



Eetu Kaivola

# Ketterän viitekehyksen luonti kasvu- yrityksessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

22.8.2022

## Tiivistelmä

Tekijä: Eetu Kaivola  
Otsikko: Ketterän viitekehysten luonti kasvuyrityksessä  
Sivumäärä: 30 sivua  
Aika: 22.8.2022

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintätekniikka  
Ammatillinen pääaine: Ohjelmistotekniikka  
Ohjaaja: Janne Salonen, osaamisaluepäällikkö

---

Insinööriyössä tarkastellaan ketterän viitekehysten luomista kasvuyritykseen. Työ pitää sisällään teoriaosuuden, käytännön tarkastelun ja myös ratkaisun jalkauttamisen yritykseen. Tarkoituksena ei ollut luoda yleisesti toimivaa uutta viitekehystä, vaan tarkastella organisaatiota yksittäistapauksena.

Työssä on huomioitu alalla vallitsevat yleisesti tunnustetut parhaat käytännöt, sekä johtavien viitekehysten ja alan pioneerien lähestymiset ketterään ohjelmistokehitykseen. Työtä ei pidä sellaisenaan pitää uutena ketteränä viitekehystenä, vaan sitä pitää tarkastella muunnelmana olemassa olevista kehyksistä, joka on vain sovitettu organisaation tarpeisiin.

Työn alkuosassa keskitytään teoriaan ja esitellään ketterät menetelmät, jotka olivat mukana vertailussa ja käydään läpi ketterä manifesti. Tämän jälkeen pureuduttiin yrityksen rakenteeseen, ongelmiin ja mahdollisiin valmiisiin viitekehysiin. Seuraavaksi luotiin kustomoitu viitekehys yrityksen tarpeita vastaamaan ja lopuksi se jalkautettiin yritykseen. Havainnot kertovat onnistumisista, mutta myös haasteista, joita tämä projekti piti sisällään.

Avainsanat: Ketteryys, Scrum, skaalattu ketteryys

## Abstract

Author: Eetu Kaivola  
Title: Creating an agile framework for growth company  
Number of Pages: 30 pages  
Date: 22 August 2022

Degree: Bachelor of Engineering  
Degree Programme: Information Technology  
Professional Major: Software Engineering  
Supervisors: Janne Salonen, Head of the Department

---

This thesis focuses on creating an agile framework for growth company in Finland. It includes theory, practical views and solution that was implemented to the target company. The focus of this thesis was not to create a new universal framework but to understand and examine the organisation as an individual case.

In this thesis we have taken into consideration the commonly recognised best practices and leading frameworks but also the founding pioneers of agile thinking and approach that they have. This thesis should not be viewed as a new agile framework but as a modification of already existing frameworks fitted to the target organisation.

The beginning of this thesis focusses on theory and agile methodologies that were used as bases for the analysis. We also present the Agile Manifesto as integral part for the whole. After that the thesis focusses on company structure, problems and possible ready-made frameworks that could be implemented. Next is presented the customized or modified framework made for the company alone and how it was implemented. In the end results we discuss success but also failures and challenges that this project had.

Keywords: Agile, Scrum, Agile Framework

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Ketterä manifesti	2
3	Ketterät viitekehykset	4
3.1	Scrum	5
3.2	Kanban	7
3.3	SAFe	9
3.4	LeSS	11
4	Organisaation ja viitekehysten arviointi	12
4.1	Organisaation rakenne ja arviointi	12
4.2	Viitekehysten arviointi	14
5	Ketterän viitekehysten luominen	15
5.1	Ennustettavuuden parantaminen	17
5.2	Läpinäkyvyyden lisääminen	18
5.3	Päätösten vaikutusten tarkasteleminen	19
5.4	Yhteenveto	20
6	Viitekehysten sovittaminen vallitsevaan toimintaympäristöön	22
6.1	Tiimitason valmennus ja koulutus	23
6.2	Organisaation ketteryys ja strategisten päätösten läpivienti	24
6.3	Johdon tuki ja uusien lähestymistapojen arviointi	25
6.4	Käytännön toteutus	26
7	Havainnot	27
8	Loppupäätelmät	28
	Lähteet	30

## Lyhenteet

- PO: *Product Owner*, Scrumissa ja SAFe:ssa käytetty termi Tuoteomistajalle
- SM: *Scrum Master*. Scrumissa ja SAFe:ssa käytetty termi Scrum Masterille.
- KPI: *Key Performance Indicator*. Mittaustapa, jolla tutkitaan organisaation tehokkuutta
- SAFe: *Scaled Agile Framework*. Skaalattu ketterä viitekehys suurelle organisaatiolle.
- LeSS: *Large Scale Scrum*. Skaalattu ketterä viitekehys keskisuurelle organisaatiolle.

# 1 Johdanto

Tämä työ käsittelee muutosjohtamista ja ketterän viitekehyksen luomista sekä jalkauttamista kasvuyrityksessä. Alalla on monia skaalattuja viitekehyksiä, mutta jokainen organisaatio on erilainen. Tästä syystä teorian soveltaminen käytäntöön on aina muistettava tehdä parhaan mahdollisen lopputuloksen saavuttamiseksi. Tässä työssä on kuvattu teorian lisäksi yrityksen lähtötilanne, viitekehyksien kartoitus ja mukautetun ketterän viitekehyksen käyttöönotto.

Lähtökohtana työlle toimeksiantajan puolesta on ollut tarve virtaviivaistaa ja yhtenäistää yrityksessä vallitsevia käytäntöjä ketterän ohjelmistokehityksen suhteen. Tämä yhdistettynä kirjoittajan työnkuvaan loi otollisen tilaisuuden kirjoittaa tästä aiheesta.

Työn toimeksiantajana on ohjelmistoalan yritys, joka toimittaa asiakkaillensa SaaS-palvelua. Organisaatiossa työskentelee kirjoituksen ajanhetkellä noin 1500 työntekijää, joista teknologiaorganisaatiossa työskentelee noin 350. Teknologiaorganisaatio on jaettu itsenäisiin yksiköihin, jotka toimivat rinnan ajaen teknologisia ratkaisuja eteenpäin

Ketteryys on ollut arkipäivää ohjelmistoalalla jo vuosia. Tämän takia alalle on muodostunut suuri määrä erilaisia viitekehyksiä tukemaan ketterää työskentelyä. Tässä työssä käsittelemme eri viitekehyksiä arvioiden niiden soveltuvuutta kasvuyritykseen, jonka koko on ylittänyt tuhat henkilöä. Tästä henkilömäärästä noin kolmasosa työskentelee tavalla tai toisella ohjelmistokehityksen parissa.

Tutkimusmenetelmänä työtä tehdessä on käytetty kirjoittajan ammatillista asemaa toimeksiantajan yrityksessä, jossa empirian kautta teorian soveltaminen käytäntöön on toteutettu osana kirjoittajan työnkuvaa. Tutkimuksen aineiston arkaluontoisuuden vuoksi ja insinöörityötä koskevan julkisen lainsäädännön johdosta tässä työssä ei mainita yritystä, eikä toimialaa yksityiskohtaisesti. Työ

keskittyy enemmän arvioimaan ketteriä viitekehyksiä ja niiden soveltamista käytäntöön.

Viitekehysten määrittäminen on suuresti riippuvainen yrityksen vallitsevasta vaiheesta, tiimien, organisaation ja henkilöstön kypsyydestä. Aluksi esittelemme tätä työtä varten valitut viitekehukset ja arvioinnin jälkeen kerromme kuinka ketterä viitekehys tässä kyseisessä tapauksessa luotiin, mitkä olivat siihen vaikuttaneet tekijät ja miten sitä sovellettiin. Muutosjohtaminen on vahvasti läsnä aina, kun kyseessä on jonkin uuden toimintamallin tai -tavan jalkauttaminen yritykseen, joten sen käsittely tämän työn ohessa on luonnollista. Tutkimuksen aiheistona on käytetty alan johtavien koulutusorganisaatioiden materiaaleja kuvaamaan viitekehyksiä mahdollisimman tarkasti. Lisäksi alan kirjallisuutta on hyödynnetty.

## 2 Ketterä manifesti

Ketterän ohjelmistokehityksen alku on vaikea määrittää, mutta yleisesti on hyväksytty käytettäväksi vuotta 2001, jolloin Ketterä manifesti (engl. *Agile Manifesto*) kirjoitettiin. Sen kirjoittajat olivat 17 merkittävää vaikuttajaa, jotka kokoontuivat Utahissa kirjoittamaan tämän manifestin. Manifesti pitää sisällään neljä arvoa, sekä kaksitoista periaatteita, joita menetelmät noudattavat. Manifesti kuuluu näin:

Ketterän ohjelmistokehityksen julistus

Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme:

Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja  
Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota  
Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja  
Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän. (1)

Manifestin ohessa on listattu myös periaatteet, jotka kuuluvat olennaisesti osaksi manifestia. Ne ovat nämä:

Tärkein tavoitteemme on tyydyttää asiakas toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti

Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyyn edistämiseksi.

Toimitamme versioita toimivasta ohjelmistosta säännöllisesti, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä.

Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.

Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä.

Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.

Toimiva ohjelmisto on edistymisen ensisijainen mittari.

Ketterät menetelmät kannustavat kestävään toimintatapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen.

Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä.

Yksinkertaisuus - tekemättä jätettävän työn maksimointi – on oleellista.

Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä.

Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan ja mukauttaa toimintaansa sen mukaisesti. (1)

### 3 Ketterät viitekehukset

Ketterien viitekehysten voidaan katsoa alkaneen ohjelmistoalalla 90-luvulla. Kiihdyttävänä tekijöinä toimivat Toyotan filosofia, jossa tekijä vetää työtä itselleen eikä niinkään saa sitä työnnettynä prosessin tuotteena. Tätä ajattelua kutsutaan nimellä Lean. Toyota pyrki tehostamaan tuotantoketjunsä toimintaa ja poistamaan turhia työvaiheita, jotta valmistusprosessi olisi nopeampaa. Vaikka prosessi on kehitetty autonvalmistukseen, sitä on sovellettu menestyksekkäästi niin ohjelmistoalalla kuin terveydenhuollossa, sekä monessa valmistavassa teollisuudenalassa. Leanin peruseriaatteisiin kuuluu arvontuoton tunnistaminen, arvovirtojen kartoitus, työskentelyvirran luominen, vetävän työskentelytavan vaikiinnuttaminen ja pyrkimys täydellisyyteen.

Tässä luvussa arvioidaan useampaa viitekehystä, joista osa keskittyy vain johonkin osuuteen tuosta filosofiasta. Alalla vallitsee kuitenkin vahva konsensus siitä, että Lean ja ketteruus kulkevat rinnakkain ohjelmistokehitykseen keskittyvissä yrityksissä. Nämä kaksi täytyy kuitenkin muistaa erottaa toisistaan, kun puhutaan ketterästä viitekehuksesta. Vaikka pohja perustuukin Leaniin, osa näistä toimintatavoista on keskitetty vain yhteen osa-alueeseen siinä, jolloin koko Lean-mallin soveltamisesta ei voida puhua.

Seuraavissa luvuissa käsitellään tätä työtä varten valitut viitekehukset, mutta ennen sitä esitetään muutama perustelu valinnoille. Kaksi ensimmäistä ovat fundamentaalisia tapoja kehittää ohjelmistoja ja käytössä tavalla tai toisella melkein jokaisessa ohjelmistoalan yrityksessä. Ne myös luovat pohjan skaalatuille viitekehyksille, jotka on luotu ratkaisemaan sellaisia skenaarioita, joissa yksi tiimi ei enää kykene hallitsemaan koko kehitystä. Jokainen käsitelty kehys on teoreettiselta tasolta riittävä ratkaisemaan sille asetetut ongelmat, mutta käytännön kokemus on kuitenkin osoittanut, että puhtaasti teoreettinen lähestymistapa on reaali maailmassa mahdoton toteuttaa. Esimerkiksi organisaatiomalli, jolle SAFe on luotu, on ollut olemassa vain ja ainoastaan siinä yrityksessä. Alati muuttuva ja kehittyvä organisaatio ei noudata teorian sääntöjä kompleksisuutensa ja inhimillisten tekijöidensä johdosta.

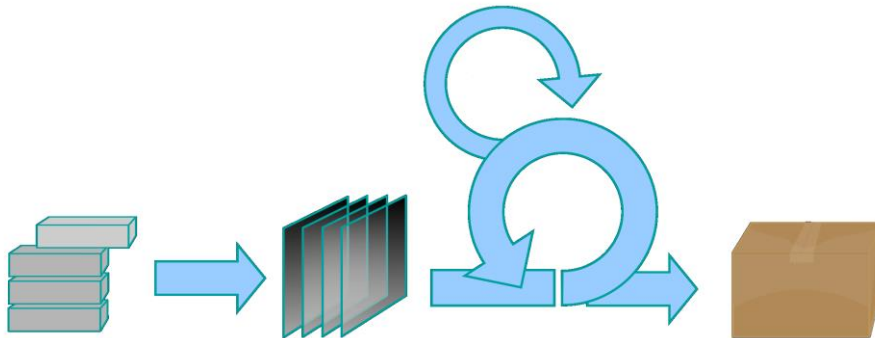
### 3.1 Scrum

Scrum on ohjelmistokehityksessä usein käytetty viitekehys, joka perustuu toistuviin seremonioihin ja päälle lisäämiseen (engl. *iterative-incremental*). Tämä tapa pyrkii parantamaan ennustettavuutta ja hallitsemaan riskejä. Scrumin seremoniat toistuvat sprintistä toiseen muuttumattomina ja sen tavoitteena on aina toimittaa sprintin päätteeksi toimiva ohjelmisto. Tämä ei tarkoita sitä, että ohjelmisto olisi täysin uudestaan kirjoitettu, vaan pikemminkin sitä, että lisäämällä siihen ominaisuuksia inkrementaalinen lähestymistapa toteuttaa tarkoitustaan lisäten toimivaan ohjelmistoon lisää ominaisuuksia.(2)

Scrumin vahvuutena voidaan pitää sen yksinkertaista toimintamallia, jossa liikkuvien palasten lukumäärää on karsittu pyrkien säilyttämään vain oleelliset. Scrum antaaakin vallan tiimille, jonka koostumus määritelmällisesti kattaa kaiken sen tarvittavan osaamisen, jota tiimi senhetkisen sprintin maaliin viemiseksi tarvitsee. Tätä kutsutaan tässä viitekehityksessä nimellä moniosaajatiimi (engl. *Cross-functional team*). Palaverien ja muiden kokousten määrää ei Scrumissa ole rajoitettu, mutta lähtökohtaisesti pakollisia palavereja on vain rajattu määrä. Niistä aikaa vievimmat keskittyvät sprintin aloitukseen ja lopetukseen, antaen kehitystiimille mahdollisimman paljon keskeytyksetöntä aikaa tuottaa toimivaa ohjelmistoa.(2)

Heikkoudeksi voidaan tässä kohtaa mainita Scrumin huono soveltuvuus ylläpitotyöhön. Lähtökohtaisesti sprintille otetaan vain ja ainoastaan ne asiat, joita sprintin aikana toteutetaan, eikä sinne lisätä sprintin aikana uusia asioita. Tarkennuksia voidaan tehdä ja työmäärän lisääntyessä luoda kokonaisuudelle uusia työtä vaativia asioita, mutta kaikki tämä toimii kuitenkin jo suunnittelussa määritellyn sisällön puitteissa. Siksi äkilliset virhetilanteet ja vaikkapa palvelupyynnöt, joita ylläpitovaiheessa oleva ohjelmisto saa paljon enemmän verrattuna kehityksessä olevaan ohjelmistoon, joutuvat odottamaan seuraavaa sprintin suunnittelua. Tässä täytyy muistaa, että tiimi voi määrittellä itselleen kaistaa omasta työnteostaan palvelupyynnöille ja muille äkillisille asioille, mutta se ei palvele tarkoitustaan, jos työstä suuri osa koostuu tällaisista tehtävistä.

Scrumin on ohjelmistokehityksen kehitysvaiheeseen suunniteltu toimintatapa, joka tukee uuden tuotteen ja sen ominaisuuksien kehittämistä iteratiivisuutensa puolesta erittäin hyvin. Siksi tämä toimintamalli on kansainvälisesti tunnustettu yhdeksi parhaimmista tavoista tuottaa ohjelmistoja, ja sitä voi kutsua jo alan normaaliksi toimintatavaksi Kanbanin ohella. Alla on esitelty Scrum-prosessin visuaalinen kuvaus ja sanallisesti selvitetty toistuvat tapahtumat, eli seremoniat.



Kuva 1 Scrum-prosessi

### Scrumin seremoniat

Sprintin aloitus tapahtuu sprintin suunnittelulla. Tässä tapahtumassa kehitystiimin jäsenet ja PO keskustelevat tulevan sprintin tavoitteesta ja muodostavat tuotteen kehitysjonon perusteella sprintille kehitysjonon. Lisäksi tässä seremoniassa määritellään sprintin tavoite, joka on linjassa tuotteen tavoitteen ja sidosryhmien tavoitteiden kanssa.

Scrumin pohjana sprintissä on Daily Scrum, eli päivittäinen viidentoista minuutin palaveri, jonka tarkoituksena on mukauttaa sprintin suunnitelma seuraavan vuorokauden ajaksi.

Sprintin päättyessä tiimillä on käsissään mahdollisesti julkaisukelpoinen ohjelmisto, joka PO:n päätöksellä hyväksytään tai ei hyväksytä. Tätä seremoniaa

voidaan kutsua sprintin katselmoinniksi, mutta puhekielessä siitä yleensä käytetään nimitystä "demo".

Katselmoinnin jälkeen kehitystiimi pitää retrospektiivin, jossa käydään läpi onnistumiset ja parannettavat asiat. Tämän jälkeen sprintti on suoritettu loppuun ja uusi sprintti voidaan aloittaa. (3)

Sprintin aikana tapahtuu myös muita asioita, jotka eivät virallisesti ole osana seremonioita, mutta liittyvät pitkälti PO:n ja SM:n työhön. Näitä ovat muun muassa kehitysjonon jalostaminen tulevia sprinttejä varten. Esteiden poistaminen, jotta kehitystiimillä on mahdollisuus saavuttaa sprintin tavoite. Koska nämä eivät ole virallisia seremonioita, niiden osuutta ei tässä luvussa käsitellä tarkemmin, mutta ne ovat kuitenkin mainitsemisen arvoisia asioita.

## 3.2 Kanban

Kanbanin voidaan katsoa alkaneen jo 1940-luvulla, kun Toyota alkoi optimoida tuotantoaan käyttäen tätä toimintatapaa. Kanbanin perusajatuksena on virtauksen optimointi, jolla pyritään minimoimaan varastojen tarve ja hallinnoimaan koko tuotantoketjua alusta loppuun siten, että jätettä ei synny. Tässä tapauksessa jäte voi olla varastointia, aikaa, materiaalihukkaa tai muuta. Ohjelmistokehityksen maailmassa varastointi tai raaka-aine on harvoin optimoinnin kohteena, vaan siinä pyritään optimoimaan koodintuoton arvovirtaa.

Scrumin ohella Kanban on toinen ohjelmistokehitysmaailmassa yleisesti käytetty toimintamalli. Sen vahvuudet tulevat parhaiten esiin ohjelmistokehityksen myöhäisemmissä vaiheissa, joissa tuottelle tehdään ylläpitoa ja jatkokehitystä. Läpimenoaikaa mittaava toimintatapa, jonka priorisointia voidaan tehdä jatkuvasti, eikä pelkästään kehitysjakson alussa.

Kanbanin heikkoutena voidaan pitää priorisointia, jonka perusteella työjono luodaan. Varsinkin ylläpitovaiheessa olevien työtehtävien tärkeysjärjestyksen asettaminen, työjonon tekeillä olevien tehtävien määrän rajoittaminen ja äkilliset

virhetilanteet aiheuttavat usein tilanteita, joissa tehtävää -sarakkeessa on liian paljon asiaa tiimin suorituskykyyn verrattuna. Siksi Kanban soveltuu parhaiten erittäin kurinalaisille ja kokeneille kehitystiimeille.

Kanbanin tärkein metriikka on läpimenoaika, jolla voidaan selvittää, onko prosessissa jotain vialla tai missä kohtaa pullonkaula sijaitsee. Tätä optimoimalla prosessista pyritään saamaan mahdollisimman sujuva ja samalla ennustamaan kuinka kauan työtehtävissä kestää alusta loppuun. Kanban toteuttaa prosessissaan oikea-aikaisuutta (engl. *just in time, JIT*). Tämä tarkoittaa sitä, että työjono on järjestetty tärkeimmän työn mukaan, joka suoritetaan aina ensimmäisenä. Kanbanista tutuin on myös muiden viitekehyksien käyttämä taulu, jossa työtehtäviä järjestellään.

Yksinkertaisimmillaan Kanban on vain työjono, josta vedetään työtehtäviä edellisen valmistuttua. Tässä menetelmässä on kuitenkin olemassa myös nykypäivänä yleisesti tunnistettuja piirteitä, joita voi verrata Scrumissa käytettävään suunnitteluun tai työjonon jalostamiseen. Kanban kuitenkin perustuu siihen, että työjonon ensimmäinen tehtävä on aina oikea ja työjono on oikeassa järjestyksessä. Monissa Kanban-tauluissa on myös visualisoitu niin kutsuttu tulipalo-kaista. Tämä tarkoittaa yllättäen ilmaantuvaa yleensä kriittistä virhetilannetta, jolloin koko kehitystiimin täytyy lopettaa se mitä olivat tekemässä ja siirtyä "sammuttamaan" tätä tulipaloa. Kaikki muu työ pysähtyy, kunnes tämä työtehtävä on saatu valmiiksi.

Kanbanin toimintamallille on yleistä, joskaan ei pakollista määritellä maksimi jokaiselle työvaiheelle. Läpimenon optimoinnin kannalta tämä on äärimmäisen tärkeää. Muuten on riskinä työtehtävien toteuttamisen viivästyminen ja priorisoinnin vääristyminen. Varsinkin monitoimittajaympäristössä Kanbanin pullonkaulat nousevat helposti esille, mutta jos niihin ei kiinnitetä huomiota koko viitekehys menettää suurimman lisäarvonsa, joka on läpimenoajan optimointi.(4)



Kuva 2 Kanban taulu, jossa neljä saraketta

### 3.3 SAFe

Scaled Agile Framework on yksi vallalla olevista skaalatuista ketteristä viitekehysistä. Kuten muutkin skaalatut viitekehukset, se on luotu ratkomaan ongelmia, joita organisaatiot kohtaavat kasvaessaan yhdestä sovelluskehitystä tekevästä tiimistä useampaan tiimiin. Toinen hyvin tätä viitekehystä kuvaava ominaisuus on yritysten suunnitelmahorisonttien huomiointi. Vaikka Scrum itsessään asettaa tuotteelle tavoitteen ja aikataulun, SAFe pyrkii luomaan avovirtoja, jotka toimivat yhteisessä kadenssissa koko kehitysorganisaation kanssa.

SAFe keskittyy tiimitasolla hyväksi havaittuun Scrum-malliin, mutta rakentaa sen ympärille uuden viitekehysten, joka on tarkoitettu synkronoimaan toimitus usean tiimin yli. Tästä SAFe käyttää nimeä Release Train. Kuten Scrum myös SAFe käyttää iteratiivista lähestymistapaa, mutta toisin kuin Scrum, SAFe ottaa mukaan pidemmän tähtäimen tavoitteet. Program Incrementit. Tällä yritetään huomioida usean tiimin toimittavat toisiinsa liittyvät kokonaisuudet, jotka Scrumissa jätetään huomiotta, koska se keskittyy vain yhden tiimin ympärille.

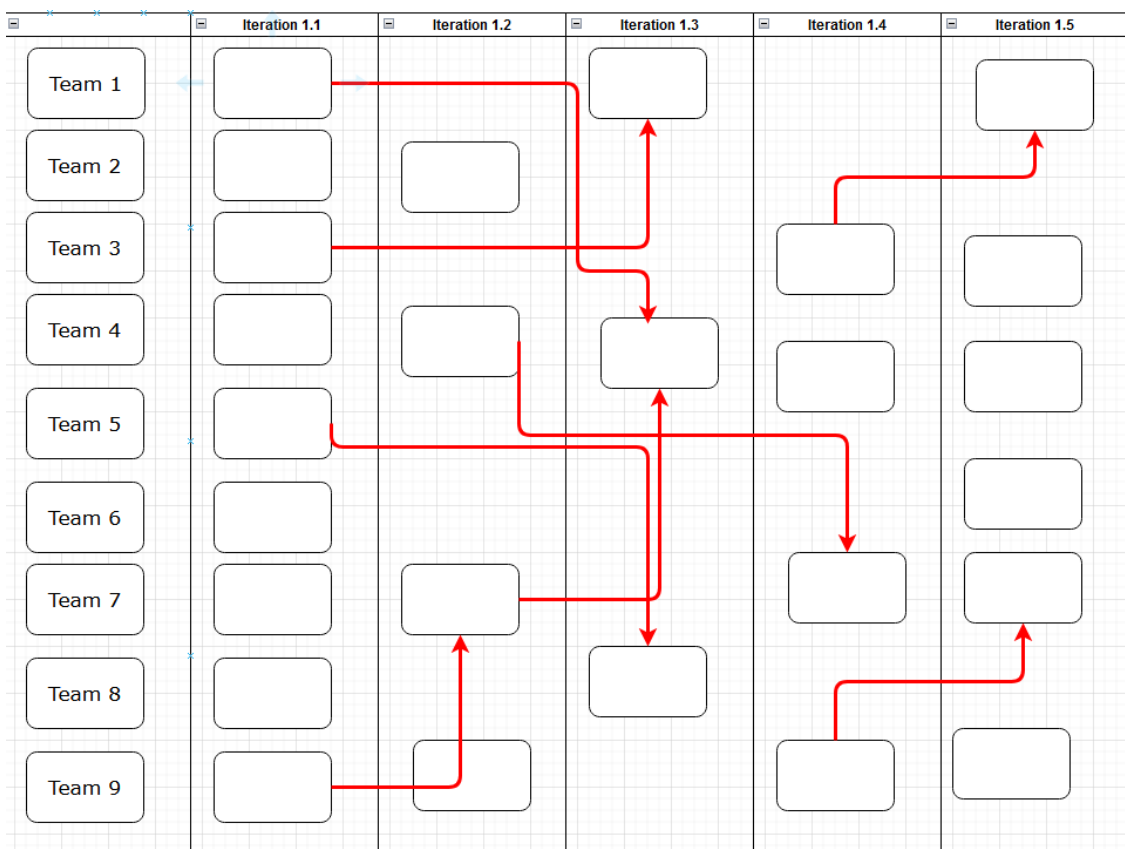
Työskentely monen tiimin yli aiheuttaa paljon toteuttavaa ohjelmakoodia, uusia ominaisuuksia ja bugikorjauksia. Tästä syystä SAFe käyttää arjessaan CDE-mallia, eli jatkuvan toimituksen mallia. Tämä perustuu siihen lähtökohtaan, että

jokaisella tiimillä on mahdollisuus tutkia, integroida ja julkaista ohjelmakoodinsa heti kun sen valmiin määritelmä on täytetty. Siten tuote on jatkuvasti uusimmassa versiossaan. Muistettava on, että tämä malli ei välttämättä näy tuotteen toiminnassa asiakkaalle ollenkaan, koska versiopäivitykset ajetaan ohjelmistoihin yleensä suunnitellusti ja aikataulutetusti. Asiakkaan tiedottaminen, palautteen kerääminen ja tarpeiden kartoitus täytyy suorittaa onnistuneesti, jotta SAFe täyttää jatkuvan lisäarvon tuottamisen, sekä välttää hukan tekemisen.

## SAFen tärkeimmät tapahtumat

Ylätasolta, tässä tapauksessa johdon suunnalta SAFe määrittää strategiset teemat, sekä strategisen portfolion. Tämä pitää sisällään arvovirtojen määrittämisen, budjetoinnin, korkean tason työjonon seurannan ja jaottelun kokonaisuuksiin. Viitekehys pyrkii huomioimaan jatkuvan arvontuoton sekä mittariston, jota yritykselle on asetettu. Tämä toimina voidaan katsoa jatkuvana, sillä se on yrityksen johdon yksi avaintehtävistä.

Alemmalle tasolle mentäessä puhutaan jatkuvasta toimittamisesta (engl. *Continuous Delivery Pipeline*), joka koostuu PI-kokonaisuuksista. Nämä kokonaisuudet ovat kasattu yhteen PI-työjonoon, josta kokonaisuutta voidaan seurata viitekehysten asettamien roolien toimesta. Näitä rooleja ovat RTE, järjestelmäarkkitehti, tuotehallinta ja liiketoiminnan avainhenkilöt. Tämän kokonaisuuden tärkein tapahtuma on PI-suunnittelu. Siinä koko toteuttava organisaatio suunnittelee samanaikaisesti seuraavan inkrementin työjonon jokaiselle tiimille. Tämä suunnitelma pitää sisällään mahdolliset julkaisut, tutkimusta vaativat asiat, riippuvuudet, sekä kaiken mahdollisen työn mitä toteuttava organisaatio pystyy senhetkellä tiedoilla kertomaan, lisäksi tilaa on jätettävä yllätyksille. PI-suunnittelussa suuri huomio kiinnitetään siihen, kuinka paljon tiimien välillä on riippuvuuksia ja miten niitä ratkaistaan. Riippuvuuksien vähentäminen ja ratkaiseminen on avainasemassa organisaatiossa, jossa on useita tiimejä toimimassa saman tuotteen ympärillä. (5)



Kuva 3 SAFe:n riippuvuustaulu

### 3.4 LeSS

LeSS on toinen tunnettu skaalatun ketteryuden viitekehys. Tätä kehystä käytetään yleisesti hieman pienemmissä organisaatioissa, joissa kuitenkin on useampi tiimi tuottamassa ohjelmakoodia. Myös LeSS on rakentunut yhden yhteisen työjonon pohjalle. Arvovirrat ja tuotehorisontti ovat keskeisessä osassa myös tässä viitekehyksessä, mutta se pyrkii pitämään tuotekehityksen mahdollisimman virtaviivaisena ja Scrumin peruseräjäiteitä noudattavana. Siitä myös tulee viitekehityksen nimi "*Large-Scale Scrum*".

LeSS:n keskiössä on Scrumista tuttu tuotteen työjono, jonka pohjalle organisaatio on rakentunut. Roolit ovat samankaltaiset Scrumin kanssa ja toisin kuin SAFe, LeSS pyrkii minimoimaan uusien roolien luomisen organisaatioon. Tämä nähdään tässä viitekehityksessä ongelmallisena ja se pyrkii välttämään

tiukasti määritellyn prosessin luomista organisaatioon. Keskiössä on ennemminkin empiirinen prosessi, jonka tuotteen avainhenkilöt luovat.

Prosessikehityksessä päädytään yleensä luomaan yleinen viitekehys, jonka organisaatio ottaa käyttöön kokonaisuudessaan. LeSS lähestyy tätä ongelmaa täysin toiselta kannalta ja tunnistaa, että kokonaisuudessaan käyttöönotettu viitekehys luo usein organisaatioon rooleja, joilla ei ole lopputuloksen kannalta suurta merkitystä. Tämä sijasta tarkoituksena on rakentaa mahdollisimman kevyt organisaatio, jonka tehtävänä on tukea ohjelmistokehitystä ja tarjota sille mahdollisuudet edetä parasta mahdollista vauhtia. Tämä perustuu usein kysymykselle “Miksi tämä prosessi on olemassa?” jota seuraa “Jos meillä on useampi tiimi, miten voimme saavuttaa saman tavoitteen isossa skaalassa?”

LeSS pyrkiikin luomaan mahdollisimman vähän uutta Scrumin toimintaan, ottaen kuitenkin huomioon sen faktan, että tiimien määrä ja samalla henkilöiden ja tuotettavan ohjelmakoodin määrä kasvaa huomattavasti. Tätä tukemaan luotavien roolien täytyy siksi olla perustettavissa yleispätevästi.(6)

## **4 Organisaation ja viitekehysten arviointi**

Pelkkien viitekehysten arvioinnilla ja esittelyllä ei voida vielä kokonaisvaltaisesti todeta jonkin asian toimivuutta. Organisaation rakenne, toimintatavat ja valinnat ovat myös vaikuttava tekijä viitekehysten valintaan. Organisaatiota arvioidessa on pyritty mahdollisimman tarkasti avaamaan rakenteet ja ohjaavat tekijät, jotka vaikuttavat viitekehysten valintaan. Tämä voi tuoda lisätietoa tai tukea tietyille valinnoille, jotka muuten puhtaasti teoreettiselta pohjalta saattaisivat poiketa paljonkin luodusta ratkaisusta.

### **4.1 Organisaation rakenne ja arviointi**

Ensin esitellään teknologiaorganisaation rakenne ja sen toimintamallit. Teknologia tiimit oli jaoteltu osastoittain siten, että kaikkien tuotteiden ympärille oli katsuttu tarvittavat tiimit ohjelmistokehityksen sujumuuden varmistamiseksi. Lisäksi

organisaatiolla oli yksi osasto, joka oli vastuussa jaettujen komponenttien ja työkalujen kehittämisestä. Organisaatiotasolla toimintaa ohjasi yrityksen teknologiajohtaja (CTO), jolle suoraan vastasivat osastojen johtajat. Nämä henkilöt olivat vastuussa osastojen yleiskuvasta, sekä operatiivisen ja teknologisen tietämyksen toteutuksen varmistamisesta. Jokainen osasto oli jaettu tiimeihin, joilla oli oma vastuualueensa ja tiimiesimies. Nämä ihmiset vastasivat päivittäisestä teknisestä tekemisestä. Jokainen tiimi oli rakennettu ketterää ajatusmaailmaa silmällä pitäen sisältämään kaiken tarvittavan osaamisen, ottaen huomioon sen hetkisen tavoitteen.

Teknologiaorganisaation rinnalle on rakennettu tuoteorganisaatio, joka omalta osaltaan vastasi ominaisuuksien priorisoinnista ja niiden ei-tekniisten vaatimusten laatimisesta, kommunikoimisesta ja tarkentamisesta tuotteen kehitysprosessin aikana. Tiimien tuoteomistajat kuuluivat tähän organisaatioon, joka jo tässä vaiheessa poikkeaa aiemmin esitellyistä viitekehyksistä, joissa tuoteomistaja on osana tiimiä ja kuuluu samaan organisaatioon kehitystiimien kanssa. Tämän jaon taustalla oli ajatus yhtenäisestä tuotteesta ja tuoteorganisaatiosta, jolloin kommunikaatio eri osastojen välillä helpottuisi tuoteomistajien ryhmäytymisen johdosta. Yrityksen tekemät hankinnat, tässä tapauksessa yritysostot, vaikuttivat myös tähän valintaan. Kansainvälisesti toimiva yritys halusi pitää jokaisen tiimin osana kokonaisorganisaatiota ja välttää sijaintipohjaisten siilojen muodostumisen, jolloin ratkaisuksi oli päädytty yhdistämään tuoteorganisaatio yli maa-rajoiden.

Kolmantena merkittävänä organisaationa tämän työn näkökulmasta on muotoilu. Tämä osasto vastasi tuotteen yleisestä visuaalisesta ilmeestä ja tarjosi palveluksiaan teknologia- ja tuoteorganisaatiolle tarpeen vaatiessa. Tällä tavoin varmistettiin yleisilmeen säilyvyys. Muotoilijat ja suunnittelijat liittyivät tarpeen vaatiessa osaksi tuotetiimiä tai kehitystiimiä suunnitellen ulkoasun ja käytettävyyden vastaamaan koko yrityksen ilmettä ja toimintatapaa.

Tässä vaiheessa oli selvää, että mitään olemassa olevaa viitekehystä ei voisi suoraan hyödyntää prosessissa, koska organisaation rakenne oli poikkeava

viitekehysten teoriapohjaan nähden. Koska kyseessä on kasvuyritys, organisaation rakenne muuttuu yrityksen kasvaessa väkisinkin ja tuleva rakenne saattaa poiketa tässä esitetystä huomattavasti. Siksi lähtötilanteessa oleva pohja otettiin huomioon enemmän suuntaa antavana kuvauksena, kuin lopullisena rakenteena, jossa yritys toimii.

Organisaation jatkuvan muutoksen tuoma epävakaus oli yksi suurimmista vaikuttavista tekijöistä kartoitusta tehdessä. Rakenteet, joita organisaatioon tuotiin, pyrkivät ottamaan huomioon tulevaisuudessa tapahtuvan henkilömäärän kasvun ja organisaation kompleksisuuden lisääntymisen tarjoamalla rakenteet ennen kuin niille välttämättä olisi tarve. Tämä lähestyminen olisi oman selvityksen arvoinen tutkimuskohde. Tässä vaiheessa tyydyn toteamaan, että tällä lähestymisellä on puolensa, hyvät ja huonot.

## 4.2 Viitekehysten arviointi

Alalla vallitsevien toimintatapojen vakiinnuttua jo useita vuosia sitten, nämä käytössä olevat viitekehukset ja niiden käyttäminen tiimien päivittäisessä työskentelyssä on täysin perusteltua. Teknologiaorganisaatio, joka ulottuu ohjelmistokehityksestä ylläpitoon ja lopulta tuotteen hallittuun alasajoon tarvitsee tiimeiltänsä mukautuvuutta. Kehitysvaiheessa oleva ohjelmisto hyötyy huomattavasti enemmän Scrumin käytöstä sen iteratiivisuuden vuoksi.

Edellisessä pääluvussa esitellyt viitekehukset käytiin läpi tarkasti ja arvioitiin niiden soveltumista yrityksen teknologiaorganisaatioon. Lopputuloksena on, että viitekehyksistä Scrum ja Kanban soveltuvat yrityksen käyttöön tiimitasolla sellaisenaan, eikä niiden suhteen tarvitse tehdä muutoksia. Organisaatiotasolla kumpikaan valituista kahdesta viitekehyksestä ei tarjonnut riittävän kattavaa ja yrityksen toimintaan sopivaa ratkaisua. SAFe:n laajuus rooleineen ja monine muine prosesseineen koettiin kokonaisuudessa liian isoksi muutokseksi, lisäksi sen tuottama lisäarvo kyseenalaistettiin niin kirjoittajan kuin yrityksen teknologiaorganisaation johdon puolesta. Periaatteen tasolla organisaation koko tukisi

SAFe:n käyttämistä viitekehystenä, mutta jo aiemmin mainittu merkityksettö-  
mien roolien määrä osoittautui liian suureksi.

LeSS on suunnattu organisaatioille, joilla on noin kahdeksan tiimiä tuotteen ympärillä. Vaikka teknologiaorganisaatio on jaettu neljään osastoon, jokaisessa niissä on kuitenkin lukumäärällisesti vähintään kahdeksan tiimiä. Tästä johtuen tilalle pitäisi ottaa skaalattu versio kyseisestä viitekehystä. Tähän tutustumisen jälkeen lopputulos kuitenkin oli, että myöskään LeSS ei tarjoa riittävän kattavaa ja yrityksen toimintamalliin sopivaa viitekehystä tukemaan toimintaa. Täydellisyyteen pyrkiminen olisi mahdottomuus, mutta kummankin skaalatun ketterän viitekehysten sopimattomuus oli liian suurta. Arvioinnissa päädyttiin lopputulokseen, jossa yritykseen luodaan muokattu skaalautuva ketterä viitekehys. Tämän päätöksen tukena oli myös osastojen onnistunut toiminta jo olemassa olevilla toimintatavoilla, joten niiden purkaminen nähtiin turhana. Näitä toimintamalleja, sekä skaalattuja viitekehymiä hyödyntämällä aloitettiin oman viitekehysten rakentaminen.

## **5 Ketterän viitekehysten luominen**

Päätös mukautetun skaalautuvan viitekehysten luomisesta loi mahdollisuuden yrittää muokata olemassa olevista viitekehyksistä yritykselle paremmin sopiva kokonaisuus. Työ aloitettiin arvioinnin pohjalta tehtyjen johtopäätösten läpikäynnillä ja nykyisten toimintatapojen kartoituksella. Näiden luoman kokonaisuuden katsottiin riittävän ensimmäiseen versioon mukautetusta kehyksestä.

Nykyisten toimintatapojen kartoitus paljasti, että kaksi osastoa työskenteli jo osittain SAFe-mallia. Tämän katsottiin luovan riittävä pohja sille, että perustana mukautetulle viitekehykselle toimisi enemmän SAFe, eikä niinkään muut alalla olevat viitekehukset. Nämä reunaehdot huomioiden luominen helpottui, koska vallitsevaa toimintatapaa ei tarvinnut ottaa pois käytöstä. Huomioitavaa on myös se, että vaikka osastot työskentelivät noudattaen SAFe:a, sitä ei missään osassa organisaatiota ollut otettu käyttöön täydessä laajuudessaan. Jos näin olisi ollut, ratkaisuna SAFe-mallin jalkauttaminen kokonaisuudessaan muihinkin

osiin organisaatiota olisi ollut varteenotettava vaihtoehto, sen monimutkaisuudesta huolimatta. (5)

Keskustelu johdon kanssa viitekehuksesta aloitettiin selkeyttämällä organisaation tavoitteita. Strategisten ja teknisten tavoitteiden tueksi toivottiin viitekehysten pystyvän tuottamaan luotettavaa dataa edistyksestä, tuomaan läpinäkyvyyttä organisaatioon ja mahdollisuutta tarkastella tehtyjen päätösten vaikutusta koko teknologiaorganisaatiossa. Tärkeimpänä kuitenkin pidettiin arvontuottoa. Arvon mittaaminen on jokaisen ketterän viitekehysten ytimessä. Scrumin sprintti päättyy toimivaan ohjelmakoodiin, joka on priorisoitu ja hyväksytetty sidosryhmillä. Tuoteomistajan yksi tärkeimmistä tehtävistä on ymmärtää ominaisuuden tuottama arvo loppuasiakkaalle ja siten priorisoida tiimin tekemistä. Tämän pohjalta pystytään luomaan myös KPI-mittarit tukemaan liiketoimintaa.

Nämä kolme teemaa, edistyminen, läpinäkyvyys ja päätösten vaikutus tarvitsevat taustalleen dataa. Viitekehysten on siis pystyttävä tuottamaan mitattavaa dataa, josta voidaan arvioida ja tehdä johtopäätöksiä edellä mainittuihin asioihin liittyen. Nämä teemat leikkaavat läpi koko organisaation, mutta on otettava huomioon, kuinka hienojakoista dataa milläkin organisaation tasolla tarvitsee käsitellä. Ensimmäisenä toimenpiteenä kartoitettiin tiimitasoa.

Tiimitason kartoituksessa selvisi, että organisaation jokainen tiimi toimi joko Scrumin tai Kanbanin viitekehysten puitteissa. Tässä kohtaa pyrimme kartoittamaan, olivatko samaa viitekehystä käyttävät tiimit samassa osassa teknologiaorganisaatiota. Paljastui, että osastojen sisällä valinnat viitekehuksesta oli tehty itsenäisesti, joten jokainen tiimi oli valinnut käyttääkseen senhetkisillä tiedoilla parhaan mahdollisen viitekehysten. Yksi vaihtoehto edistymisen seurannan viraviivaistamiseksi olisi ollut tuoda vain toinen viitekehys tiimien käyttöön. Tämä vaihtoehto kuitenkin suljettiin pois, koska osastojen sisällä on olemassa niin uuden kehitystä kuin ylläpitotyötä tekeviä tiimejä. (2,7)

## 5.1 Ennustettavuuden parantaminen

Datan yhdenmukaistaminen eri viitekehysten takia ei siis ole mahdollista. Kuitenkin Scrum ja Kanban molemmat keskittyvät työjonossa suoritettavien tehtävien valmistumiselle. Tämä hyväksyttiin metriikaksi tiimitasolle. Sen hyödyntämiseksi organisaatiossa aloitetaan tiedon seuranta ja kerääminen, jotta niiden pohjalta voidaan tulevaisuudessa tehdä ennustuksia. Tämän mittarin datan tuottamiselle annettiin tiimeille vain vaatimus kirjata jokaisen sprintin tulos ylös Jiran raportteihin. Tämä ei yhdenmukaista kaikkien tiimien tuottamaa dataa, mutta tarjoaa mahdollisuuden arvioida yksittäisen tiimin edistymistä ja nopeutta verraten sitä itseensä. Silloin saamme kuvan siitä mitä jokainen tiimi tuottaa tietyn ajanjakson aikana.

Tästä seuraava looginen askel oli tarjota tuoteomistajille tämä sama data ja luoda heille mittari, joka ei ole yhtä hienojakoinen, vaan keskittyy enemmän tarjoamaan tuoteomistajalle tärkeää dataa. Tiimin työjono koostuu tuoteomistajan järjestämästä työlistasta, tämä taas koostuu tuoteorganisaation luomasta järjestetystä ominaisuuslistasta. Tuoteomistaja saa tiimitasolta tietoonsa oman tiiminsä kehitysnopeuden. Koska tekninen arkkitehtuuri yrityksessä on toteutettu siten, että tiimien välillä on riippuvuuksia, kokonaisuuden valmistuminen on usein kiinni yli yhden tiimin tuottamasta arvosta. Tämä tieto kasataan tuoteomistajien yhteistyönä, jolloin tiedossa on jokaisen tiimin käyttämä työmäärä, joka ominaisuuden valmistumiseen tarvittiin. Tätä työmäärää vasten tuoteomistaja pystyy yhteistyössä muiden samaa työtä tekevien henkilöiden kanssa kasamaan riittävän määrän tietoa ennustaakseen tulevia ominaisuuksia.

Tuoteomistajatason jälkeen tarvitsee jälleen ottaa yksi askel kauemmas hienojakoisuudesta. Tällöin voimme ennustaa hankkeiden aikaa ja ennustaa sitä, kuinka kauan hankkeen valmistumisessa kuluu aikaa. Tämä edellyttää sitä, että on olemassa tiekartta ja ymmärrys hankkeiden vaatimista ominaisuuksista. Tällöin strateginen tiekartta ja sen ennustettavuus paranee, mitattava arvo on näkyvää ja konkreettista.

On kuitenkin tärkeää huomioida ennustettavuuden tarkkuus. Tiimi, joka työskentelee kahden viikon sprinteissä ennustaa kokemuksesta ja odottamattomista sattumista johtuen tekemisensä epätarkasti. Scrum ohjaa tiimiä tavoittelemaan maksimaalista suoritusta, jolloin normaalijakaumalle laitettuna puolet sprinteistä epäonnistuu. Siksi tiimitasolla tehtävä työjonon asioiden pilkkominen on oltava huolellista. Kerätty data ja sen vaihteluväli pysyy riittävän tarkalla tasolla, vaikka epäonnistumisia tulee. Mitä kokeneempi ja parempi tiimi on pilkkomaan työjonon tehtäviä sitä, tarkempaa dataa se pystyy tuottamaan. Seuraava askel ottaa jo huomioon useamman tiimin ennustukset. Tämä tarkoittaa virheen kertautumista aina tasolta toiselle siirryttäessä, jolloin jo pieni virhe kertautuu. Sama tapahtuu seuraavalla tasolla, joten ennustettavuudessa huomioitava virhe pitää muistaa sovittaa tasoon, jolla toimitaan.

## 5.2 Läpinäkyvyyden lisääminen

Seuraavaksi käsiteltäväksi asiaksi otettiin läpinäkyvyyden lisääminen. Ennustettavuuden parantamisella voidaan katsoa olevan läpinäkyvyyttä lisäävä vaikutus. Tämä kuitenkin on vain osittain totta, sillä ilman ennustettavuuden seuraamista, kommunikointia ja varmistusta, ei voida sanoa ennustettavuuden parantavan läpinäkyvyyttä. Viitekehyksen täytyy siis ottaa huomioon myös tapa, jolla tasosta riippumatta tulokset ovat kaikkien saatavilla ja ne ovat ymmärrettäviä. Tätä varten viitekehyksen pitää tarjota jonkinlainen julkinen paikka, jossa tätä edistystä voidaan seurata ja siitä voidaan tiedottaa.

Vaihtoehtoina esitettiin Jira-näkymää, jossa jokainen tiimi, hanke ja kokonaisuus olisi nähtävillä yhdessä näkymässä. Tämän tekninen toteuttaminen ottaa hieman aikaa, mutta ei ole kuitenkaan mahdoton toteuttaa nykyaikaisilla järjestelmillä ja työkaluilla. Toinen vaihtoehto oli toistuvat palaverit, joissa jokainen osallinen kertoisi nykytilanteensa ja ennustuksen. Toistuvan palaverin tiheys täytyy sovittaa kehitysrytmiin ja tarkasteltavaan tasoon sopivaksi. Riskinä on, että viitekehys tuo turhia aikaa vieviä palavereja, jotka eivät edistä työntekoa.

Lopputulena kuitenkin päädyimme ottamaan käyttöön molemmat keinot. Jira-näkymää käytettäisiin päivittäisessä tai muuten tiheämpää tarkastelua vaativassa tilanteessa. Lisäksi tuoteomistajien ja sidosryhmien tiedotusta varten järjestetään tarpeen vaatiessa tilaisuus, jossa edistymistä käydään läpi. Täten koollekutsumisen tiheys voi vaihdella, jos tilanne joskus vaatii intensiivisempää tarkastelua.(8)

### 5.3 Päätösten vaikutusten tarkasteleminen

Tämä viimeinen kohta vaatii onnistuakseen oikein rakennetun mittariston, sekä pitkäaikaisen mittausjakson dataa. Vasta silloin, kun läpinäkyvyys ja ennustettavuus ovat kunnossa, voidaan tätä mittaria hyödyntää. Lähtökohtana on se, että jokainen tehty päätös muuttaa lyhyen ja pitkän aikavälin tuloksia. Esimerkiksi päätös korjata ohjelmiston teknistä velkaa on osattava asettaa oikeaan kohtaan. Jos mittamme tiimien valmiiksi saatuja tehtäviä, silloin myös tehtäviin pitää tarkemmin kirjata onko kyseessä esimerkiksi ylläpitotyö, laadunparannus tai uuden kehittäminen. Tämä aiheuttaa jälleen yhden askeleen lisää, joka tarvitsee täyttää tehtävienhallintaohjelmistossa.

Tämä osa-alue päätettiin tässä vaiheessa kirjata muistiin, mutta koska kyseessä on kuitenkin koko teknologiaorganisaation läpi kuljetettava projekti, tämän mittarin käyttöönottoa päätettiin lykätä siihen saakka, kunnes organisaatio on saavuttanut riittävän hyvän tason kahden edellisen asian suhteen. Tämä päätös tehtiin myös siksi, että siihen liittyy myös työaika. Tämä viitekehys ei kuitenkaan halua ottaa kantaa käytettyyn aikaan, sillä ajan mittaaminen työmäärien kanssa on epäpätevä mittari. Eri ihmisiltä kuluu eri aika tehtävien suorittamiseen, jolloin aika-arvion antaminen tarkoittaisi myös sitä, että tehtävät olisi jaettu henkilöille. Kuitenkin Scrum ja Kanban eivät tätä toimintaa tue. Kanbanissa työ otetaan työjonon kärjestä, jolloin tekijä ei voi, ilman tarkoituksen mukaista viivästelyä tai kiirehtimistä, valita seuraavaa tehtävää. Scrum taasen pitää pienimpänä mahdollisena yksikkönä tiimiä. Siksi ennustukset sprintin alussa tulisi tehdä siten, että vaikka joku tiimin jäsenistä olisi sairaana kokonaisen sprintin ja tehtävä olisi hänen ydinosaamisaluettaan, tehtävä saadaan suoritettua.

Siksi vaikutusten tarkasteleminen päätettiin jättää myöhemmälle ajalle. Lisäksi datan kerääminen, jota jotkin tiimit olivat jo ansiokkaasti tehneet pidemmän aikaa, ei ollut kuitenkaan koko teknologiaorganisaation toimintamalli. Siksi tiedon määrä ja tarkkuus vaihteli suuresti tiimien välillä. Vasta kun dataa on tarpeeksi ja sen tarkkuus on hyväksyttävällä tasolla, voidaan tätä mittaria tarkastella ja ottaa käyttöön.

## 5.4 Yhteenveto

Organisaation nykyisen tilan ja toimintamallien huomioonottaminen oli tärkein ohjaava tekijä, kun viitekehystä suunniteltiin. Siksi lopputuloksena päädyttiin antamaan tiimeille edelleenkin vapaus valita miten he järjestävät työnsä. Kuitenkin niin, että koko organisaatio noudattaa samoja ydinperiaatteita ja käyttää samaa termistöä ja ymmärtää mistä puhutaan.

Osastotasolla ja tuoteorganisaatiossa otettiin käyttöön viitekehys, joka pyrkii hyödyntämään organisaatiossa jo olemassa olevia käytäntöjä. Kaikki osastot siirtyvät käyttämään kahdentoista viikon kehityssykliä, joka alkaa ja päättyy kailta samanaikaisesti. Tällöin on mahdollista kartoittaa riippuvuudet, sopia mahdollisista osastojen välisistä yhteishankkeista ja tarkistaa suunta, jotta koko teknologiaorganisaatio etenee yhdessä sovittuun suuntaan.

Tuoteomistajille luodaan rakenne, joka tukee heidän ammatillista kehittymistänsä, sekä tarjoaa mahdollisuuden keskustella yleisesti teemoista, joita organisaation sisällä kehitystyössä ilmenee. Tuoteomistajille järjestetään myös mahdollisuus kouluttautua Scrum-sertifioituiksi tuoteomistajiksi, jolloin vahvistetaan yhteistä osaamis pohjaa ja käytettyä sanastoa.

Kehitystiimit, jotka käyttävät Scrumia koulutetaan virallisen kouluttajan puolesta viitekehyksen käyttäjiksi, jolloin toimintatavat jälleen kerran siirtyvät hieman yhdenmukaisempaan suuntaan. Scrum Mastereille tarjotaan sama mahdollisuus kouluttautua sertifioituiksi Scrum Mastereiksi työn sujuvuuden parantamiseksi. Tiimit sopivat myös yhteisistä käytännöistä ja tiedottavat toisiaan hyväksi

havaituista toimintamalleista, muistaen kuitenkin, että jokaisella tiimillä on vapaus valita toimintatapansa parhaaksi katsomallaan tavalla. Tiimeiltä vaaditaan, että he toimittavat aiemmin mainitusti työn edistymistä kuvaavan datan organisaation käyttöön, jolloin mittaristo on mahdollista luoda.

Tämä kokonaisuus mahdollistaa tehokkaan organisaation läpi kulkevan toimintamallin, joka tukee ennustettavuudellaan strategista työtä korkealla tasolla, auttaa tuoteomistajia ja –organisaatiota pysymään tilanteen tasalla riippumatta kehityksen rytmistä tai ajanhetkestä ja tarjoaa tiimeille mahdollisimman suuren autonomian valita parhaat mahdolliset työskentelytavat tukemaan päivittäistä työtä.

Viitekehys on riittävän yksinkertainen, että sitä pystytään tarpeen vaatiessa muokkaamaan nopeasti, eikä se luo ylimääräisiä rakenteita turvottamaan organisaatiota. Jokaisella tasolla voidaan tehdä tarkennuksia, suunnanmuutoksia tai muita liikkeitä ilman, että kokonaisuus kärsii. Tässä vaiheessa kuitenkin pitää muistaa, että vaikka asia teoreettisella tasolla toimii, on otettava huomioon myös reaali maailma.

Kommunikaatio ja tiedonkulku ryhmässä, jossa on yli kymmenen jäsentä, muodostuu haasteeksi. Silloin muodostuu tiedonkulkuun välittäjä, joka toimittaa tiedon seuraavalle ryhmälle. Pelkästään tämän vuoksi organisaatio, jossa on 350 henkeä, joutuu suuriin haasteisiin viestin välittymisen vuoksi. Lisäksi vuonna 2020 alkanut pandemia on muokannut työelämää täysin poikkeuksellisella tavalla, jossa työn maantieteellinen sijainti, joka ei tietotekniikan alalla ole aikaisemminkaan ollut merkittävä tekijä, vähenee entisestään. Vaikka on osoitettu, että samassa fyysisessä paikassa olevat ihmiset kommunikoivat suuremmalla todennäköisyydellä keskenään, kuin internetin välillä olevat etätiimit. (3)

## 6 Viitekehityksen sovittaminen vallitsevaan toimintaympäristöön

Viitekehityksen suunnittelun jälkeen seuraavana vuorossa oli viitekehityksen sovittaminen ja jalkauttaminen yrityksessä vallitsevaan toimintaympäristöön. Lähtökohtana käytettiin aikaisempaa kartoitusta organisaatiosta ja sen jälkeen suunniteltua viitekehityskokonaisuutta. Ensimmäinen askel oli esitellä osastojen johtajille uusi malli. Tähän tapahtumaan kutsuttiin mukaan kaikki teknologiaorganisaation johtajat, tuoteorganisaation johtajat ja päivittäistä operatiivista toimintaa hallinnoivat Release Managerit. Roolijako viitekehityksen jalkauttamisessa luotiin tässä vaiheessa. Release Managerit, jotka ovat osastoittain vastuussa siitä, miten ja millä tiheydellä ohjelmistoa julkaistaan, sekä hallinnoivat kehityksen teknistä kokonaisuutta saivat vastuulleen mallin kommunikoimisen teknologiaorganisaation tiimeille.

Tätä tukemaan tarvittiin sisäistä osaamista ja organisaatio päätyikin avaamaan muutosta tukevan roolin yksi sisäinen ja yksi ulkoinen työpaikka. Nimikkeenä toimi ketterä valmentaja, joka on myös kirjoittajan työnkuva. Tämä ratkaisu nosti organisaation tukitoiminnot uudelle tasolle. Roolijako jaettiin siten, että yksi valmentajista keskittyy tiimitason valmentamiseen ja kouluttamiseen, yksi organisaation ketteryyssymmärryksen parantamiseen ja strategisten päätösten läpiviintiin, mittaristoon ja tiekarttojen parantamiseen. Kolmas rooli oli tukea osastojen johtajia viestin jalkauttamisessa ja uusien mahdollisten lähestymistapojen arvioinnissa.

Nämä kolme valmentajaa muodostavat yhdessä Release Managerien kanssa ydinryhmän, jonka vastuulla tämän muutoksen kommunikointi ja läpivieminen on. Seuraavaksi käsittelemme valmentajien eri roolit, sekä Release Managerien tehtävät tarkemmin. Lopuksi esittelemme konkretiaa, miten asiat etenivät organisaatiossa ja minkälaisia tuloksia saatiin.

## 6.1 Tiimitason valmennus ja koulutus

Tiimitason kartoituksessa paljastui, että vaikka tiimit olivat ottaneet käyttöönsä suurimmilta osin Scrumin tai Kanbanin, ei heille kuitenkaan ollut järjestetty riittävästi perehdytystä tai koulutusta asian suhteen. Tämä oli jättänyt osaamistyhjiön, jonka täyttäminen piti aloittaa. Koska koulutettavana oli tiimitasolla tuoteomistajia, Scrum Mastereita ja tiimejä, työmäärä oli liian suuri yhdelle ihmiselle varsinkin lyhyessä aikaikkunassa.

Tuoteomistajaorganisaatio antoikin valmentajalle tukensa ulkoisen avun hankkimiseen. Tämä tarkoitti käytännössä sitä, että tuoteomistajien koulutus kilpailutettiin ja sitä kautta ostettiin ulkoiselta toimijalta. Tämän hyödyt organisaatiolle olivat muitakin, kuin pelkästään työmäärän keveneminen. Tuoteomistajat saavat koulutuksestaan virallisen sertifiointitodistuksen, joka parantaa kokonaisuutta ja antaa mahdollisuuden yhtenäistää termistöä. Lisäksi koulutus tarjoaa yritykseen liittymättömiä hyötyjä. Ulkoisen toimijan koulutuksissa on läsnä myös muiden yritysten ja toimialojen ihmisiä, jolloin oppia kertyi muustakin kuin pelkästään oman organisaation toiminnasta ja viitekehuksesta. Lisäksi tuoteomistajat järjestäytyivät sisäisesti osastojen yli luoden mahdollisuuden ratkaista ongelmia ja keskustella esiin nousevista asioista tarpeen vaatiessa. Tästä ryhmästä, jota kutsutaan yrityksessä PO-heimoksi, vastasi yksi vanhempi tuoteomistaja, keventäen myös valmentajan työtaakkaa tällä alueella.

Scrum-koulutuksen kanssa toimittiin samankaltaisella tavalla. Tässä kuitenkin hyödynnettiin yhtä valmentajaa ja tämän kykyä kouluttaa Scrum Mastereita. Halukkaille järjestettiin koulutus, jonka jälkeen yritykseen palkattavat ihmiset voivat halutessaan osallistua ulkoisen kouluttajan tarjoamalle sertifiointikurssille, jolloin keskitetysti järjestettävän kurssin ajankohta ei ole esteenä koulutuksen saamiselle tai aiheuta tarpeettoman pitkää odotusaikaa uusille työntekijöille. Ulkoisen koulutuksen hyödyt ovat samat kuin edellisessä kappaleessa esitetystä tuoteomistajan koulutuksessa ja sen painoarvoa punnittiin tarkasti, mutta kokonaiskustannuksellisesti tämä ratkaisu oli järkevin.

Tiimitason koulutus jätettiin täysin sisäisen valmentajan työksi. Tämä lähtökohteisesti johtui siitä, että organisaation tiimit olivat hyvin heterogeeninen ryhmä ja myöskin koulutusten ajankohdan sopiminen on haastavaa ulkoisen toimijan kanssa. Valmentajan osallistuminen tiimin päivittäiseen työhön ja sitä kautta koulutuksen tai valmentamisen tarjoaminen suoraan on huomattavasti tehokkaampi ja täsmällisempi tapa toimia, kuin että koko tiimi lähetettäisiin koulutukseen tiettyinä ajanhetkenä. Samoin tiimi- ja organisaatiokohtaiset ongelmat voidaan käsitellä sisäisen valmentajan kanssa helpommin, koska asiaan ei liity salassapitosopimuksen rikkomista työntekijöiden ollessa saman yhtiön palveluksessa.

## 6.2 Organisaation ketteryys ja strategisten päätösten läpivienti

Jotta tiimitason työskentely on sujuvaa, pitää organisaation olla myös muilta osin ketterä. Muuten pullonkaula muodostuu vain toiseen paikkaan. Tästä syystä yksi valmentaja laitettiin työskentelemään organisaation ketteryyden ja strategisten päätösten kanssa. Organisaatio, jolla on tuote, joutuu tasapainottelemaan monen asian kanssa, mutta ketteryyden puolesta tässä mainitaan kaksi. Ensimmäinen on nykyisen ohjelmiston ylläpitämisen ja mahdollisen bugikorjausten tuoma tekeminen. Toinen on uusien ominaisuuksien tuominen. Pysyäkseen kilpailukykyisenä, uusia ominaisuuksia on tuotettava tasaiseen tahtiin ja niiden on vastattava markkinan kysyntään. Tällä varmistetaan, että tuote ja yritys pysyy jatkossakin kilpailukykyisenä. Pelkkä uuden kehittäminen ei riitä, sillä huonosti toimivat ominaisuudet tai rikkinäinen tuote on korjattava. Siksi tämä vaatii myös kehitystiimien huomiota.

Tasapainoilu näiden kahden välillä ja oikea-aikainen resurssien keskittäminen kulloinkin järkevään kohteeseen mahdollistaa yrityksen kilpailukykyyn säilymisen. Kumpikaan ääripää ei ole kestävä, edes hetkellisesti. Siksi toinen valmentaja keskittyy osastojen johtajien valmentamiseen tässä ajattelussa. Näiden tunnistaminen ja hyödyntäminen on tärkeässä osassa ketteryyttä.

Strategisten päätösten läpivienti on valmennuksen seuraava osa. Yritys rakentaa strategisen tiekartan ja pyrkii mahdollisimman tehokkaasti kuvaamaan tulevat haasteet ja tarvittavat ominaisuudet, joita kehitystiimit ottavat tehtäväkseen. Työjonon luominen tiekartasta on kuitenkin vaikeampaa, kuin moni kuvittelee. Suurin ongelma on siinä, että työkartan tarkkuus on joko liian tarkka tai epätarkka. Usein kuvattava kokonaisuus muuttuu vaatimuslistaksi tiimeille, jolloin asiantuntijoiden potentiaali jää käyttämättä. Liian yksityiskohtainen tiekartta tukahduttaa luovuuden ja poistaa mahdollisuuden tehdä ketterästi tiimitason päätöksiä. Vastakohtana on epätarkka tiekartta, josta puuttuu konkretia. Korkealla tasolla tehtävät epämääräiset kokonaisuudet eivät anna tarpeeksi tietoa toteuttavalle organisaatiolle, mitä pitäisi tehdä. Tämä versio tiekartasta aiheuttaa epätietoisuutta tavoitteista ja siitä, mitä seuraavaksi pitäisi lähteä tekemään.

Siksi suurin huomio tässä vaiheessa ohjattiin tiekartan rakentamiseen ja siitä työstettävän työjonon luomiseen. Strategisten päätösten on oltava riittävän korkealla tasolla, jotta ne eivät ole pelkkä vaatimuslista ominaisuuksista, mutta sisältävät kuitenkin riittävästi konkretiaa välittämään johdon tahtotilan tiimeille. Vasta silloin tullaan tiimitasolla siihen pisteeseen, että strategisesta tiekartasta voidaan luoda työjono, joka vastaa yrityksen tarpeita. Tämän vaiheen jälkeen tiimitason ja strategisen tason valmennus ovat saavuttaneet sen pisteen, jossa konkreettiset hyödyt alkavat näkymään.

### 6.3 Johdon tuki ja uusien lähestymistapojen arviointi

Kolmannen valmentajan tehtäväksi jäi johdon tukeminen ja ohjaaminen ajattelemaan oikeita asioita, sekä auttamaan uusien lähestymistapojen arvioinnissa. Organisaation kasvu ja rakenteen muuttuminen loi uuden ulottuvuuden siihen, mitä lähestymistapoja voitiin hyödyntää, joten niiden arviointi oli tärkeässä osassa. Pelkästään uusien hierarkiatasojen luominen organisaatioon luo aina uuden tason kompleksisuuteen. Tämä lisää kommunikaatiokuormaa ja mahdollisia väärinkäsityksiä. Lisäksi johdon etäännyminen päivittäisestä tekemisestä muodostuu ongelmaksi. Tätä ongelmaa ratkaisemaan on nykyisessä teknologia maailmassa monia keinoja, mutta koska viitekehys tarjosi mahdollisuuden

lisätä läpinäkyvyyttä ja ennustettavuutta, joka oli myös johdon toive, pyrittiin yhteisymmärryksessä etsimään parhaat mahdolliset keinot, jotka hyödyntävät kerättyä dataa ilman uusia raskaita rakenteita.

Tarkoituksena oli rakentaa oppiva organisaatio. Tästä syystä uusia lähestymistapoja on oltava jatkuvasti arvioimassa. Koska johdon ydinosaamisalue ei ole ketterissä viitekehyksissä, valmentajan rooli tässä on tarjota heille oma arvionsa lähestymistavan vahvuuksista ja heikkouksista. Tämän lisätiedon valossa johdolla on käsissään paremmat mahdollisuudet tehdä tietoon perustuvia päätöksiä ja hyötyä organisaation jo oppimista hyvistä tavoista toimia, sekä välttää toimimattomiksi todetut tavat.

#### 6.4 Käytännön toteutus

Release Managerien tehtäväksi jäi jalkauttaa malli kehitystiimeille. Koulutus ja valmentajien tuki olivat toki osana kokonaisuutta, mutta ilman tätä roolia ja asian konkretisoimista päivittäisessä työssä olisi tämä muutos jäänyt toteutumatta. Ensimmäinen askel oli aloittaa yhteinen rytmi. Tämä toteutettiin luomalla yhteinen aikaikkuna seuraavan kolmen kuukauden työn suunnittelulle. Tässä vaiheessa käyttöön otettiin kahdessa osastossa jo käytössä ollut Release Train –malli. Termi on SAFe:n ja yhteinen suunnittelu päätettiin nimetä saman viitekehyksen tapahtumalla inkrementtisuunnittelu. Asiasta käytiin debattia, mutta lopputulemana oli, että täysin oman sanaston luominen vaikeuttaa uusien työntekijöiden mukaantuloa, sekä aiheuttaa sekaannusta.

Ketterään ajatusmaailmaan kuuluu palautteen kerääminen, oppiminen ja toiminnan säätäminen.(3) Ensimmäisen yhteisen suunnittelun jälkeen jokainen Release Manager järjesti oman osastonsa kanssa retrospektiivin, jossa käytiin läpi onnistumiset ja epäonnistumiset inkrementtisuunnittelussa. Tämä palaute kerättiin yhteen ja analysoitiin valmentajien sekä Release Managerien kesken. Analyysin valmistuttua suoritettiin asian vaatimat korjaustoimenpiteet, joiden toimivuus tarkastellaan jälleen seuraavan suunnittelun jälkeen.

Went well	To improve	Action points
Thing 1	Thing 1	Action 1. Responsible person: Jane Doe
Thing 2	Thing 2	Action 2. Responsible person: John Doe
Thing 3		Action 3 Responsible person: unassigned

Kuva 4 Retrospektiivitaulesimerkki

## 7 Havainnot

Malli luotiin johdon havaintojen ja toiveiden perusteella. Tämän takia johdon tuki oli viitekehityksen luomiselle ja sen jalkauttamiselle alusta alkaen vahvaa. Suurin haaste johdon suunnalta oli löytää yhteistä aikaa käydä asioita tarpeellisella tarkkuudella läpi.

Tuoteomistajien motivointi oli helppoa, mutta koska malli poikkesi heidän aikaisemmasta toimintatavastaan ja siinä oli paljon uusia asioita, vaikkakin yksinkertaisia, tarjotun tuen määrä oli alkuun aivan riittämätön ja ohjeistus puutteellinen, että onnistumisia olisi saavutettu. Malli jalkautettiin kuitenkin askel kerrallaan, eikä mikään tapahtuma tullut yllätyksenä. Kuitenkin tilanne oli liian monelle

ihmiselle epäselvä. Tähän lisäksi vielä kasvuyrityksen valtava rekrytointitahti, toimialan monimutkaisuus ja yleinen tietotekniikan alan valtava muutosvauhti toivat suuren määrän haasteita. Muutosjohtamisen kannalta itseohjautuvuuteen luottaminen otti liian suuren osan jalkautuksesta, joten varsinaiselle konkreettiselle ohjaamiselle ei löytynyt sopivaa aikaa. Tämä olisi voitu ratkaista kouluttamalla paremmin tai lisäämällä resursseja valmennukseen.

Tiimitasolla muutosvastarinta oli kaikista suurinta, pitkälti tiimien mukavuudenhalun takia. Uusi malli vaati paljon kurinalaisempaa työskentelyä, kuin aikaisemmin on vaadittu, joten vakiintuneet toimintatavat vaativat täydellistä ravistelua. Lisäksi oman toimintatavan muuttaminen toiseksi, sekä alalla vallitseva vastenmielisyys tiettyjä termejä, metodologioita ja viitekehyksiä vastaan vaikeutti tehtävää.

## 8 Loppupäätelmät

Vaikka ketterän viitekehyksen luominen on yksinkertainen tehtävä, sen jalkauttaminen on vaikeaa. Asian ei tarvitse olla monimutkainen. Ennakkoluulot tiettyjä viitekehyksiä kohtaan ovat vahvat, myös kirjoittajalla, täytyy muistaa, että jokainen askel kohti ketteryyttä on hyödyllinen modernille ohjelmistoalan yritykselle. Tästä syystä kaikki pienetkin toimivat yksityiskohdat kannattaa validoida ja käydä tarkasti läpi. Itseohjautuvuus ja itsemääräämisoikeus antavat parhaimmillaan tiimeille suuren itseluottamuksen, jonka pohjalta syntyy loistavia ratkaisuja. Kuitenkin kontrollin ja kurinalaisuuden puute aiheuttaa usein harhailua pois olennaisen ääreltä.

Ketterien menetelmien käytössä harhana usein on, että työnteosta poistuvat ne ikävät asiat. Näin monesti onkin, mutta kun kaikki eivät poistukaan, leimataan se usein virheellisesti viitekehyksen syyksi, vaikka kyseessä on puhtaasti työelämän lainalaisuudet. Jotkin asiat nyt vain on tehtävä. Tähän lisättynä vapaa ja luottava kulttuuri, epäkypsät tiimit, sekä oman osaamisen yliarviointi aiheuttavat usein isoja ongelmia. Niin uusien toimintatapojen tuojalle kuin myös uusille tiimin jäsenille, jotka haluaisivat parantaa olemassa olevia käytäntöjä. Jokainen

ihminen, jonka kanssa tästä asiasta yrityksessä keskusteltiin toi esille halunsa kehittyä ja muuttaa toimimattomia käytäntöjä. Kuitenkin siinä kohdin, kun viitekehyksen jalkautus tuli ajankohtaiseksi nousi vastarinta erittäin suureksi.

.

## Lähteet

1. Manifesto for Agile Software Development [Internet]. [viitattu 22. elokuuta 2022]. Saatavissa: <https://agilemanifesto.org/>
2. Scrum Guide | Scrum Guides [Internet]. [viitattu 20. elokuuta 2022]. Saatavissa: <https://scrumguides.org/scrum-guide.html>
3. Jeff, Sutherland, James O, Coplien, Lachlan, Heasman, Mark, den Hollander, Cesário Oliveira, Ramos. A Scrum Book: The Spirit of the Game. 1. p. The Pragmatic Bookshelf;
4. 5 Lean Principles Every Engineer Should Know [Internet]. [viitattu 20. elokuuta 2022]. Saatavissa: <https://www.asme.org/topics-resources/content/5-Lean-Principles-Every-Should-Know>
5. SAFe 5.0 Framework [Internet]. Scaled Agile Framework. [viitattu 20. elokuuta 2022]. Saatavissa: <https://www.scaledagileframework.com/>
6. Introduction to LeSS [Internet]. Large Scale Scrum (LeSS). [viitattu 22. elokuuta 2022]. Saatavissa: <https://less.works/less/framework/introduction>
7. Atlassian. Kanban - A brief introduction [Internet]. Atlassian. [viitattu 20. elokuuta 2022]. Saatavissa: <https://www.atlassian.com/agile/kanban>
8. Manage dashboards and gadgets | Jira Service Management Cloud [Internet]. Atlassian Support. [viitattu 22. elokuuta 2022]. Saatavissa: <https://support.atlassian.com/jira-service-management-cloud/docs/manage-dashboards-and-gadgets/>