muutos. (Petersen & Wohlin, 2010b.)

Viimeinen vaihe on *Pakkaa ja varastoi saavutetut kokemukset*. Vaiheen tehtävänä on säilöä saavutetut kokemukset ja kerätty data, jotta tulokset eivät katoa ja niitä voidaan käyttää tulevaisuudessakin. (Petersen & Wohlin, 2010b.)

SPI-LEAM on mukautuva, sillä se sallii organisaatioiden valita varastot ja käytettävät analyysimenetelmät, jotka sopivat heidän tarpeisiinsa parhaiten. SPI-LEAM -mallin mukautuvuuden ansiosta sitä voidaan soveltaa useissa eri konteksteissa. SPI-LEAM kaipaisi jatkotutkimusta muun muassa siitä, miten se eroaa erilaisten ketterien kehitysmenetelmien kuten Scrumin ja XP:n välillä. Toinen jatkotutkimuksen tarve olisi tutkia, miten SPI-LEAM toimii eri kompleksisuustasoilla, kuten esimerkiksi suurilla, useiden kehitystiimien tuottamien ohjelmistotuotteiden yhteydessä verrattuna pieniin vain muutamien kehitystiimien ohjelmistotuotteisiin. (Petersen & Wohlin, 2010b.)

5.2.2 Muita arvioinnin mittareita

Lähdemateriaaleissa ei ole aina Leanin mukaista ohjelmistotuotantoa mitattaessa käytetty jotain tiettyä arviointimallia tai -viitekehystä. Monet tutkimukset tarkastelevat arviointia yhden tai useamman peruseriaatteen näkökulmasta ja vertaavat organisaation ohjelmistokehitysprosessien nykytilannetta näihin. Seuraavassa käydään läpi tunnetuimpia ja käytetyimpiä yksittäisiä mittareita Leanin mukaisten ohjelmistokehitysprosessien arviointiin.

Jätteen minimointi

Ohjelmistokehityksen mukaisen Lean-ajattelun ensimmäisiä läpimurtoaskelia on oppia näkemään, mikä on jätettä. Mikäli jokin ei lisää suoraan arvoa, se on jätettä. Mikäli on olemassa suora tapa tehdä jotain ilman tiettyä vaihetta, on kyseessä jäte. (Poppendieck, 2002.)

Ohjelmistokehitykseen on periytynyt teollisuustuotannon Lean-ajattelusta työntekijöitä vastuunottamiseen rohkaiseva ja motivoiva lause: "Tee se oikein jo ensimmäisellä kerralla" (Do It Right the First Time), joka on ymmärretty ohjelmistokehityksen puolella usein täysin väärin. Ohjelauseetta on pidetty perusteenä ja tekosyynä sille, että on voitu tuottaa valtavia määriä dokumentaatiota ja käyttää suurta joukkoa suunnittelijoita kehittäjiä ja asiakkaan välissä, vaikka tämän tyyppistä toimintaa voidaan pitää suoraan massatuotantoperiaatteiden mukaisena arvoa tuottamattomana jätteenä. (Poppendieck, 2002.)

Oikeampana käännökseenä ohjelauseelle voitaisiin pitää: "Testaa ensin". Tämä voidaan nähdä siten, että koskaan ei tulisi kirjoittaa koodia ymmärtämättä, mitä sen tulisi tehdä, eikä keinona nähdä toimiiko koodi. Hyvä ymmärrys alasta yhdistettynä lyhyisiin kehityssykleihin ja automaattiseen testaamiseen ovat perustana sille, kuinka ohjelmistokehittäjät "Tekevät sen oikein jo ensimmäisellä kerralla". (Poppendieck, 2002.)

Virtauksen nopeus

Lean-ohjelmistokehityksessä ideana on maksimoida informaation ja toimitetun arvon virtaus. Samoin kuin teollisuustuotannon puolellakin virtauksen maksimointia ei tule ymmärtää prosessien ja prosessivaiheiden pelkkänä automatisointina. Pikemminkin virtauksen maksimointi tulisi ajatella siirtojen rajoittamisena. Siirrot pitäisi pystyä minimoimaan määrällisesti sekä etäisyyksiltään mahdollisimman lyhyiksi, niin että käytössä on mahdollisimman suuri viestintäkapasiteetti niin myöhään kuin mahdollista. Valtavien jäädytettyjen dokumentaatiokasojen siirtely on massatuotantomentaliteettia. Lean-ohjelmistokehityksessä ideana on minimoida niin paljon dokumentaatiota ja siirtoja kuin mahdollista. Dokumentit, jotka eivät ole asiakkaalle hyödyllisiä korvataan automatisoiduilla testeillä, jotka varmistavat, että asiakasarvoa toimitetaan nyt ja tulevaisuudessa aina kun väistämättömiä muutoksia tarvitaan. (Poppendieck, 2002.)

Lean-ohjelmistokehityksellä tarkoitetaan nopeaa, juuri oikeaan aikaan toimitettua arvoa. Lean-ohjelmistokehityksessä avain nopean toimituksen saavuttamiseksi on ongelman jakaminen pienempiin palasiin, jotka imuohjautuvat asiakaskertomukseen ja -testaukseen. Tehokkain yksittäinen mekanismi Lean-ohjelmistokehityksen implementoinnissa on pienten, oikeaa liiketoiminta-arvoa sisältävien palasten toimittaminen lyhyiden aikaikkunoiden sisällä. (Poppendieck, 2002.)

Maglyasin ym. (2012) mukaan Lean-käytänteiden mukaisella virtauksen implementoinnilla saadaan lyhennettyä pitkiä julkaisusyklejä. Lean-käytänteet tekevät kehitysprosessista ennustettavampia implementoitujen ominaisuuksien suhteen, sillä iteraatiot ovat lyhyitä ja implementoitavat ominaisuudet ovat tiedossa jo iteraation alussa. Uusien ominaisuuksien tuominen keskeneräiseen iteraatioon on kiellettyä. Tämä tieto auttaa muita osastoja ja sidosryhmiä kuten asiakasta ja markkinointia, sillä he voivat aloittaa oman työnsä aiemmin jo tiedossa olevien ominaisuuksien perusteella, ilman että heidän on tarvetta odottaa iteraation loppuun. Näin ollen tuotteen toimitusaika markkinoille lyhenee. (Maglyas ym., 2012.)

Myös Middleton ja Joyce (2010) käyttivät tutkimuksessaan yhtenä Leanin mukaisen toiminnan arviointimittarina läpimenoaikaa. Middletonin ja Joycen (2010) mukaan tutkittavassa yrityksessä Leanin ja Kanbanin käyttöönoton myötä vuoden mittaisella testijaksolla läpimenoaikaa saatiin alennettua ja ohjelmistoja toimitettua lopulta keskimäärin 37 % nopeammin.

Arvonmuodostus

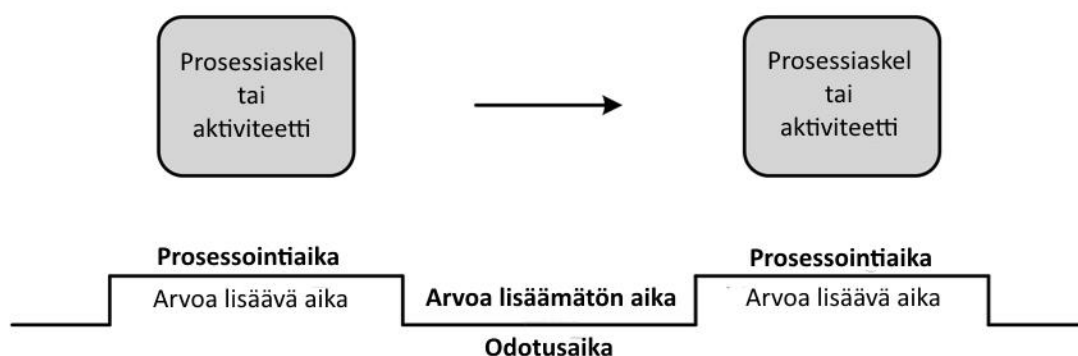
Ohjelmistoja sisältävissä tuotteissa, kuten autoissa ja telealan järjestelmissä, ohjelmistot on perinteisesti nähty kustannuserinä, eikä niiden arvoa ole ymmärretty suhteessa koko tuotetarjontaan. Tänä päivänä ohjelmistot ovat koko ajan yhä suurempi osa kilpailuetua markkinoilla ja tuotedifferointia laitteistojen (hardware) muuttuessa yhä standardoidummiksi. Näin ollen huomio kiinnittyy yhä tiiviimmin ja kriittisemmin ohjelmistojen arvoon. (Khurum, Gorshek & Wilson, 2012.)

Lean-ajattelu painottaa mittareiden käyttöä arvioinnissa ja prosessien sekä koko arvovirran parantamisessa. Tuotepäällikön avaintehokkuusindikaattorien (key performance indicators) tulisi edustaa organisaation tuotekohtaisia tavoitteita numeerisessa muodossa sekä esittää koko tiimin suorituskykyä. Tuotepäällikön avaintehokkuusindikaattorit liittyvät enemmän tuotteisiin itseensä kuin persoonallisiin piirteisiin. (Maglyas ym., 2012.)

Arvovirtakartoitus on yksi monista Lean-käytänteistä, joka ovat herättänyt viimeaikoina kiinnostusta ohjelmistotalalla. Aiemmin useilla muilla aloilla, kuten sotilas-, terveys- ja teollisuusaloilla, arvovirtakartoitusta käyttämällä on saatu aikaan todistettua huomattavia parannuksia sekä prosesseissa että tuotteissa. Arvovirtakartoituksen avulla voidaan poistaa jätettä ja optimoida kokonaisuutta. Tämän lisäksi sen avulla voidaan tunnistaa pullonkaulat, jotka aiheuttavat virtauksen hidastumista. (Khurum, Petersen & Gorshek, 2014.)

Perinteisesti arvovirtakartoituksella on keskitytty seuraamaan läpimenoaikoja, käynnistysaikoja, sykli-aikoja sekä varastomääriä. Arvovirtakartointi auttaa visualisoimaan prosessin näkemällä virtauksen päästä päähän, opastaa keskittymään jätteen lisäksi myös arvoon, edistää yhteisen kielen käyttöä, opastaa aikataulutusta ja ennustusaktiviteetteja sekä muodostaa pohjan Leanin mukaiselle tuotannolle. Menetelmällä saavutetut tulokset ovat positiivisia, ja siksi myös ohjelmistokehityksen tulisi hyödyntää niitä. On kuitenkin huomattava, että arvovirtakartointi keskittyy pääasiassa aikaperustaisiin kohtiin, mutta ohjelmistokehityksessä sekä arvo että jäte ovat monimutkaisempia. (Khurum, Petersen & Gorschek, 2014.)

Arvovirtakartoituksen idea ja notaatio ovat varsin yksikertaisia, ja ohjelmistotuotantoon niiden notaation esittelivät teoksissaan Poppendieckit (2003). Kuviossa 8 on kuvattu arvovirtakartoituksen notaatio. Prosessilaatikko kuvaa yksittäistä askelta tai aktiviteettia prosessissa, jossa työ virtaa eteenpäin. Prosessin askeleen tai aktiviteetin kuvaus on kirjoitettu laatikon sisälle. Nuoli esittää virtauksen suuntaa. Aikajana esittää prosessiaskelten sekvenssiä kronologisessa järjestyksessä kuvaamalla samalla ajan tilan, kuten arvoa lisäävä prosessointiaika tai arvoa lisäämätön odotusaika. (Mujtaba, Feldt & Petersen, 2010.).



Arvovirtakartointinotaatio

KUVIO 8 Arvovirtakartoituksen notaatio (Mujtaba, Feldt & Petersen, 2010, 140)

Arvovirtakartoituksen mittareita ovat *läpimenoaika* (lead-time), joka kuvastaa keskimääräistä aikaa, joka pyynnöllä kestää edetä päästä päähän koko prosessin läpi aina alusta loppuun saakka sisältäen myös jonottamisajat sekä aliprosessien odotusajat. *Jonotusaika* (queue time) eli odotusaika, kuvastaa keskimääräistä aikaa, jonka pyyntö odottaa paikallaan. *Prosessointiaika* (processing time) kuvastaa aikaa, jona pyyntöä työstetään yhden henkilön tai tiimin toimesta. (Mujtaba ym., 2010.)

Arvovirtakartoituksessa hyödynnetään nykytilan ja halutun tulevaisuuden tilan kuvaamista. Nykytilan kartta kuvaa prosessin tilaa nykyisessä muodossaan. Sitä käytetään lähtötietona parannuksille. Halutun tulevaisuuden tilan kartta kuvastaa taas sitä, miltä prosessin tilan halutaan näyttävän ja miltä se voi

näyttää myöhemmin, mikäli tunnistetut jätteet pystytään muutoksilla ja parannuksilla välttämään. (Mujtaba ym., 2010.)

Koodirivien määrä

Ehkä suoraviivaisin tapa arvioida Lean-ohjelmistokehityksen implementointia ja tehokkuutta on seurata tuotettujen koodirivien määrää (Lines of Code, LOC). Perera ja Fernando (2007) seurasivat tutkimuksessaan koodirivien määriä ketterien kehitysmenetelmien ja Lean-Agile hybridimenetelmän välillä. Koodirivit jaettiin kolmeen osaan: uusi LOC, muutettu LOC ja poistettu LOC. Jako osioihin toteutettiin siksi, että projektin kokonaisen LOC-määrän seuraaminen voisi helposti peittää alleen sen, mistä muutokset tulevat ja näin ollen taustasyyt voivat jäädä huomaamatta. (Perera & Fernando, 2007.)

Koodirivien lisäksi seurattiin myös korjattujen vikojen määrää, arvioitua työmäärää ja toteutunutta työmäärää. Korjattujen vikojen suhteen tulee ymmärtää kahden vertailtavan kehitysparadigman eroavaisuudet kehitystoimintojen laadun parannuksessa. Hybridilähestymisellä on odotettavissa suurempi vikojen korjausmäärä verrattuna ketteriin menetelmiin implementoinnin alkuvaiheissa. (Perera & Fernando, 2007.)

Koodirivien määrää, korjattuja vikoja sekä arvioitua työmääriä seurattiin määrällisesti sekä ajallisesti työtunneissa. Työtuntien seuraaminen on tärkeää erilaisten työmäärien jakautumisen seuraamiseksi ja vertaamiseksi kehitysmenetelmien välillä. (Perera & Fernando, 2007.)

Tutkimuksen alkuvaiheessa hybridilähestymisellä toteutetussa projektissa virheiden määrä kasvoi nopeasti varsin korkealle tasolle. Kuitenkin ajan kuluessa ja tiimin harjaantuessa virheiden määrä väheni selkeästi perinteisten ketterien kehitysmenetelmien alapuolelle, ja näin ollen voidaan todeta, että Lean-käytänteillä voidaan selkeästi parantaa ketteriä kehitysmenetelmiä. (Perera & Fernando, 2007.)

Virheiden tunnistus ja seuranta

Middleton (2001) tutki kahta koeryhmää Lean-käytänteiden mukaisessa ohjelmistokehityksessä ja sen käyttöönotossa. Tutkimus toteutettiin organisaation oikeassa liiketoimintaympäristössä oikealla ohjelmistotuotteella. Organisaation henkilöstö oli ahkeraa ja osaavaa käytettävien ohjelmointikielten ja -järjestelmien suhteen. Tästä huolimatta jopa pienetkin ohjelmistoprojektit myöhästyiivät aikatauluistaan hyvin usein ja organisaatiossa oli havaittavissa tästä johtuvaa turhautumista. (Middleton, 2001.)

Useat kehittäjät työskentelivät samojen projektien ja tuotteiden parissa ja käytänteet, tavat ja standardit vaihtelivat niin yksilöiden kuin eri tiimien välillä. Organisaatiossa oli käytössä selkeä ylhäältä alaspäin johdettava malli, ja käytössä ollut seuranta perustui viikoittaiseen talous- ja johtoraportointiin johtoryhmälle. Organisaatio ei kerännyt lainkaan tilastotietoa ohjelmistokehittäjien suorituskykyyn tai tuottavuuteen liittyen. Yrityksessä seurattiin tuntilistoja,

mutta nämä kertoivat enemmänkin paikalla olemisesta kuin suoritetusta työstä. (Middleton, 2001.)

Tutkimuksessa muutettiin kehitysprosessia Lean-periaatteiden mukaiseksi niin, että johtajalla ei ollut enää roolia laadun valvonnassa. Suunnittelijoiden (system analysts) pyydettiin välittämään työ ohjelmoijille välittömästi, kun analyysisivu tehtävästä työstä oli valmis. Analyysisivua käytettiin ohjeena niin, että prosessissa oleva työ edusti dokumentaatiota, jonka määrää haluttiin pienentää Lean-käytänteiden mukaisesti. Työntekijät eivät saaneet kerätä varastoon isoja pinoja töitä, vaan pienemmällä määrällä haluttiin saada nopeutettua virtausta. Aiemmin organisaatiossa suunnittelijat olivat saattaneet työskennellä päiviä tai viikkoja analyysisivujen parissa ennen niiden luovutusta eteenpäin kehittäjille. Lean-tekniikoiden avulla oli odotettavissa, että tietoa saadaan kerättyä ja tallennettua prosessista aina virheiden löytyessä. (Middleton, 2001.)

Ohjelmoijia pyydettiin palauttamaan välittömästi takaisin suunnittelijoille kaikki työt, jotka eivät täyttäneet standardeja tai sisälsivät virheitä. Ohjelmoijat kirjasivat virheet ylös tukkimiehen kirjanpidolla projektitiedostoon. Tiimien kokemukseen perustuen yleisimmät virhekategorioiden olivat:

- vääränlaiset nimeämiskäytännöt
- epälooginen pseudokoodi
- projektitiedoston huomiotta jättäminen
- ohjelmoija ei ymmärrä tuotettua analyysia

Ohjelmoijia opastettiin myös olemaan ottamatta muita töitä sillä aikaa, kun suunnittelija korjasi analyysitiedostoa. Ohjelmoijat saivat avustaa suunnittelijaa virheiden korjaamisessa, mikäli tähän oli tarvetta. Näin pyrittiin varmistamaan huomion kiinnittyminen prosessivirheisiin, jotta ne saataisiin ratkaistua nopeasti. Lean-paradigman mukaan toimimisessa on oleellista keskeyttää työ ja korjata prosessivirheet niiden löytyessä ja välttää virheitä sisältävien töiden kasautuminen. (Middleton, 2001.)

Molemmissa tutkittavissa tiimeissä havaittiin, että virheen sisältävät analyysitiedostot palautettiin suunnittelijoille joko välittömästi, mikäli virhe oli ilmeinen, tai useimmiten noin puolen tunnin kuluttua, kun ohjelmoija oli ehtinyt syventyä analyysin yksityiskohtiin. Tätä pidettiin turhauttavana matalan tuottavuuden ajanjaksona. Ilmiö on täsmälleen sama, jonka Ohno (1988) havaitsi Toyotalla 1940-luvulla. Womackin ym. (1990) mukaan Ohnon aloittaessa koikeilunsa tuotantotiimit pysäyttivät toimintaansa jatkuvasti ja tämä johti työntekijöiden lannistumiseen. Tiimien kokemuksen karttuessa virheiden tunnistamisessa ja jäljittämässä tämä häiritsevä ja pettymyksiä aiheuttanut vaihe kuitenkin ylitettiin ja virheiden määrä alkoi vähentyä merkittävästi. Sama ilmiö havaittiin tutkimuksen kahta ohjelmistotiimiä tutkittaessa. Alkuvaiheen pysähtelyjen jälkeen sekä suunnittelijat että ohjelmoijat tulivat työssään tarkemmiksi virheiden suhteen. (Middleton, 2001.)

Leanin mukaisen ohjelmistojen ylläpidon arviointi

Ohjelmistokehityksessä ohjelmistotuotteelle tehdään ensin vaatimusmäärittelyt, suoritetaan varsinainen ohjelmointi ja testaus. Näiden jälkeen ohjelmistotuote toimitetaan markkinoille ja siirrytään ohjelmiston ylläpitovaiheeseen. Ylläpito voidaan jakaa kahteen osaan, joita ovat lisäykset (enhancements) ja korjaukset (corrections). Lisäyksillä tarkoitetaan esimerkiksi tuotteen ominaisuuksiin ja toiminnallisuuksiin tehtäviä pieniä parannuksia sekä resurssien ja suorituskyvyn optimointia. Korjauksilla tarkoitetaan olemassa olevan ohjelmiston virheiden ja ongelmien paikkausta esimerkiksi loppuasiakkaiden raportoimien käyttöongelmien pohjalta. (Petersen, 2010, s.251.)

Ohjelmiston ylläpidon viemä osuus koko ohjelmistokehityksen vaatimasta työmäärästä on 50 % - 80 %. Näin merkittävällä osuudella työmäärästä ylläpitoa voidaan pitää olennaisena osana koko ohjelmistokehitystä. Parannukset ylläpidon prosesseissa voivat kasvattaa taloudellisuutta ja tehokkuutta aina yksittäisen ohjelmistotuotteen ja koko organisaation osalta. (Petersen, 2010, s.251.)

Petersenin (2010, s.251) mukaan Leanin tarjoamien tuotekehityshyötyjen (korkeampi laatu, nopeampi asiakastarpeisiin vastaaminen, juuri oikeaan aikaan tuottaminen) takia Lean on tullut suosituksi myös ohjelmistokehitysalalla.

Lean-ohjelmistokehitys ja ketterät kehitysmenetelmät sisältävät molemmat useita samoja piirteitä ja käytänteitä. Tällaisia ovat muun muassa itseorganisoituvat tiimit, kestävä nopeus, kasvokkain keskustelu ja fokuoituminen tekniseen erinomaisuuteen. Ketteriin kehitysmenetelmiin verrattuna Lean keskittyy vahvemmin tarjoamaan periaatteita ja käytänteitä, joiden avulla systemaattinen prosessien analysointi ja parannus olisi helpompaa saada reilusti tuottavammaksi ja taloudellisemmaksi. Esimerkkejä tällaisista käytänteistä ja periaatteista ovat muun muassa kokonaisuuden näkeminen, jätteen välttäminen ja minimointi, lyhyet syklijat ja nopeat palautteet sekä laadun rakentaminen sisään jo alusta alkaen. Periaatteita Leanissa tuetaan Lean-työkaluilla, joista parhaiten tunnetuimpia ovat arvovirtakartat, kumulatiiviset virtausdiagrammit ja Kanban. (Petersen, 2010, s.251-252.)

Vaikka näiden seikkojen valossa ohjelmiston ylläpitoa voidaan pitää merkittävänä osana ohjelmistotuotantoa kehitysmenetelmästä riippumatta, toistaiseksi ei ole saatavilla ratkaisua ohjelmiston ylläpidon analysointiin Lean-ohjelmistokehityksen näkökulmasta (Petersen, 2010, s.252).

5.3 Yhteenveto

Tässä luvussa esiteltiin Leanin mukaisen ohjelmistotuotannon ja ohjelmistokehityksen arviointia. Lean-ohjelmistotuotannon arviointimalleista esiteltiin LESAT for Software -malli, jota verrattiin Womackin ja Jonesin (1996) sekä Poppendieckin ja Poppendieckin (2006) Lean-periaatteisiin sekä Ericssonin kehittämään Lean Amplifier -malliin. LESAT for Software on Karvosen ym. (2012)

kehittämä ohjelmistotuotannon tarpeisiin kehitetty muunnelma MIT:n (2001) LESAT -työkalusta. LESAT for Software koostuu arviointimatriiseista, jotka jakautuvat kolmeen arviointiosion ympärille. Arviointiosiot ovat MIT:n (2001) pohjalta ohjelmistotuotannon tarpeisiin räätälöidyt Lean transformaatio / -johtajuus, elinkaari ja infrastruktuurin mahdollistaminen. Nämä kolme eri arviointiosiota sisältävät yhteensä 54 arviointikohtaa.

Lean-ohjelmistokehityksen arviointimenetelmistä esiteltiin Petersenin ja Wohlinin (2010b) kehittämä SPI-LEAM. SPI-LEAM koostuu kuudesta vaiheesta, ja se auttaa organisaatioita lähestymään Lean-ohjelmistokehitysprosessien parantamista jatkuvan parantamisen kautta. Malli perustuu erilaisten varastojen mittaamiseen sekä yhdistettyihin analyysiin varastojen mittaamisesta.

Luvussa esiteltiin myös muita Leanin mukaisen ohjelmistokehityksen arviointiin käytettyjä mittareita. Arviointi on yleensä ankkuroitu yhteen ja pariin yksittäiseen peruseriaatteeseen. Yleisimmät Leanin mukaisen ohjelmistokehityksen arviointiin käytetyt mittarit arvioivat kehitysprosesseja pääasiassa jätteen minimoinnin (Poppendieck, 2002), virtauksen nopeuden (Poppendieck, 2002; Maglyas ym., 2012; Middleton & Joyce, 2010), arvon sekä arvonmuodostuksen (Krurum ym., 2012; Maglyas ym., 2012; Poppendieck & Poppendieck, 2003; Mujtaba ym., 2010), koodirivien määrän (Perera & Fernando, 2007) ja virheiden tunnistuksen ja seurannan (Middleton, 2001) näkökulmista. Yleisesti edellä mainittujen tutkimusten tulokset osoittivat, että Lean-periaatteet voivat olla hyödyllisiä ohjelmistokehityksessä. Lean-periaatteiden tarkempi arviointi on kuitenkin tärkeää, jotta voidaan ymmärtää kuinka ne vaikuttavat ohjelmistoprosessien tehokkuuteen. Tutkimukset eivät tarkalleen esittäneet yksityiskohtia siitä, mitä Lean-periaatteita ja -työkaluja käytettiin ja kuinka ne implementoitiin. Yksikään tutkimuksista ei keskittynyt arvioimaan tiettyjä menetelmiä tai periaatteita Lean-työkalujen joukosta. (Petersen & Wohlin, 2010a.)

Luvussa esiteltiin myös Leanin mukaisen ohjelmistojen ylläpidon arviointia. Ylläpito muodostaa jopa 50 % - 80 % koko ohjelmistokehityksen vaatimasta työmäärästä, ja näin ollen parannukset ohjelmistojen ylläpidon prosesseissa voivat kasvattaa taloudellisuutta ja tehokkuutta. Toistaiseksi ei ole kuitenkaan saatavilla ratkaisua ohjelmistojen ylläpidon arviointiin Lean-ohjelmistokehityksen näkökulmasta. (Petersen, 2010, s.251.)

6 POHDINTA

Tässä luvussa pohditaan tutkielmassa käsiteltyjä aiheita, kuten Leania teollisuudessa ja sen juuria, Lean-ohjelmistotuotantoa sekä Leanin mukaisen ohjelmistotuotannon arviointia, sekä esitellään aiheita kriittisesti käsitteleviä lähteitä.

6.1 Lean teollisuudessa

Dybån ja Sharpin (2012) mukaan Womackin ym. (1990) kirjassaan esittämiä väitteitä siitä, että Lean on ylivertainen tuotantomenetelmä, joka tulisi ottaa käyttöön kaikilla tuotannon aloilla niin pian kuin mahdollista, voidaan pitää melko rohkeina väitteinä ja niitä tulisi tutkia tarkemmin ja kriittisesti todisteiden valossa.

Womack ym. (1990) pohjustivat kirjassaan esittämänsä laskelmat tehokkuudesta MIT:n tekemään International Motor Vehicle Program (IMVP) -tutkimukseen, joka tutki autoteollisuuden tehtaiden tuottavuutta eri puolilla maailmaa 1980-luvun lopulla. IMVP-tutkimuksessa käsiteltiin panosten ja tulosten (I/O) suhdetta työtunteihin ja rakennettujen autojen määrään. Womackin ym. (1990) teos nosti IMVP-tutkimustulokset maailman tietoisuuteen raportoidulla muun muassa, että parhaiden japanilaisten ja heikoimpien amerikkalaisten välinen suhde oli 2:1 ja että Lean-tuotanto vähensi selkeästi työtuntien määrää autojen rakentamisessa riippumatta tehtaan automaatiotasosta. (Dybå & Sharp, 2012.)

IMVP-tutkimus keskittyi mittaamaan ainoastaan autotuotannon myöhäisempiä vaiheita, kuten hitsaamista, maalausta, moottorien kokoamista, varustelua ja lopullista kokoamista. Womack ym. (1990) perustelivat näiden vaiheiden valintaa korostamalla, että IMVP-tutkimuksen tarkoitus oli verrata kokoamistehtaita ja että kokoamistehtaat ovat helposti verrattavissa toisiinsa. Coffeyn (2006) mukaan tämä on hämmästyttävää, sillä näin tekemällä saadaan vähen-

nettyä kaikki vaihtelu tehtaan asettelussa ja organisaation merkitys tuottavuuden lähteenä. (Dybå & Sharp, 2012.)

Dybån ja Sharpin (2012) mukaan IMVP-tutkimuksen tuottavuusmittauksia on kritisoitu muun muassa korjatuista tuloksista, kumulatiivisuudesta ja kompleksisuudesta muuan muassa suoran ja epäsuoran työntekijämäärän, standardityöaikojen sekä autojen keskiarvottamisen suhteen. Dybå ja Sharp (2012) korostavat kuitenkin kahden saman kohdan merkitystä kuin Coffey (2006). Näitä ovat tapa, jolla työtunteja mitattiin, sekä tehtaan arvioitu automaatiotasoa.

IMVP-tutkimus käytti tulostensa perusteena työtuntien määrän mittamista suhteessa yhteen rakennettuun autoon. Ensimmäinen Coffeyn (2006) esille tuoma tärkeä huomio on siinä, miten IMVP-tutkimus ei sallinut standardityövuoron mittaa ylittävää työmäärää. Tutkimus sivutti kaiken ylityön ja sisälsi vain työntekijöiden kokonaismäärän. Näin ollen arvioidut työtunnit yhden vuoron perusteella jaettiin tuotettujen autojen määrällä. (Dybå & Sharp, 2012.)

Oletetaan, että tehtaalla on 1500 työntekijää tekemässä yhtä standardoitua 8 tunnin työvuoroa ilman ylityölupaa ja päivän tuotettujen autojen määräksi saadaan yhteensä 400 autoa. Tällöin tuottavuus voidaan laskea kaavalla $(8 \times 1500) / 400 = 30$ tuntia yhtä tuotettua autoa kohti. Mikäli oletetaan, että tehtaalla olisi yhdessä vuorossa vain 750 työntekijää ja kaikki vuoron työntekijät tekisivät kaksinkertaisia vuoroja sekä tuotanto olisi sama 400 autoa, laskettaisiin tuotanto samalla kaavalla $(16 \times 750) / 400 = 30$ tuntia yhtä autoa kohti. Yhteen tuotettuun autoon tarvittu tuntimäärä olisi näin ollen täsmälleen sama, mutta mikäli tutkimuksessa ei sallita ylityötä lainkaan ja tarkastellaan tuloksia standardityövuorojen perusteella ja oletetaan kaikkien tekevän standardoitua 8 tunnin vuoroa, tämä puolittaa tuotantoon tarvittavan ajan $(8 \times 750) / 400 = 15$ tuntia yhtä tuotettua autoa kohti. (Dybå & Sharp, 2012.)

Koko autoteollisuusala on pitkään ollut tunnettu sekä monimutkaisista vuorokaavoistaan että ylityömäärästään. Nämä tietäen Womackin ym. (1990) tulkinnat IMVP-tutkimukseen perustuen herättävät vakavia epäilyksiä, erityisesti kun saatavilla olevan tiedon perusteella keskimääräinen työviikko Japanissa on 10 tuntia pidempi kuin Euroopassa. IMVP-tutkimuksen aikaan 1980-luvun lopulla japanilaiset autonvalmistajat olivat erityisen tunnettuja massiivisista ylityömäärästään. (Dybå & Sharp, 2012.)

Toinen Coffeyn (2006) esille nostama ongelma liittyy yhden auton valmistamiseen tarvittavaan tuntimäärän ja tehtaan arvioidun automaatiotason välisen suhteen tulkintaan. IMVP-tutkimuksen datan perusteella japanilaiset tehtaot olivat pitkälle automatisoituja. Jopa vähiten automatisoidut japanilaiset tehtaot saivat korkeammat pisteet automatisoinnin määrässä verrattuna useimpiin muihin autotehtaisiin muualla maailmassa. (Dybå & Sharp, 2012.)

Automaation merkitys japanilaisessa autoteollisuudessa oli tärkeässä roolissa, ja se vähensi selkeästi japanilaisissa tehtaissa IMVP-tutkimuksessa seurattua auton tuottamiseen tarvittua työtuntimäärää. Womack ym. (1990) havaitsivat, että tehtaan automaatiotasojen variaatio aiheutti noin yhden kolmasosan vaihtelun auton valmistamiseen tarvittavassa työtuntimäärässä. Kuitenkin au-

tomaation merkitystä yhtenä tärkeänä tekijänä vähäteltiin. (Dybå & Sharp, 2012.)

IMVP-tutkimuksen mukaan korkeasti automatisoidut autotehtaat Euroopassa vaikuttivat suoriutuvan poikkeavalla tavalla samantasoisesti automatisoituihin japanilaisiin ja muun maalaisiin tehtaisiin verrattuna. Coffey (2006) käytti tätä esimerkkiä osoittaakseen, miten erilaisia päätelmiä voitiin tehdä samasta tutkimusaineistosta. Mikäli Eurooppaa koskeva tutkimusaineisto olisi otettu erilleen, niin todiste ”Lean-vallankumouksesta” olisi ollut huomattavasti hankalammin havaittavissa. (Dybå & Sharp, 2012.)

Coffeyn (2006) uusi analyysi IMVP-tutkimusaineistosta osoitti, että automaation vaikutus variaatioon oli kolme neljäsosaa. Monien standardien mukaan tätä voitaisiin pitää huomattavan merkittävänä vaikutuksena, ja siksi Coffey (2006) päätteli, että todiste organisaationaalista ylivoimaisuudesta (tai Lean-tuotannosta) korkean tason automaatiota vasten olisi ollut vaikeaa havaita. (Dybå & Sharp, 2012.)

IMVP-tutkimustietoon pohjautuen Coffey (2006) päätteli, että uusittu tutkimus esittää aiemman tiedon uudessa valossa ja eroavaisuudet japanilaisen ja muiden maiden välissä autoteollisuudessa selittyvät paremminkin vääristyneillä tuottavuusmittauksilla ja automaatiotason määrällä (Dybå & Sharp, 2012).

Dybå ja Sharp (2012) sekä Coffey (2006) ovat kuitenkin sitä mieltä, että IMVP-tutkimus voidaan ottaa vakavissaan ja on sinällään validi, mutta joiltain osin sen tulkintoja voidaan pitää kyseenalaisena poiketen yleisessä tiedoissa olevista tulkinnoista.

Dybån ja Sharpin (2012) esittämien tietojen valossa Leanin historia ja perusta tuntuvat hieman kevein perustein rakennetulta. Kieltämättä IMVP-tutkimuksessa olisi ollut selkeästi korjattavaa mittauksien tarkkuuden suhteen. Mutta olen Dybån ja Sharpin (2012) kanssa samaa mieltä, etteivät nämä puutteet sinällään tee itse tutkimuksesta merkityksetöntä. IMVP-tutkimuksen jälkeen Womackin ym. (1990) tuotua Lean-tuotannon koko maailman tietoisuuteen on se yleistynyt niin monelle alalle ja organisaatioihin hyvillä tuloksilla, ettei ole epäselvää, etteikö Leanin avulla voitaisi saada selkeää parannusta organisaation toimintaan.

6.2 Lean-ohjelmistotuotanto

Lean ja Lean-ohjelmistotuotanto ovat käsitteinä varsin moniulotteisia ja kohdittuun uusia länsimaisessa teollisuustuotannossa ja ohjelmistotuotannossa. Lean-periaatteiden mukainen toiminta on jokaisessa organisaatiossa omanlaisensa. Leanin implementoinnin tasoa voidaan lähestyä länsimaisemman lattia-tason prossienhallinnan kautta aina koko organisaation ja sen sidosryhmät kattavaan japanilaiseen filosofiaan saakka. Siten voidaan jo lähtökohtaisesti olettaa, että käyttökokemukset ovat vaihtelevia.

Jonsson ym. (2013) määrittelevät Kettusen (2010) mukaisesti, että Leania on käytetty ohjelmistotuotannossa 2000-luvun alusta saakka, mutta siitä huoli-

matta yksiselitteinen vastaus siihen, miten Lean-ohjelmistotuotanto määritellään, on vailla tarkkaa vastausta. Olen asiasta samaa mieltä Jonssonin ym. (2012) kanssa, ja tämän ilmiön huomaa hyvin tutkimuksen lähdemateriaaleissa. Tutkimuksen lähdemateriaalien näkökulmat Lean-ohjelmistotuotantoa koskien painottavat eri näkökulmia Leanista. Lähdemateriaalissa on nähtävissä samantyyppinen kahtiajako eri koulukuntiin kuin teollisuuden puolella. Osa Lean-ohjelmistotuotantoa koskevista lähteistä edustaa selvästi filosofisempaa japanilaista lähestymistä Leaniin, kun taas osa on selkeästi suoraviivaisempaa länsimaisempaa näkemystä edustavia.

Vilkin (Vilkki & Erdogmus, 2012) mukaan Lean-ajattelu tuo ohjelmistotuotantoon merkittäviä parannuksia, vaikka se ei suoraan määrittele tarkkaa keinoa käytänteisiin tai siihen, miten kirjoittaa ohjelmiston koodia. Merkittävimpänä asiana on Lean-ajattelu eivätkä Lean-käytänteet. Vilkin (Vilkki & Erdogmus, 2012) mukaan käytänteet ovat aina kontekstisidonnaisia ja toimivat paremmin toistoa, vähäistä vaihtelua ja tasakoosteista virtausta sisältävässä teollisuustuotannossa kuin yksittäisiä tehtäviä, suurta vaihtelua ja sekakoosteista virtausta sisältävässä tuotekehityksessä.

Erdogmuksen (Vilkki & Erdogmus, 2012) mukaan Leanin mukanaan tuoma arvo on nimenomaan Leanin konkreettisissa käytänteissä ja työkaluissa, eikä niinkään Leanin taustalla olevassa filosofiassa, kuten Vilkki väittää. Erdogmuksen (Vilkki & Erdogmus, 2012) mukaan aiemmissa johtamisen muotiaalloissa kokonaiskuvat tulevat, menevät ja muuttavat väriään aina sen hetkisten poliittisten prioriteettien mukana, kun taas käytänteet ja työkalut juurtuvat organisaatioon ja sen kulttuuriin syväälle ja näin ollen kestävät pidempään.

Vaikka Vilkin ja Erdogmuksen (2012) näkemykset Leanin mukanaan tuomien hyötyjen taustasyistä eroavatkin. Molemmat ovat samaa mieltä siitä, että Lean ei ole missään nimessä turha ja sen avulla on saatu selkeitä parannuksia toimintaan. Vilkki ja Erdogmus (2012) ovat samaa mieltä myös siitä, että Lean voi olla oman aikansa ohimenevä muotiaalto, joka tulee syrjäytymään jonkin toisen ohjelmistoprosessin alta tulevaisuudessa. Saatujen tulosten perusteella Lean sisältää kuitenkin paljon potentiaalia ollakseen muutakin kuin nopea ohimenevä trendi. Lean saattaa pysyä kokonaan tai osittain mukana ohjelmistotuotannossa mahdollisesti pitkäänkin joko filosofiana tai tiettyjen käytänteiden ja työkalujen osalta. (Vilkki & Erdogmus, 2012).

Mielestäni on hyvin mielenkiintoista nähdä, kuinka Lean-ajattelulla on saatu aikaan positiivisia tuloksia myös ohjelmistotuotannossa riippumatta siitä, onko käytössä japanilainen, filosofisempi näkemys vai länsimainen, prosessikeskeisempi näkemys. Vaikka Dybå ja Sharp (2012) ja Coffey (2006) herättävätkin kysymyksiä liittyen koko Lean-tuotannon ylivertaisuuteen ja sen toteamiseen käytettyihin mittareihin liittyen, on mielestäni ehkä jopa hieman yllättävää, miten hyviä kokemuksia Lean-ohjelmistotuotannosta on saatu selkeistä koulukuntien näkemyseleroista huolimatta.

Kaikki Lean-implementoinnit eivät ole kuitenkaan päättyneet positiivisiin tuloksiin (Jonsson ym., 2013; Karvonen ym., 2012; Kuusela ym., 2011; Kuusela & Koivuluoma, 2011). Leanin käyttöönotto vaatii huomattavia investointeja täysin

uuden malliseen toimintaan, jossa vastuuta jaetaan johdolta alaspäin työntekijöille. Tämä tilanne herättää kysymyksiä ja huolia niin johdon kuin työntekijöidenkin puolella. Työntekijöiden ja johdon motivointi, sitouttaminen ja muutostavastarinta Leanin mukaiseen toimintaan ovat käyttöönoton yhteydessä yleisiä haasteita (Bozdogan ym., 2000; Parnell-Klabo, 2006).

Omat haasteensa implementoinnin onnistumiselle luovat varmasti Leanin kaksi koulukuntaan sekä se, mille tasoille organisaatiossa Lean tullaan implementoimaan. Tiettyjä Lean-työkaluja tietyssä prosessissa hyödyntämällä voidaan saada Leanin avulla hyötyjä ja parannusta toimintaan, mutta aiemmista tutkimuksista ei selkeästi löydy mainintaa siitä, millä organisaatiotasolla ja miten laajalla implementoinnilla voitaisiin sanoa ohjelmistotuotannon olevan Leanin mukaista.

Monissa ohjelmistokehitysorganisaatioissa on usein tapana yhdistellä alalla parhaiksi havaittuja käytänteitä, ja tämä johtaa helposti organisaation sisällä tilanteeseen, jossa eri osastojen, tiimien ja jopa yksittäisten prosessien sisällä voidaan käyttää useita erilaisia käytänteitä ja prosessikehitysmenetelmiä. Tässä tilanteessa Jonssonin ym. (2013) peräänkuuluttama tarkka määritelmä sille, mitä Lean-ohjelmistotuotanto tarkalleen on, olisi jälleen tarpeellinen. Ilman tarkkaa määritelmää ei voida sanoa, missä vaiheessa organisaation nykymallinen ohjelmistotuotanto muuttuu Leanin mukaiseksi tai sitä voidaan pitää Lean-ohjelmistotuotantona. Mikäli organisaatiossa, tiimissä tai yksittäisessä prosessissa käytetään Lean-hybridilähestymistapaa, eli käytössä on sekä Leanin mukaisia käytänteitä ja vaikka osia Scrumista, XP:stä tai muista kehitysmenetelmistä, on hankalaa sanoa tarkalleen, onko organisaation ohjelmistotuotanto Leania, Scrumia vai Scrumbania, ja minkä periaatteen tai arviointiviitekehityksen mukaan sitä tulisi tällöin mitata.

Lean-ohjelmistotuotanto on ollut viimeisten vuosien aikana kuuma ja trendikäs aihe ja tämä voi epäilemättä johtaa siihen, että sen suuremmin aiheeseen tutustumatta organisaation käyttöön voidaan implementoida jokin tai jotkin yksittäiset Lean-käytänteet lähinnä markkinointitarkoituksessa. Mikäli tähän liittyy vielä organisaation heikko panostus Lean-koulutukseen ja Leanin mukaiseen toimintaan tai implementoituihin käytänteisiin sitoutumiseen ei mielestäni voida odottaa kovinkaan suurta läpimurtoa organisaation tehokkuudessa tai työntekijöiden kokemuksissa Leanin mukaisesta toiminnasta ylivertaisena ohjelmistotuotantomenetelmänä.

Huomion arvoista mielestäni Lean-ohjelmistotuotannon implementointia suunniteltaessa on myös ottaa huomioon organisaation koko. Kuuselan ym. (2011) mukaan Womackin ja Jonesin (1996) aikaikkuna globaalille Lean-organisaatiolle on jopa viisi vuotta kestävä transformaatioprosessi. Useiden lähteiden (Ohno, 1988; Womack ym., 1990; Middleton, 2001) perusteella tiedetään, että Lean-implementaation alkuvaiheissa organisaation prosesseissa nähdään usein tiheällä tahdilla pysähdyksiä ja toiminnan keskeytymistä virheiden löytyessä. Implementoinnin ja organisaation koon skaala vaikuttavat epäilemättä tähän. On varmasti helpompaa ja nopeampaa omaksua Leanin mukainen toiminta ja sen käytänteet pieneen alle kymmenen hengen, yhden tuotteen start-

up-yritykseen kuin globaalisti toimivan ja maantieteellisesti hajautetun, tuhansien työntekijöiden organisaation kaikille tasoille. Jopa vuosien mittainen aikaikkuna huomioiden ei suurissa organisaatioissa tulisi huolestua ja hylätä implementointiaietta heti ensimmäisten vastoinkäymisten esiintyessä ja prosessien pysähtyessä.

6.3 Lean-ohjelmistotuotannon arviointi

Jonsson ym. (2013) toteavat Kettusen (2010) mukaisesti, että Leania on käytetty jo reilun vuosikymmenen ajan ohjelmistotuotannossa, mutta tästä huolimatta ei vielä ole yksiselitteistä vastausta siihen, miten Lean-ohjelmistotuotanto tulisi määritellä. Tämä on selkeästi nähtävissä jo tutkimalla Lean-ohjelmistotuotantoa käsittelevää peruskirjallisuutta. Mutta vielä selkeämmin tämä tilanne näkyy mielestäni Lean-ohjelmistotuotannon arviointia käsittelevissä teoksissa. On epäilemättä hankalaa arvioida jotain, jolla ei ole tarkkaa määritelmää. Lähdemateriaalissa Lean-ohjelmistotuotannon arviointia käsitteleviä teoksia on useampia (Middleton, 2001; Middleton, 2005; Perera & Fernando, 2007; Parnell-Klabo, 2006; Mujtaba ym., 2010; Petersen & Wohlin, 2010b; Karvonen ym., 2012; Maglyas ym., 2012), mutta niiden sisältö ja arviointimenetelmät ovat varsin vaihtelevia. Osa keskittyy vain yhden tai parin Lean-käytännön seuraamiseen ja arviointiin (Middleton, 2001; Middleton, 2005; Perera & Fernando, 2007; Parnell-Klabo, 2006), mutta laajemman skaalan kattavia, Lean-ohjelmistotuotantoon spesifisti rakennettuja ohjelmistoprosessien arviointimenetelmiä ei SPI-LEAM:in (Petersen & Wohlin, 2010b) lisäksi juuri löytynyt.

Lean-ohjelmistotuotannon selkeän määritelmän puute näkyi hyvin myös yhtä tai muutamaa yksittäistä Lean-periaatetta tai -käytännettä painottavissa mittareissa (Poppendieck, 2002; Maglyas ym., 2012; Khurum ym., 2014; Perera & Fernando, 2007; Middleton, 2001). Mittaaminen ankkuroidaan yleisimmin Lean-ohjelmistotuotannon ja kehitysprosessien osalta jätteen minimoinnin (Poppendieck, 2002), virtauksen nopeuden (Poppendieck, 2002; Maglyas ym., 2012; Middleton & Joyce, 2010), arvon ja arvonmuodostuksen (Krurum ym., 2012; Maglyas ym., 2012; Poppendieck & Poppendieck, 2003; Mujtaba ym., 2010), koodirivien määrän (Perera & Fernando, 2007) ja virheiden tunnistuksen ja seurannan (Middleton, 2001) näkökulmista. Omat haasteensa tässä tuo se, ettei kaikissa lähteissä selkeästi mainita, mitä organisaatioissa oli tarkalleen arvioitu ja millä menetelmillä. Toinen haaste arviointimenetelmien tunnistamisessa on Lean-käytänteiden ja -periaatteiden väliset läheiset suhteet toisiinsa. Vaikka tutkimuksissa keskitytään seuraamaan ja arvioimaan esimerkiksi jätteen minimointia, on sillä lähes suora vaikutus virtauksen nopeuteen, virheiden tunnistamiseen ja arvon muodostukseen, vaikka lähdemateriaalissa näitä ei olisi nostettukaan erikseen esille tutkittavina ja arvioitavina kohtina.

Leanin mukaisen ohjelmistotuotannon organisaatiotason arviointimalleja löytyy muutamia, joista tarkemmin esiteltiin LESAT for Software (Karvonen ym., 2012) sekä Ericssonin Lean Amplifier (Karvonen ym., 2012). Arviointimal-

lien sisällöt ja painotukset ovat huomattavan erilaisia, vaikka ne ovatkin samaan kontekstiin suunniteltuja. LESAT for software -malli sisältää yhteensä 54 arviointikohtaa, kun taas Lean Amplifier -malli sisältää yhteensä 167 arviointikohtaa. Nämä arviointikohdat painottuvat eri osioihin, joista esimerkkinä Lean Amplifier -mallin vähäinen johtaja-arviointiosion painoarvo muihin osioihin verrattuna. LESAT for Software -mallissa suurin osa arviointikohdista painotuu taas juurikin johtajuus- ja transformaatio-osion alle.

Toinen mielenkiintoinen huomio malleihin liittyen on Karvosen ym. (2012) toteamus siitä, että vaikka Lean Amplifier painottaa voimakkaasti tiimin ja organisaation arviointikohtia Leanin ja ketterien menetelmien käyttöönotossa, tulokset kuitenkin osoittivat, että LESAT for Software omaa paremmin koko organisaation kattavan valikoiman pienemmästä arviointikohtien määrästä huolimatta. Malleja vertailtaessa tulee kuitenkin huomata, että LESAT for Software on yleinen arviointimalli Lean-ohjelmistotuotannon tarpeisiin, kun taas Lean Amplifier on nimenomaan Ericssonin omiin tarpeisiin kehitetty arviointimalli. Näin ollen siinä mahdollisesti esiintyvät puutteet ja heikkoudet voivat olla Ericssonilla paikattavissa Lean Amplifier -mallin ulkopuolisilla menetelmillä, jotka eivät ole yleisessä tiedossa.

Eräs huomionarvoinen seikka koskien erityisesti tutkimuksessa esitettyihin muihin Lean-ohjelmistotuotannon arvioinnin menetelmiin liittyen on se, että useassa tutkitussa organisaatiossa nykytilan arviointi ennen Leanin mukaisen toiminnan implementointia ei välttämättä ollut kovinkaan vahvasti käytössä. Hyvänä esimerkkinä tästä on Middletonin (2001) tutkima organisaatio, jossa ainoat ohjelmistotuotantoa koskevat seuranta- ja arviointimenetelmät olivat viikoittainen talous- ja johdonraportointi johtoryhmälle sekä työntekijöiden tuntiseuranta, joka ei sinällään kerro juurikaan muusta kuin paikalla olost. Uskallan väittää, että mikäli organisaation ja sen toimintojen mittaaminen on tällä tasolla, ei toiminnan parantamisen kannalta sinällään ole niinkään suurta merkitystä sillä, implementoidaanko käyttöön Lean vai jokin muu prosessikehitysmenetelmä. Toki Lean tarjoaa useilla aloilla ja organisaatioissa hyväksi havaitut menetelmät ja työkalut prosessien ja työnseurantaan, mutta mitä todennäköisimmin mikä tahansa parannustoimi yrityksen prosessien ja ongelmakohdient seurannassa olisi parantanut merkittävästi organisaation toimintaa sen lähtötilanteesta.

Vaikka Leanin implementointia pohtivassa ohjelmistoalan organisaatiossa Lean-ohjelmistotuotantoon ja sen arviointiin liittyvät epäjohtonmukaisuudet ja variaatiot voivat aiheuttaa huolia ja kysymyksiä, voidaan tutkimuksen lähdemateriaalien perusteella todeta, että mitä todennäköisimminkin Lean-ohjelmistotuotannon implementoinnilla ja arvioimisella voidaan saada aikaan selkeää parannusta toiminnassa ja tehokkuudessa. Parannuksen aikaansaamiseksi ei aiempia tutkimuksia vertaamalla tunnu olevan niinkään suurta merkitystä sillä, implementoidaanko Lean enemmänkin toimintaa ohjaavana filosofiana vai prosessitason käytänteinä. Samoin arviointia ajatellen tutkimuksen lähdemateriaalien perusteella ei ole selkeästi havaittavissa, että yhden tietyn arviointimenetelmän tai arviointikohdan avulla saataisiin aikaan merkittävästi

parempia tuloksia kuin toisella. Osa tästä selittyy varmasti Lean-periaatteiden, -käytänteiden ja -työkalujen läheisillä sidoksilla ja rinnakkaisvaikutuksilla toisiinsa.

7 YHTEENVETO

Tämän tutkielman tarkoituksena oli esittää kirjallisuuteen perustuva tiivis ja yhtenäinen kuvaus siitä, mitä Lean-ajattelulla tarkoitetaan teollisuudessa ja ohjelmistotuotannossa, miten transformointi Lean-periaatteiden mukaiseen toimintaan voidaan toteuttaa ohjelmistotuotannossa sekä siitä, kuinka Leanin mukaista toimintaa ja Lean-ohjelmistotuotantoa voidaan arvioida.

Tutkielman tutkimusongelma on ”Miten Lean-ajattelua voidaan soveltaa ohjelmistotuotannossa?”. Tutkimusongelma on jaettu neljään tutkimuskysymykseen. Ensimmäisen tutkimuskysymyksen tarkoitus oli selvittää, *mistä Lean on lähtöisin ja mikä sen sisältö on?* Lean on alun perin Toyotan autoteollisuuden 1940-luvulla kehittämä teollisuuden tuotantomenetelmä. Kirjallisuudessa Leanin perusteoksina pidetään Toyotan alkuperäistä dokumentaatiota, jonka pohjalta Ohno (1988) on julkaissut kirjan. Maailmanlaajuisesti Lean tuli kuitenkin tunnetuksi vasta 1990-luvulla Womackin ym. (1990) sekä Womackin ja Jonesin (1996) teoksien myötä. Lean kehitettiin nimenomaan tehostamaan tuotantoa, ja sen perimmäisenä ajatuksena onkin tuottaa mahdollisimman vähällä mahdollisimman paljon. Lean perustuu viiteen pääperiaatteeseen, joita ovat arvo, arvovirta, virtaus, imu ja täydellisyys. Helpoin ja eniten Leanin yhteydessä käytetty keino arvon ja tuottavuuden lisäämiseksi on hukkan eliminointi.

Toisen tutkimuskysymyksen tarkoitus oli selvittää, *mitä tarkoitetaan Lean-ajattelulla ja -periaatteilla ohjelmistotuotannossa.* Tutkielmassa esiteltiin Lean-ohjelmistotuotannon perusteoksina pidettävien Poppendieckin ja Poppendieckin (2003, 2006) mukaan teollisuuden ja ohjelmistotuotannon prosessien eroavaisuudet. Koska nämä eroavat toisistaan huomattavasti, niin Poppendieckit (2003, 2006) ovat esittäneet Lean-periaatteet ohjelmistotuotannon näkökulmasta. Lean-ohjelmistotuotannon seitsemän perusperiaatetta ovat: poista hukka, rakenna laatu ohjelman sisälle jo alussa, luo tietoa, lykkää sitoutumista, toimita nopeasti, kunnioita ihmisiä ja optimoi kokonaisuus. Ohjelmistotuotannossa on tunnistettu seitsemän erityyppistä hukkaa, joita ovat: osittain tehty työ, ylimääräiset ominaisuudet, uudelleen oppiminen tai tiedon hukka, tuotteen luovutukset, tehtävien vaihdot, viivästykset ja viat. Verrattaessa näitä Poppendieckien (2003, 2006) näkemyksiä Womackin ja Jonesin (2003) ja Likerin (2004) näkemyk-

siin on havaittavissa selkeitä yhtäläisyyksiä. Poppendieckien näkemys on ikään kuin ohjelmistotuotannon näkökulmasta tehty yhdistelmä näitä molempia.

Kolmannen tutkimuskysymyksen tarkoituksena oli selvittää, *millä tavalla voidaan tukea Leanin mukaiseen toimintaan siirtymistä*. Tutkimuksessa todettiin, että Leanin mukaiseen toimintaan siirtymistä voidaan tukea käyttämällä apuna transformaatiomalleja tai -ohjeistoja (Bozdogan ym., 2000; Liker, 2004). Donaldsonin (1996) mukaan ei ole olemassa yhtä mallia, joka olisi sovellettavissa tehokkaasti kaikkiin organisaatioihin, vaan malli tulee valita tilannekohtaisesti. Tutkielmassa esitettiin Bozdoganin ym. (2000) malli, joka on kolmesta syklistä rakentuva transformaatiomalli. Bozdoganin ym. (2000) malli sisältää organisaatiotasolla yleisluotoisen kuvauksen vaiheista, jotka panevat alulle, ylläpitävät ja jatkuvasti parantavat organisaation muuttamista Lean-periaatteiden ja -käytänteiden mukaiseksi. Tutkielmassa esitettiin myös Likerin (2004) transformaatiomalli, joka eroaa muodoltaan huomattavasti Bozdoganin ym. (2000) mallista. Likerin (2004) malli ei tarjoa suoraa ohjeistusta siitä, mitä tulee tehdä ja miten, vaan on enemmänkin omaa oppimista, omaa ajattelua, Lean-filosofiaa ja oman tavan kehittämistä Toyotan ideoiden ja peruseriaatteiden päälle korostava ohjeisto. Ohjelmistotuotannon muuttamiseksi Leanin mukaiseksi on myös tehty omia transformointimalleja ja -ohjeistoa (Kuusela ym., 2001; Bhasin & Burcher, 2006; Hines, 2004). Näissä on nähtävissä saman tyyppinen kahtiajako kuin teollisuuslähtöisissä malleissa. Osa malleista on hyvin suoraviivaisia transformaatiomalleja (Kuusela & Koivuluoma, 2011) ja osa enemmän omaa ajattelua ja filosofisempaa lähestymistä noudattavia (Bhasin & Burcher, 2006; Middleton & Sutton, 2005).

Neljännän tutkimuskysymyksen tarkoitus oli selvittää, *miten Leanin mukaista ohjelmistotuotantoa voidaan arvioida*. Tutkimuksessa todettiin Karvosen ym. (2012) mukaan, että Lean voi olla organisaatiossa käytössä useilla eri tasoilla aina yksittäisestä prosessista tai käytänteestä koko organisaation tasolle. Riippuen siitä kuinka laaja Lean-transformaatio organisaatiossa on, sitä suuremiksi organisaation muutokseen liittyvät riskit kasvavat (Karvonen, ym., 2012). Organisaatio voi käyttää yksittäisiä Lean-käytänteitä tai toimia tietyllä organisaatiotasolla Leanin mukaisesti. Mutta ei voida kuitenkaan puhua Lean-organisaatiosta, ellei organisaatio käytä Leania kaikessa, mitä se tekee (Karvonen ym., 2012; Womack ym., 1990). Syvällisen nykytilan sekä halutun tulevaisuuden tilan arvioinnin on todettu oleelliseksi transformaation ohjaamisessa oikeaan suuntaan (Karvonen ym., 2012; Kuusela & Koivuluoma, 2011).

Leanin mukaisen toiminnan arvioinnin avuksi on esitelty useita kyselytutkimuksia, ohjeistoa ja arviointimalleja. Eräs tunnetuimpia Lean-arviointimalleja on MIT:n kehittämä LESAT (Lean Enterprise Self-Assessment Tool). LESAT jakautuu kolmeen pääosiin, joita ovat Lean-transformaatio- / johtajuusosio, elinkaariprosessiosio sekä infrastruktuurin mahdollistamisosio (Karvonen ym., 2012). LESAT ei ole suoraan sovellettavissa ohjelmistotuotantoon, mutta Karvonen ym. (2012) ovat tehneet siitä ohjelmistotuotantoon suunnatun muunnelman, jota kutsutaan LESAT for Software. Karvosen ym. (2012) mukaan LESAT for Software -malli voi olla avuksi ja ohjata organisaation trans-

formaatiota kohti Leanin mukaista ohjelmistotuotantoa sekä vaikuttaa aina organisaatiotasolle asti. LESAT for Software sisältää kuitenkin vielä puutteita ja jatkokehitystarpeita.

Lean-ohjelmistotuotantoa voidaan arvioida myös ohjelmistokehityksen näkökulmasta. Aiempaa tutkimusta Leanin mukaisesta ohjelmistokehityksestä ja sen käytön ja tulosten arvioinnista yrityksissä ovat tehneet muun muassa Middleton (2001), Middleton ym. (2005), Perera ja Fernando (2007) sekä Parnell-Klabo (2006). Ohjelmistokehityksen parantaminen tähtää ohjelmistoprosessien tehostamiseen ja tuotteen laadun kasvattamiseen jatkuvalla arvioinnilla sekä prosessin säätämällä. Tätä varten on luotu useita prosessikehitysmalleja, joista tunnetuimpia ovat Capability Maturity Model Integration (CMMI) (CMMI-Product-Team, 2006) ja Quality Improvement Paradigm (QIP) (Basili, 1985; Basili & Green, 1994). CMMI- ja QIP-malleja ei ole suunniteltu Lean-ohjelmistotuotantoa ajatellen, ja näitä puutteita paikkaamaan Petersen ja Wohlin (2010b) ovat kehittäneet oman ohjelmiston laadun parantamismallin, jota kutsutaan nimellä SPI-LEAM (Software Process Improvement through the Lean Measurement). Basilin (1985) QIP-malliin perustuva malli yhdistää ohjelmiston laadun parantamisen ajatusmallin Lean-ohjelmistokehityksen periaatteisiin. SPI-LEAM kuvaa tavan, jolla implementoida Lean-periaatteet mittaamisen kautta siten, että saadaan aikaan parannusta ohjelmistoprosesseissa. (Petersen & Wohlin, 2010b.)

Erillisten mallien lisäksi Lean-ohjelmistokehitystä voidaan arvioida myös ankkuroimalla näkökulma johonkin yksittäiseen tai useampaan Lean-periaatteeseen. Periaatteiden kautta mittaamisessa ja arvioinnissa kiinnitetään huomiota ohjelmistokehitykseen ja ohjelmistoprosesseihin pääasiassa jätteen minimoinnin (Poppendieck, 2002), virtauksen nopeuden (Poppendieck, 2002; Maglyas ym., 2012; Middleton & Joyce, 2010), arvon sekä arvonmuodostuksen (Krurum ym., 2012; Maglyas ym., 2012; Poppendieck & Poppendieck, 2003; Mujtaba ym., 2010), koodirivien määrän (Perera & Fernando, 2007) ja virheiden tunnistuksen ja seurannan (Middleton, 2001) näkökulmista. Vaikka näkökulma tietyn tai tiettyjen peruseriaatteiden perusteella mittaamiseen vaihtelee eri tutkimuksissa, kaikkien avulla on pystytty toteamaan selkeää parannusta Leanin mukaisen ohjelmistotuotannon ja sen prosessien toiminnassa.

Tämän tutkimuksen tuloksia voidaan hyödyntää miettiessä ohjelmistotalan yrityksissä, millä tavalla Lean-ajattelua ja -periaatteita voitaisiin hyödyntää, miten niitä voitaisiin ottaa käyttöön ja miten Leanin mukaista toimintaa voidaan arvioida. Tutkimuksessa käytetty lähdemateriaali on suurelta osin alan perusteoksia, joiden pohjalta myöhempi tutkimus ja kirjallisuus on rakennettu.

Lähdemateriaalissa on vain vähän tutkittua tietoa liittyen käyttökokemuksiin Lean-transformaatiosta ohjelmistotuotannossa. On kuitenkin huomattava, että tämän aiheen tarkempi tarkastelu vaatisi varmasti oman tutkielmansa, jotta voitaisiin saada aiheesta riittävän kattava kuvaus. Leaniin liittyvät näkemykset eroavat myös jonkin verran toisistaan, ja tämä saattaa johtaa aiheeseen perehtyvää helposti harhaan ja johtaa jopa transformoinnin epäonnistumiseen. Suurimpana erona näkemyksissä on ero filosofisemman japaninaisen ja prosessi-

keskeisemmän länsimaisen näkemyksen välillä. Tutkielmassa on pyritty huomioimaan molemmat näkökulmat ja niiden merkittävimmät yhtäläisyydet ja eroavaisuudet. Leanin mukaisen toiminnan arvioinnin haasteena on sen implementoinnin taso eri organisaatiotasolla. Osa organisaatioista käyttää vain tiettyjä Lean-käytänteitä tietyissä prosesseissa ja tietyllä organisaatiotasolla, kun taas osaa voidaan pitää puhtaina Lean-organisaatioina, jotka käyttävät Leania kaikessa mitä tekevät. Tämän tasoisten skaalaerojen valossa on todettava, ettei täysin kattavaa aiheen käsittelyä ole ollut mahdollista toteuttaa tämän tutkielman puitteissa.

Tutkimuksen tuloksia tarkastellessa tulee ottaa huomioon, että iso osa käytetystä lähdemateriaalista perustuu vain muutamaaan alan perusteokseen. Ohjelmistotuotannon näkökulmasta Lean ja sitä koskevat tutkimukset ovat varsin nuoria, eikä yleensä vuosia kestävien Lean-transformaatioiden tuloksista ja näihin liittyvistä transformaatiota ohjaavista arvioinneista ole vielä saatavilla riittävästi tutkimusta.

Tutkimuksesta rajattiin pois ohjelmistokehityksen hybridi-lähestymistapoja. Näiden osalta olisikin syytä tehdä jatkotutkimusta muun muassa siitä, mitkä Lean-periaatteet ja -käytännöt toimivat parhaiten muiden kehitysmenetelmien kuten Scrumin tai XP:n kanssa ja miten tämän tyyppistä ohjelmistokehitystä voidaan arvioida. Leanin mukaisen ohjelmistokehityksen ja sen arvioinnin skaalautuvuus organisaation koon ja organisaatiotasojen mukaan tarjoaisi myös mahdollisuuden syvempään tutkimukseen, joka ei tämän tutkielman puitteissa ole ollut mahdollista.

LÄHTEET

- Anderson, D. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, Washington: Blue Hole Press.
- Anderson, D., Concas, G., Lunesu, M. & Marchesi, M. (2011). Studying Lean-Kanban Approach Using Software Process Simulation. *Teoksessa Agile Process in Software Engineering and Extreme Programming* (s.12-26). Springer Berlin Heidelberg.
- Basili, V.R. (1985). *Quantitative evaluation of software methodology*. Technical Report. TR- 1519, University of Maryland.
- Basili, V.R. & Green, S. (1994). Software process evolution at the SEL. *IEEE Software*, 11(4), 58–66.
- Beck, K. (1999). *Extreme programming explained: embrace change*. Reading, Addison-Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for agile software development. Haettu 10.9.2014 osoitteesta <http://www.agilemanifesto.org/>
- Bhasin, S. & Burcher, P. (2006). Lean viewed as a philosophy. *Journal of Manufacturing Technology Management*, 17(1), 56–72
- Bozdogan K., Milauskas R., Mize J., Nightingale D., Taneja A. & Tonaszuck D. (2000). *Transitioning to a Lean Enterprise: A Guide for Leaders, Volume 1 Executive Overview*. Massachusetts: Massachusetts Institute of Technology.
- CMMI-Product-Team. (2006). *Capability Maturity Model Integration for Development, version 1.2*. Technical Report. CMU/SEI-2006-TR-008.
- Coffey, D. (2006). *The Myth of Japanese Efficiency: The World Car Industry in a Globalizing Age*. Edward Elgar Publishing.
- Cusumano, M. (1994). The Limits of Lean. *Sloan Management Review*, 35, Summer, 27-32.
- Doolen, T. & Hacker, M. (2005). A review of Lean assessment in organizations: an exploratory study of Lean practices by electronics manufacturers, *Journal of Manufacturing Systems*, 24(1), 55-67.
- Dybå, T. & Sharp, H. (2012). What's the Evidence for Lean? *IEEE Software*, 29(5), 19-21.
- Fujitsu. (2010). Lean creates a solid platform for growth. *Fujitsu Services Limited*. FUJITSU. Haettu 15.04.2013 osoitteesta <http://www.fujitsu.com/downloads/SVC/fs/casestudies/fujitsu-lean.pdf>
- Fullerton, R., McWatters, C. & Fawson, C. (2003). An examination of the relationships between JIT and financial performance. *Journal of Operations Management*, 21(4), 383-404.

- Hines, P., Found, P., Griffiths, G. & Harrison, R. (2011). *Staying Lean: thriving, not just surviving*. CRC Press.
- Hibbs, C., Jewett, S. & Sullivan, M. (2009). *The Art of Lean Software Development*. USA: O'Reilly Media Inc.
- Hines, P., Holweg, M. & Rich, N. (2004). Learning to evolve. A review of contemporary Lean thinking. *International Journal of Operations & Production Management*, 24(10), 994-1011.
- ISO/IEC. (2003). *ISO/IEC 15504 Information Technology - Process Assessment*. ISO/IEC.
- ISO/IEC. (2008). *ISO/IEC 12207-2008 Standard for systems and software engineering - software life cycle processes*. ISO/IEC.
- Jones, G. (2004). *Organizational theory, design and change*. London: Pearson Prentice Hall
- Jonsson, H., Larsson, S. & Punnekkat, S. (2013). Synthesizing a comprehensive framework for lean software development. *Teoksessa Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on (s. 1-8)*. IEEE.
- Jordan, J. A. & Michel, F. J. (2001). *The lean company: making the right choices*. Society of Manufacturing Engineers.
- Karvonen, T., Rodriguez, P., Kuvaja, P., Mikkonen, K. & Oivo, M. (2012). Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain. *Teoksessa 38th Euro Micro Conference on Software Engineering and Advanced Applications, (s. 266-273)*. IEEE.
- Kerzazi, N. & Robillard, P. (2013). Kanbanize the Release Engineering Process. *Teoksessa Proceedings of the 1st International Workshop on Release Engineering (s. 9-12)*. IEEE Press.
- Kettunen, P. (2010). A tentative framework for Lean software enterprise research and development. *Teoksessa Lean Enterprise Software and Systems (s. 60-71)*. Heidelberg: Springer Berlin
- Kettunen, P. (2012). Systematizing Software Development Agility: Towards an Enterprise Capability Improvement Framework. *Journal of Enterprise Transformation*, 2(2), 81-104.
- Khurum, M., Gorschek, T. & Wilson, M. (2012). Software value map - an exhaustive collection of value aspects for the development of software intensive products. *Journal of Software: Evolution and Process*, 25(7), 711-741.
- Khurum, M., Petersen, K. & Gorschek, T. (2014). Extending value stream mapping through waste definition beyond customer perspective. *Journal of Software Evolution and Process*, 26(12), 1074-1105.
- Kniberg, H. & Skarin, M. (2010). *Kanban and Scrum: making most of both*. C4Media Inc.
- Krafcik, J. (1988). *Comparative Analysis of Performance Indicators at World Auto Assembly Plants*. Dissertation Thesis. Sloan School of Management, Massachusetts Institute of Technology.

- Kuusela, R., Huomo, T. & Korkala, M. (2011). *Lean thinking principles for cloud software development*. A Research Summary of VTT Technical Research Center of Finland.
- Kuusela, R. & Koivuluoma, M. (2011). Lean transformation framework for software intensive companies: responding to challenges created by the cloud. Teoksessa *37th Euro Micro Conference on Software Engineering and Advanced Applications (SEAA)* (s. 378-382). IEEE.
- Ladas, C. (2008). *Scrumban: Essays on Kanban Systems for Lean Software Development*. Seattle, Washington: Modus Cooperandi Press.
- Lean Enterprise Implementation Group. (1999). *The 360° Lean Audit*.
- Lean Learning Center. (2003). *The Lean Company Survey*. www.leanlearningcenter.com
- Liker, J. K. (2004). *The Toyota way – Toyotan tapaan*. (M. Niemi, Suom.) Jyväskylä: Readme.fi
- Maglyas A., Nikula U. & Smolander K. (2012). Lean Solutions to Software Product Management Problems. *IEEE Software*, 29(5), 40-46.
- Mehta, M., Anderson, D. & Raffo, D. (2008). Providing Value to Customers in Software Development Through Lean Principles. *Software Process: Improvement and Practice*, 13(1), 101-109.
- Messerschmitt, D. & Szyperski, C. (2003). *Software ecosystem: Understanding an indispensable technology and industry*. MIT Press Books, 1.
- Middleton, P. (2001). Lean software development: two case studies. *Software Quality Journal* 9(4), 241-252.
- Middleton, P. & Joyce, D. (2010). Lean software management: BBC worldwide case study. Teoksessa *Proceedings of Lean Software & Systems Conference* (s. 23-44). Atlanta.
- Middleton, P., Flaxel, A. & Cookson, A. (2005). Lean software management case study: Timberline inc. Teoksessa *Extreme Programming and Agile Processes in Software Engineering* (s. 1-9). Springer Berlin: Heidelberg.
- Middleton, P. & Sutton, J. (2005). *Lean software strategies: Proven techniques for managers and developers*. Kraus Productivity Organization Ltd.
- MIT. (2001). *Lean Enterprise Self-Assessment Tool Version 1.0*. Massachusetts Institute of Technology. Haettu 19.11.2014 sivustolta <http://dspace.mit.edu/handle/1721.1/81903>
- MIT. (2012a). *LAI Enterprise Self-Assessment Tool (LESAT) Version 2.0*. Massachusetts Institute of Technology. Haettu 12.11.2014 sivustolta <https://dspace.mit.edu/handle/1721.1/84688>
- MIT. (2012b). *LAI Enterprise Self-Assessment Tool - LESAT Facilitator's Guide*. Massachusetts Institute of Technology. Haettu 2.10.2014 sivustolta <http://dspace.mit.edu/handle/1721.1/84694>
- Morgan, T. (1998). *Lean manufacturing techniques applied to software development*. Master's Thesis, Master Thesis at Massachusetts Institute of Technology.
- Mujtaba, S., Feldt, R. & Petersen, K. (2010). Waste and Lead time reduction in software product customization process with value stream maps.

- Teoksessa *Proceedings of 21st Australian Software Engineering Conference (ASWEC 2010)*, Auckland, Uusi-Seelanti, 139-148. IEEE.
- Northwest High Performance Enterprise Consortium. (2002). *HPEC Assessment*. <http://www.nwhpec.org>
- Nightingale, D. & Mize, J. (2002). Development of a Lean enterprise transformation maturity model. *Information Knowledge Systems Management*, 1(3), 15-30.
- Nikitina, N., Kajko-Mattsson, M. & Stråle, M. (2012). From Scrum to Scrumban: A Case Study of a Process Transition. Teoksessa *Proceedings of the International Conference on Software and System Process* (s. 140-149). IEEE Press.
- Ohno, T. (1988). *Toyota production system beyond large-scale production*. Diamond Inc.
- Oppenheim, B.W. (2004). Lean product development flow. *Systems Engineering*, 7(4), 352-376.
- Parnell-Klabo, E. (2006). Introducing Lean Principles with Agile Practices at a Fortune 500 Company. Teoksessa *Proceedings of Agile Conference* (Vol. 6, s. 232-242), IEEE.
- Panizzolo, R. (1998). Applying the lessons learned from 27 lean manufacturers: The relevance of relationships management. *International Journal of Production Economics*, 55(3), 223-240.
- Perera, G. & Fernando, M. (2007). Enhanced agile software development: Hybrid paradigm with Lean practice. Teoksessa *International Conference on Industrial and Information Systems*, 2007. ICIIS 2007 (s. 239-244). IEEE.
- Pérez, M. P. & Sánchez, A. M. (2000). Lean production and supplier relations: a survey of practices in the Aragonese automotive industry. *Technovation*, 20(12), 665-676.
- Petersen, K. (2010). *Implementing Lean and agile software development in industry*. Dissertation thesis, Blekinge Institute of Technology, School of Computing. Sweden. Haettu 20.11.2014 sivustolta: [http://www.bth.se/fou/forskinfor/nsf/0/45e1d377134f2ac4c12577200040034f/\\$file/Petersen_diss.pdf](http://www.bth.se/fou/forskinfor/nsf/0/45e1d377134f2ac4c12577200040034f/$file/Petersen_diss.pdf)
- Petersen, K. & Wohlin, C. (2010a). Measuring the flow of Lean software development. *Software: Practice and Experience*, 41(9), 975-996
- Petersen, K. & Wohlin, C. (2010b). Software process improvement through the Lean Measurement (SPI-LEAM) method. *The Journal of Systems and Software*, 83(7), 1275-2287.
- Petersen, K., Wohlin, C. & Baca, D. (2009). The waterfall model in large-scale development - state of the art vs. industrial case study. Teoksessa *Proceedings of the 10th International Conference on Product Focused Software Development and Process Improvement* (s. 386-400). Springer Berlin Heidelberg.
- Poppendieck, M. (2002). Principles of Lean Thinking. Teoksessa *17th Annual ACM Conference on Object-Oriented Programming, System, Languages and Applications*. Washington.

- Poppendieck, M. & Cusumano M. (2012). Lean software development: A tutorial. *Software, IEEE* 29 (5), 26-32.
- Poppendieck, M. & Poppendieck, T. (2003). *Lean software development: An agile toolkit*. Addison-Wesley Professional.
- Poppendieck, M. & Poppendieck, T. (2006). *Implementing Lean software development: From concept to cash*. Addison-Wesley Professional.
- Rajala, R., Rossi, M. & Tuunainen, V. (2003). A framework for analysing software business models. Teoksessa *European Conference on Information Systems 2003* (s. 1614-1627).
- Robert Abair Associates, Inc. (2002). *Lean Checklist Self-Assessment*. www.robertabairassociates.com
- Rodrigues, P., Markkula, J., Oivo, M. & Turula, K. (2012). Survey on Agile and Lean usage in Finnish software industry. Teoksessa *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (s. 139-148). ACM.
- Qumer, A. & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, 81(11), 1899-1919.
- Salo, O. (2006). Enabling Software Process Improvement in Agile Software Development Teams and Organisations. *VTT Publications*, 618.
- Schwaber, K. (1995). SCRUM Development Process Process, Teoksessa *Proceedings of OOPSLA'95 Workshop*.
- Scotland, K. (2009). Kanban, Flow, and Cadence. Haettu 28.05.2015 osoitteesta <http://availagility.co.uk/wp-content/uploads/2009/04/kfc-development-spa2009.pdf>
- Seikola, M., Loisa, H-M. & Jagos, A. (2011). Kanban Implementation in a Telecom Product Maintenance. Teoksessa *37th EUROMICRO Conference on Software Engineering and Advanced Applications* (s. 321-329). IEEE.
- Shah, R. & Ward, P. (2003). Lean manufacturing: context, practice bundles, and performance. *Journal of operations management*, 21(2), 129-149.
- Shalloway, A., Beaver G. & Trott, J. (2009). *Lean-Agile Software Development – Achieving Enterprise Agility*. Stoughton, Massachusetts: Pearson Education Inc.
- Sidky, A. & Arthur, J. (2007). A disciplined approach to adopting agile practices: The agile adoption framework, *Innovations in Systems and Software Engineering*, 3(3), 203-216.
- Staron, M. & Meding, W. (2011). Monitoring bottlenecks in agile and Lean software development projects - A method and its industrial use. Teoksessa *Product-Focused Software Process Improvement* (s. 3-16). Springer Berlin Heidelberg.
- Sureshchandra, K. & Shrinivasavadhani, J. (2008). Adopting agile in distributed development. Teoksessa *Proceedings of the 2008 IEEE International Conference on Global Software Engineering*, s. 217-221. IEEE.
- Vilki, K. & Erdogmus, H. (2012). Point/Counterpoint. *IEEE Software*, 5(29), 60-63.

- Williams, K., Haslam, C., Williams, J. & Sukhdeval, J. (1994). Deconstructing Car Assembler Productivity. *Teoksessa International Journal of Production Economics*, 3(34), 253-265.
- Wisconsin Manufacturing Extension Partnership. (2001). *How Lean is Your Culture?* <http://www.wmep.org>
- Wisconsin Manufacturing Extension Partnership. (2002). *Lean Business Assessment*. <http://www.wmep.org>
- Womack, J.P. & Jones, D.T., (1996). *Lean Thinking: Banish waste and create weath in your corporation*. New York: Free Press..
- Womack, J.P. & Jones, D.T. (2003). *Lean Thinking: Banish waste and create weath in your corporation - Revisited and Updated*. New York: Free Press.
- Womack, J.P., Jones, D.T., & Roos, D. (1990). *The machine that changed the world*. New York : Rawson Associates.