



Tiitus Telke

SAFe-toteutuksen kehittäminen palauteiden perusteella

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikan tutkinto-ohjelma

Insinöörityö

11.11.2024

Tiivistelmä

Tekijä: Tiitus Telke
Otsikko: SAFe-toteutuksen kehittäminen palautteiden perusteella
Sivumäärä: 44 sivua + 1 liite
Aika: 11.11.2024

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Mobile Solutions
Ohjaajat: Peter Hjort, Lehtori
Emil Salmi, Release Train Engineer

Insinöörityössä tutkittiin Scaled Agile Frameworkin (SAFe) Tarkasta ja mukauta -tapahtuman aikana kerättyä palautemateriaalia käyttäen apuna kirjallisuutta ja Capability Maturity Model Integration (CMMI) -mallia. Tutkimuksen pohjalta oli tarkoitus koota palautteet aihealueittain ja luoda ratkaisuehdotuksia toimeksiantajalle. Lisäksi tutkittiin CMMI-mallin hyödyntämistä palautteiden analysoinnin apukeinona.

Työn aikana tutkittiin SAFe-viitekehystä ja CMMI-mallia käyttäen kirjallisuutta ja verkkomateriaalia. Palautemateriaali jaettiin eri aihealueisiin. Aihealueiden tärkeyden määrittelyssä käytettiin apuna CMMI:n prosessialueita ja niiden kypsyysluokituksia. Kypsyysalueiden avulla määriteltiin, kuinka tärkeitä palautteet olivat ohjelmistotuotannolle.

Palautteet saatiin jaettua kuuteen eri aihealueeseen, ja nämä aihealueet liitettiin CMMI-prosessialueisiin. Lopuksi luotiin ratkaisuehdotuksia havaittuihin ongelmiin. Työtä tehdessä havaittiin, että CMMI-mallista on mahdollista saada apua palautteiden luokitteluun ja parannusehdotusten laatimiseen.

Tulokset ovat hyödynnettävissä prosessinkehittämisessä ja SAFe-viitekehysten toteutuksessa havaittujen ongelmien parantamisessa.

Avainsanat: ketterä kehitys, prosessinkehittäminen, Scaled Agile Framework, Capability Maturity Model Integration

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Tiitus Telke
Title: Developing a SAFe implementation based on feedback
Number of Pages: 44 pages + 1 appendix
Date: 11 November 2024

Degree: Bachelor of Engineering
Degree Programme: Information and Communications Technology
Professional Major: Mobile Solutions
Supervisors: Peter Hjort, Senior Lecturer
Emil Salmi, Release Train Engineer

In this project feedback that was collected during Inspect and Adapt event of the Scaled Agile Framework's (SAFe) was studied with the help of literature and the Capability Maturity Model Integration (CMMI). After studying, this feedback was grouped by topic and proposals for solutions for the problems highlighted in the feedback were made.

During the project the SAFe framework and the CMMI model were studied by using literature and online material. The priority of the feedback topics was defined by using CMMI assigning them to CMMI process areas and then the maturity level of the process areas was used to determine how crucial a piece of feedback was for smooth software production.

As results the feedback was grouped under six different topics and these topics were assigned to CMMI process areas. Finally, solution proposals were made. During the work it was observed that the CMMI model can be helpful for prioritizing feedback and coming up with solution proposals.

The understanding gained in this project can be used for process improvement and creating solutions for problems in an organization's SAFe implementation.

Keywords: agile development, process improvement, Scaled Agile Framework, Capability Maturity Model Integration

The originality of this thesis has been checked using Turnitin Originality Check service.

Sisällys

Lyhenteet ja SAFe-sanasto

1	Johdanto	1
2	SAFe	1
2.1	Tärkeimmät periaatteet ja arvot	2
2.2	SAFen käytännöt	5
2.3	SAFen roolit	15
2.4	Hyödyt	21
2.5	Haasteet	22
3	CMMI	23
3.1	CMMI ja sen soveltaminen	23
3.2	Vaiheistetun ja jatkuvan kehittämisen mallit	24
3.3	Prosessialueet	26
3.4	CMMI ketterässä kehityksessä	27
3.5	Prosessien arviointi	28
4	Menetelmät	31
5	Tulokset	32
5.1	Havainnot	32
5.2	Ratkaisuehdotukset	33
6	Tuloksien arviointi ja pohdinta	37
	Lähteet	40
	Liitteet	
	Liite 1: Prosessialueiden kyvykkyyssluokitukset	

Lyhenteet ja SAFe-sanasto

ART:n suunnittelutaulu:

ART Planning Board. Visualisointi PI:den julkaisupäivistä, tiimien välisistä riippuvuuksista ja muista merkittävistä virstanpylväistä.

CI/CD: *Continuous Integration / Continuous Deployment*. Kokoelma käytäntöjä, joiden tarkoitus on sujuvoittaa sovellusten rakentamista, testausta ja julkaisua käyttämällä jatkuvaa automatisointia.

CMMI: *Capability Maturity Model Integrated*. Prosessinkehittämiseen suunniteltu malli, joka sisältää tehokkaan prosessin ominaisuuksia.

Epic: Suuri työkokonaisuus, jota ei ole mahdollista suorittaa yhden suunnitteluvälin aikana. Tämä pilkotaan ominaisuuksiin, joista kunkin ominaisuuden pystyy suorittamaan yhden suunnitteluvälin aikana.

Innovaatio- ja suunnittelutoistokierros:

Innovation and Planning Iteration, IP iteration on toistokierros SAFe-viitekehyksessä, jonka aikana on tarkoitus keksiä uutta, suunnitella ja oppia. IP-toistokierros toimii lisäksi suunnitelman puskurina.

Julkaisujunan päällikkö:

Release Train Engineer, RTE. Ketterän julkaisujunan johtamisesta vastaava SAFen rooli.

Järjestelmädemo:

System Demo. SAFen tapahtuma, jonka aikana esitetään käytännön näyte työn etenemisestä ja kerätään palautetta.

Kehittämisen arvovirrat:

Development Value Streams. Sarja tarvittavia toimintoja muuttamaan liikeidea digitaaliseksi ratkaisuksi, joka tarjoaa hyötyä asiakkaalle.

Ketterä julkaisujuna:

Agile Release Train, ART. Noin 50–125 henkilön monitaitoinen yksikkö, johon kuuluu useita tiimejä.

Ketterä tiimi:

Agile Team. Monitaitoinen ryhmä, joka määrittää, rakentaa, testaa ja julkaisee liikearvoa tuottavia tuotteita asiakkaalle.

Lean-portfolion hallinta:

Lean Portfolio Management (LPM) -kyvykkyudessa sovelletaan lean- ja systeemiajattelua strategiaan, investointien rahoitukseen, ketterän portfolion operointiin ja hallintoon.

Liikearvo:

Business Value. Kaikenlaista liiketoiminnalle pitkällä aikavälillä hyödyllistä arvoa. Liikearvo voi olla konkreettista, aineetonta tai kumpaakin.

Liiketoiminnan edustaja:

Business Ownerit eli BO:t edustavat junan päivittäisessä toiminnassa liiketoiminnan ja asiakkaiden etua.

Mahdollistaja:

Enabler. Suunnittelutaulun kohde, jonka tarkoitus on tehostaa nykyistä tuotantoa mahdollistavaa koneistoa tai kehittämisen arvovirtaa.

Portfolion kehitysjono:

Portfolio backlog. Kehitysjono liiketoiminnan edistämiseen suunniteltuja mahdollistajia ja epicejä varten.

Portfolion visio:

Portfolio vision. Esitys, jossa on portfolion tulevia arvovirtoja ja ratkaisuja.

Ratkaisu: *Solution.* Ratkaisu on tuote, järjestelmä tai palvelu, joka tarjoaa lisäarvoa sisäisille tai ulkoisille asiakkaille.

SAFe: *Scaled Agile Framework.* Erilaisia organisaation tehokkuutta edistäviä käytäntöjä sisältävä viitekehys.

SCAMPI: *Standard CMMI Appraisal Method for Process Improvement.* Organisaation CMMI-prosessialueiden tilaa mittaava arviointimenetelmä.

Scrum Master:

Scrum Master, SM / Team Coach, TC. Tiimin palveleva johtaja, jonka tehtäviä on muun muassa varmistaa SAFen prosessien toteutuminen tiimissä ja tukea tiimiä ja koko junaan arvon toimittamisessa.

Sisäänrakennettu laatu:

Built-In Quality. SAFeen sisäänrakennettuja käytäntöjä, jotka auttavat varmistamaan, että kunkin tiimin ratkaisut täyttävät tarkoitukseenmukaiset laatustandardit koko tuotantoprosessissa.

Suunnitteluväli:

PI, planning interval. Aikaikkuna, jonka aikana julkaisujunat tekevät PI-tavoitteiden mukaista hyödyllistä, liikearvoa tarjoavaa työtä asiakkaille.

Tarina:

Story. Lyhyt kuvaus halutusta toiminallisuudesta kerrottuna käyttäjän näkökulmasta.

Tarkasta- ja mukauta:

Inspect and Adapt. SAFen tapahtuma, jossa selvitetään ratkaisun nykyinen tila ja pyritään parantamaan ratkaisun tilaa kehitysehdotusten avulla.

Tiekartta:

Roadmap. SAFen tapahtumien ja virstanpylväiden aikataulu. Aikataululla jaetaan tietoa tiimeille ja ennustetaan tulevia ratkaisun toimituksi

1 Johdanto

Muuttuvat vaatimukset ja jatkuva innovaatio edellyttävät jatkuvaa parannusta. Jos yritykset pysyisivät aina samana, asiakkaiden vaatimuksiin olisi vaikeaa yltää. Siksi monien yritysten toimintaan kuuluu jatkuva prosessien kehittäminen. Prosessien mallintaminen on tunnistettu tärkeäksi työkaluksi kuvailemaan asiakasta hyödyttävät työvaiheet. Prosessiajattelu on osa lean-ajattelua, johon pohjautuu SAFe-viitekehys. Insinööriyön tavoitteena oli kerätä tietoa siitä, mitkä palautteiden aiheet toistuvat usein SAFEn tapahtumien yhteydessä kerätyssä palautteessa ja luoda parannusehdotuksia. Prosessienparannukseen on kehitetty CMMI-malli. Työssä tutkittiin, voiko CMMI-mallia hyödyntää palautteiden analysoinnissa ja ratkaisuehdotusten löytämisessä.

Työn toimeksiantajana toimi CGI Suomi Oy. CGI on kanadalainen monikansallinen yritys, joka on perustettu 1976. Yritys on laajentunut siitä lähtien voimakkaasti, ja Suomeen CGI laajeni Logican oston myötä vuonna 2012. Nykyään yrityksessä on yli 3 800 työntekijää. Tämä tekee CGI:stä Suomen suurimman it-alan työnantajan. CGI:n asiakaskunta Suomessa on laajaa käsittäen muun muassa finanssialaa, julkishallintoa ja erikokoisia yrityksiä. Työ tehtiin OMNI360 -potilastietojärjestelmää kehittäväälle yksikölle. [1.]

Työn idea syntyi tekijän tiimin scrum masterin aloitteesta. Hän oli huomannut, että Confluencessa oli paljon dataa, jota olisi mahdollista hyödyntää työprosessien parantamiseen. Tutkimusmenetelmänä oli empiirinen tutkimus.

2 SAFe

SAFe on kokonaisuus organisaation rakennetta ja työnkulkua ohjaavia käytäntöjä. Käytännöt perustuvat lean-ajattelun ja ketterän kehityksen periaatteisiin [2]. Näillä käytännöillä haetaan parannuksia tuottavuuteen, työtyytyväisyyteen, markkinoilletuontiaikaan ja laatuun.

OMNI360 -kehityksessä käytetty portfoliokokoonpano sisältää budjetoinnin, ideoinnin, salkunjohtamisen ja päätöksenteon sisältävän lean-portfolion hallinnan ja ulottuu kehitystiimin jäsenten arkipäivään saakka. Portfolio tarkoittaa IT-yrityksessä kaikkea aineellista ja aineetonta omaisuutta, jota yritys tarvitsee strategisten tavoitteidensa saavuttamiseen, kuten laitteistoja, verkkoja ja tuotannossa tai suunnitteluvaiheessa olevia projekteja. [3; 4.]

Tässä luvussa esitellään lyhyesti SAFen tärkeimmät periaatteet, käytännöt ja arvot niiltä osin, kun ne ovat tavoitteenmukaisia tähän opinnäytetyöhön. Käännöksissä on käytetty virallista Scaled Agile Inc -organisaation suomenkielistä sanastoa siltä osin kuin sieltä on saatavilla käännöksiä [5].

2.1 Tärkeimmät periaatteet ja arvot

Tärkeimmät SAFen elementit ovat lean-ketterä ajattelu, SAFe-periaatteet ja lean-ketterät johtajat. Lean-ketterä ajattelu käsittää kasvun ajattelutavan (growth mindset) ja lean-ajattelun. Kasvun ajattelutavassa tunnistetaan asenteen merkitys omiin kykyihin ja nähdään epäonnistumiset ja haasteet tilaisuutena kasvaa. Lean-ajattelu on ajattelutapa, jonka tarkoituksena on maksimoida turha työ ja keskittyä tarjoamaan hyötyä eli arvoa asiakkaalle. Ajattelutapa pyrkii virtaviivoittamaan työnkulun ja parantamaan laatua. Lean-ajatteluun kuuluu myös turhan työn minimointi sekä jatkuva prosessien ja resurssienkäytön optimointi. [6.]

Periaatteet

SAFe perustuu kymmeneen periaatteeseen, jotka perustuvat lean-ajatteluun ja ketterään kehitykseen:

- Ota taloudellinen näkemys
- Hyödynnä systeemiajattelua
- Oleta, että töiden vaatimukset vaihtelevat; suunnittele vaihtoehtoisia ratkaisuja

- Rakenna hyödyntäen toistokierroksia, jotka sisältävät oppimiskierroksen
- Perusta välitavoitteet toimivan järjestelmän objektiiviselle arvioinnille
- Luo häiriötön arvovirtaus
- Hyödynnä tasaista rytmiä eri tiimien suunnittelu- ja kehitystyön synkronointiin
- Vapauta tietotyöläisen sisäinen motivaatio
- Hajauta päätöksenteko
- Organisoidu liikearvon ympärille

SAFe-periaatteet painottavat talouden huomioonottamisen jokapäiväisessä työskentelyssä. Tämän lisäksi periaatteet painottavat organisaation osien riippuvuuksien huomioonottamista ja ymmärtämistä suurena kokonaisuutena, jossa kukin osa vaikuttaa toisiinsa eli järjestelmäajattelua ja kehittämistä toistosykleissä liiketoiminnan tavoitteiden mukaisesti. Työ jaetaan arvovirtoihin, jotka työstetään järjestelmässä, johon kuuluu ominaisuuksia kuten keskeneräiset työt, pullonkaulat, vaihdot seuraavalle tekijälle, palaute, työntekijät ja määräykset. Näistä arvovirtaa työstävistä järjestelmistä pyritään poistamaan esteitä kuten liiallinen keskeneräisten töiden määrä ja turhat pullonkaulat. Päätöksenteko suoritetaan epäkeskitetysti, sillä keskitetty päätöksenteko on liian hidasta eikä päätöksentekijöillä voi olla tarpeeksi tietoa päätöksen kohteista. [7.]

Lean-ketterä johtajuus tarkoittaa leanin ja ketterän kehityksen ajattelutapojen ja periaatteiden noudattamista johtajuudessa. Lean-ketterään johtajuuteen kuuluu työn esteiden poistaminen ja työympäristön pitäminen motivoivana. Esteiden poistaminen tapahtuu vähentämällä liiallista yhtäaikaista keskeneräistä työtä eli rajoittamalla WIP (Work in Process) -määrää sekä poistamalla pullonkauloja ja viiveitä. Työympäristö pidetään motivoivana poistamalla epämotivoivia käytäntöjä ja menetelmiä, vaalimalla jatkuvaa kehittämistä ja tilaa oppia sekä inspiroimalla ja motivoimalla muita. [8.]

Arvot

SAFen versiossa 5.0, jota CGI:n OMNI360 -kehityksessä kirjoitushetkellä käytetään, olivat ydinarvot yhteinen suunta, sisäänrakennettu laatu, läpinäkyvyys ja ohjelman suoritus [9]. Uusimmassa 6.0 -versiossa sisäänrakennettu laatu ja ohjelman suoritus on vaihdettu ihmisten kunnioittamiseen (Respect for People) ja sisukkaaseen parannukseen (Relentless Improvement) [10].

Yhteisellä suunnalla tarkoitetaan periaatetta, jossa tiimien suunta ja tietoisuus päämäärästä pidetään yhtenäisenä. Suunnan ei tulisi olla pelkästään tiimien oman käsityksen varassa. Suuntaa tarvitaan välttämään viiveitä ja laatuongelmia. Yhteinen suunta alkaa portfoliotason strategia- ja investointipäätöksillä, jotka heijastuvat strategisiin teemoihin, portfolion visioon, portfolion kehitysjoonoon ja osallistavan budjetoinnin tuloksiin. Kun yhteinen suunta on mukana näissä kaikissa, se heijastuu visioon, tiekarttaan ja kaikkiin kehitysjonoihin. [10.]

Visio, tehtävä ja strategia täytyy jakaa kaikille SAFen tasoille. Yksi tapa saavuttaa tämä on pitää visio, tehtävä ja strategia mukana PI-suunnittelutapahtumissa eli tasaisin väliajoin toistuvissa työsuunnittelutapahtumissa pidettävässä tuotemistajien tiedotustilaisuudessa. Ymmärrys näistä elementeistä täytyy vahvistaa säännöllisesti SAFen tapahtumissa, työkaluissa ja kasvokkaisissa keskusteluissa. [9; 10.]

Läpinäkyvyys tarkoittaa nimensä mukaisesti tiedon jakamista avoimesti. Työympäristöön rakennetaan luottamusta ja läpinäkyvyyttä. Lupauksista pidetään kiinni. Ongelmista keskustellaan suoraan ja peittelemättä. Virheitä ei pelätä tai niitä tehnyttä ei syytetä, vaan niistä tehdään oppimishetkiä. [10.]

Ihmisten kunnioittaminen on suoraan Lean-periaatteista. Scaled Agile Inc. [10] ehdottaa mm. seuraavia tapoja toteuttaa ihmisten kunnioittamista organisaatiossa:

- Tehostetaan yksilöiden luovuutta ja arvostetaan tiimityötä.
- Arvostetaan ihmisten ja heidän mielipiteidensä kirjavuutta.
- Valmennetaan ja mentoroidaan ihmisiä.
- Kohdellaan asiakkaita kunnioittaen ja empaattisesti, sillä he ovat työn lopullisia hyötyjiä.
- Luodaan pysyviä asiakassuhteita, jotka perustuvat molemminpuoliseen hyötyyn.

Sisukas parannus tarkoittaa parantamista oppimalla virheistä ja luomalla painetta kiireellisyyden tunnolla. SAFessa painotetaan säännöllistä pysähtymistä suorittamisesta, jonka aikana keskustellaan virheistä avoimesti. [9.] Ketterät tiimit toteuttavat virheiden tarkastelun tekemällä aina suunnitelman aikaikkunaa kohden, toteuttamalla suunnitelman, tarkistamalla suunnitelman toteutumisen ja lopuksi tekemällä muutoksia, jotta suunnitelma toteutuisi seuraavalla toistokierroksella paremmin [7].

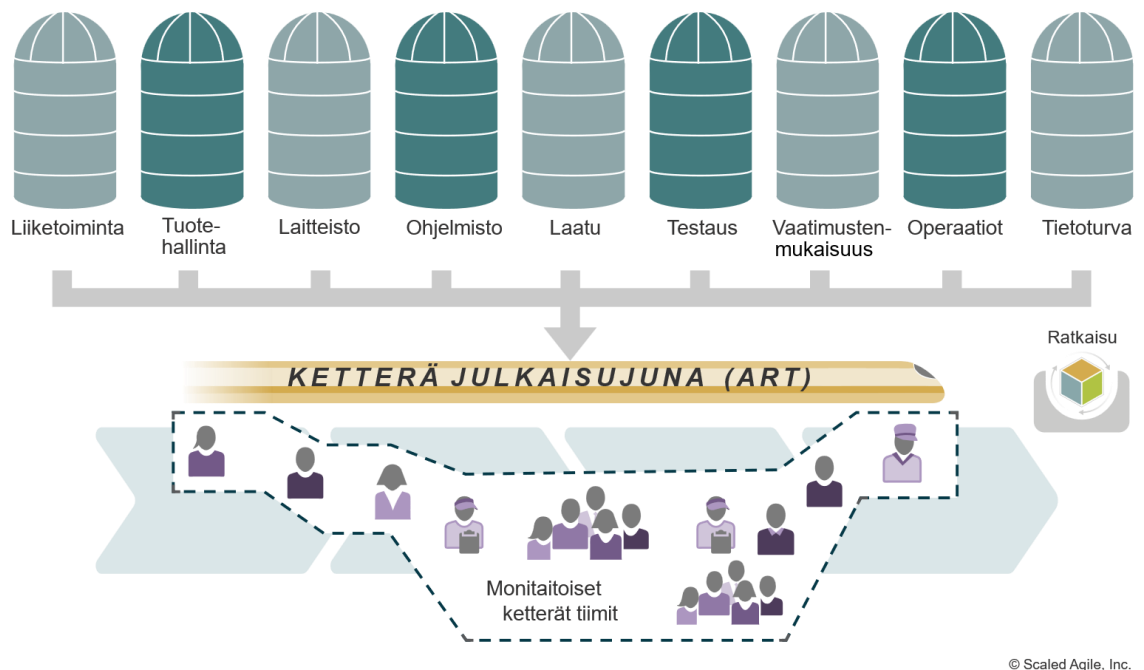
2.2 SAFen käytännöt

Vaikka SAFe 6.0:n periaatteista on poistettu sisäänrakennettu laatu, joka käsiteli muun muassa DevOps-menetelmiä ja ketterää arkkitehtuuria, nämä käytännöt kuuluvat edelleen SAFeen [11]. Ketterään arkkitehtuuriin kuuluu jatkuva integraatio ja jatkuva toimitus, yhteistyö, yksinkertainen suunnittelu sekä etukäteen suunnittelun ja toistokierrosten aikana tapahtuvan suunnittelun tasapainotus. Ketterä arkkitehtuuri välttää phase-gate- ja Big Design Up Front (BDUF) -menetelmien kulut ja viiveet. [4.]

Suunnittelemalla tuotteen etukäteen, kuten phase-gate ja Big Design Up Front -menetelmissä, hukataan aikaa ja rahaa sitouttamalla suunnitteluun paljon resursseja. Menetelmissä tehdään oletettuihin tarpeisiin ja vaatimuksiin perustuen liian täydellinen ratkaisu, joka ei lopulta vastaakaan asiakkaiden tarpeita ja vaatimuksia. Lisäksi liian monimutkaiset vaatimukset ja suunnitelmat voivat olla

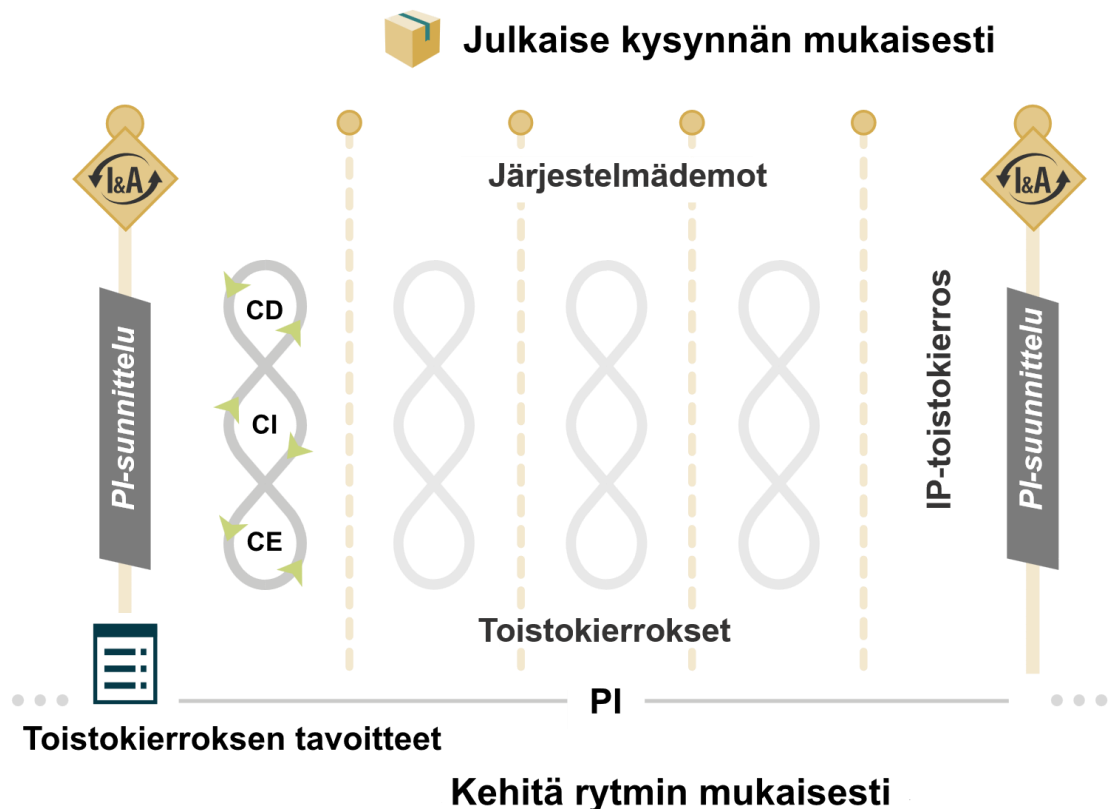
vaikeita toteuttaa. Kun tuotetta suunnitellaan osissa ja kerätään välissä palautetta asiakkaalta, saadaan luotua innovaatiota, ja on helpompi keskittyä toimivan ohjelmiston kehittämiseen. [4.]

SAFessa tiimit jaetaan julkaisujuniin (Agile Release Train, ART). Jokaisessa junnassa on tyypillisesti 50–125 jäsentä. Junien idea on välttää niin sanottuja siloja ja olla sen sijaan monitaitoisia yksiköitä. Kaikki tiimit työskentelevät toistokierrosten mukaisesti yhdessä ja esittävät todisteen päämääriensä saavuttamisesta järjestelmädemossa [12]. Tällä tavalla virheet voidaan havaita nopeammin, kuin jos tiimit toimisivat erillään eivätkä esittäisi saavutuksiaan muille. Kuvasta 1 voidaan havaita, kuinka erilliset yksiköt yhdistetään yhtenäiseksi monitaitoiseksi junnaksi. [13.]



Kuva 1. Julkaisujunat yhdistävät sovelluskehityksen eri siloista monitaitoisiksi tiimeiksi [13].

Suunnitteluväli eli PI (Planning Interval) on aikaikkuna, jonka tahdissa junat kulkevat eteenpäin. Tyypillisesti PI koostuu 4 kehitystoistokierroksesta, joita seuraa yksi innovaatio- ja suunnittelu eli IP (Innovation and Planning) -toistokierros. PI on ikään kuin koko junan toistokierros tai ketterän kehityksen sprint, jonka aikana suunnitellaan, toteutetaan, varmistetaan, tuotetaan liikearvoa ja saadaan nopeaa palautetta. Kuva 2 havainnollistaa suunnitteluvälin toistuvia tapahtumia. [14.]



Kuva 2. Tyypillinen PI, joka alkaa PI-suunnittelusta ja sisältää usean samanpi-tuisen toistokierroksen, joiden jälkeen alkaa uuden PI:n suunnittelu [14].

Kuvasta voidaan havaita, miten suunnitteluväli sisältää kehitystoistokierroksia, jotka hyödyntävät jatkuvaa toimitusta ja jatkuvaa integraatiota sekä innovaatio- ja suunnittelutoistokierroksen. Jokaisessa toistokierroksessa järjestetään järjestelmädemo. Järjestelmädemo on keskeinen käytäntö SAFessa. Siinä kukin ketterä tiimi esittää viime PI:n aikana aikaan saamansa työn. Järjestelmädemo järjestetään jokaisen PI:n loppupuolella, esimerkiksi Tarkasta ja mukauta -tapahtuman

yhteydessä. Tiimit valmistelevat demon etukäteen, jotta siitä tulee riittävän laadukas. Tapahtumaan osallistuvat sidosryhmät, kuten liiketoiminnan edustaja, asiakkaat tai heidän edustajansa ja ketterät tiimit. [12; 14.]

Järjestelmädemon tarkoitus on varmistaa, että tiimien työ etenee oikealla tavalla. Johto tai tiimit näkevät, mitä muut tiimit tekevät käytännössä, ja voivat antaa välitöntä palautetta työskentelystä. Järjestelmädemo ohjaa jatkuvaa integraatiota toistumaan joka PI:llä. Joissain tapauksissa, joissa täysi integraatio joka PI:llä on kalliimpaa toteuttaa, voidaan täyden integraation sijaan toteuttaa osittainen integraatio tai täysi integraatio harvemmin, esimerkiksi joka toinen PI. [12.]

PI:n päättävä Innovaatio- ja suunnittelutoistokierros (*IP sprint*) on tauko heikistä kehitystyöstä ja aika innovoida, suunnitella ja oppia. IP-toistokierros toimii myös suunnitelman puskurina odottamattomien tilanteiden varalta, mutta toistuva työn vuotaminen IP-toistokierrokselle on merkki siitä, että suunnitelmaa täytyy muuttaa jatkossa. IP-toistokierroksen ideaan kuuluu hackathonien tapaiset innovaatiotapahtumat tai muu innovatiivinen tai opettava toiminta, kunhan se liittyy yrityksen tavoitteisiin. [15.]

IP-toistokierroksessa järjestetään Tarkasta ja mukauta (Inspect and Adapt) -tapahtuma. Tapahtuman tarkoitus on kartoittaa ratkaisun nykyinen tila ja parantaa tilaa kehitysehdotusten avulla. Tässä insinööriyössä käsiteltävät kehitysehdotukset on kerätty muutaman vuoden aikana tällaisista tapahtumista. Vaikka jatkuva parantaminen on suuressa osassa SAFen muita osa-alueita, Tarkasta ja mukauta -tapahtuma varmistaa SAFen periaatteiden mukaisesti toistuvan aikavarauksen parantamista varten. Tarkasta ja mukauta -tapahtuma koostuu kolmesta osasta: [16.]

- PI:n järjestelmädemosta (*PI System Demo*)
- määrällisistä ja laadullisista mittareista
- menneen tarkastelusta (retrospective) ja ongelmanratkaisutyöpajasta.

PI:n järjestelmädemossa, jota ei pidä sekoittaa tavalliseen järjestelmädemoon, esitellään PI:n aikana saatua tulosta vertaamalla edellisessä PI-suunnittelussa arvioitua liikearvoa kuluneen PI:n aikana saavutettuun liikearvoon. Saavutus pisteisiin (achievement score) lasketaan yhteen suunnitellut pisteet muttei epävarmoja tavoitteita (uncommitted objectives), kun taas toteutuneesta arvosta epävarmat tavoitteet lasketaan mukaan. Kuva 3 havainnollistaa liikearvon laskentaa erilaisille tavoitteille. [16.]

Tavoitteet PI 3:lle	Suunnitelma	Toteutunut
• Structured locations and validation of locations	7	7
• Build and demonstrate a proof of concept for context images	8	8
• Implement negative triangulation by: tags, companies and people	8	6
• Speed up indexing by 50%	10	5
• Index 1.2 billion more web pages	10	8
• Extract and build URL abstracts	7	7
Epävarmat tavoitteet		
• Fuzzy search by full name	7	0
• Improve tag quality to 80% relevance	4	4
Yhteensä	50	45
Saavutettu osuus: 90%		

© Scaled Agile, Inc.

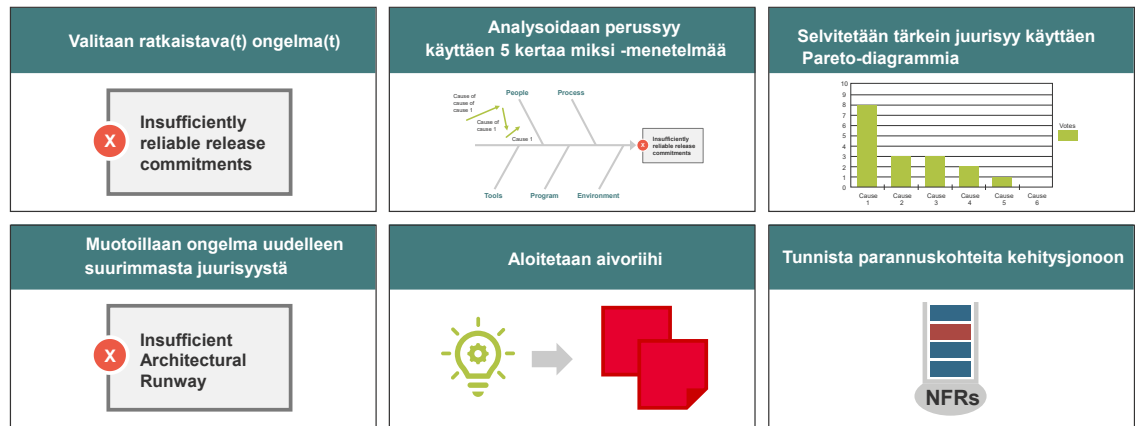
Kuva 3. Esimerkki suunniteltujen tavoitteiden toteutumisen perusteella tehtävästä saavutus pisteiden laskennasta [16].

PI:n järjestelmädemon jälkeen tiimit katselmoivat yhdessä keräämäänsä määrällistä ja laadullista dataa. Käyttämällä tiimien saavutuspisteitä tarkastellaan kunkin tiimin ennustettavuutta. Koko junan yhteisen tason tulisi olla vähintään 80 %, jotta suunnittelu on riittävän tehokasta. [16.]

Menneen tarkastelussa havaitaan ongelmat käyttämällä mitä vain ketterää retrospektiivimenetelmää. Tavallisesti kerätään tiimeiltä palautteita taululle siitä, mikä meni hyvin, mikä ei mennyt hyvin ja mitä täytyisi tehdä eri tavalla. Lopuksi kun ongelmat on havaittu, niiden ratkaisemiseksi järjestetään ongelmanratkaisutyöpaja. Ongelmanratkaisutyöpajassa on seuraavat vaiheet:

- Valitaan ratkaistava ongelma tai ongelmat
- analysoidaan perussy
- syiden analysoinnin jälkeen selvitetään tärkein juurisyy
- ongelma muotoillaan uudelleen käyttäen apuna havaittua tärkeintä juurisyytä
- aloitetaan aivoriihi
- äänestetään enintään kolme toteuttamiskelpoisinta ideaa

Kukin ratkaistava ongelma täytyy muotoilla ja ilmaista selvästi ja ytimekkäästi, painottaen kysymyksiä mitä?, missä?, milloin? ja mitä seuraamuksia tämä ongelma aiheuttaa? Perussyyn analysointiin Scaled Agile Inc. ehdottaa hyviksi työkaluiksi kalanruotokaaviota ja viisi kertaa miksi -menetelmiä [17; 18]. Juurisyy selvitetään tämän jälkeen käyttämällä Pareto-diagrammia [19]. Ongelmanratkaisutyöpajan vaiheet on havainnollistettu kuvassa 4. Saaduista parannuskohteista kirjoitetaan tarinoita ja ominaisuuksia. Nämä tehdään valmiiksi seuraavaa PI-suunnittelua varten. [16; 20.]

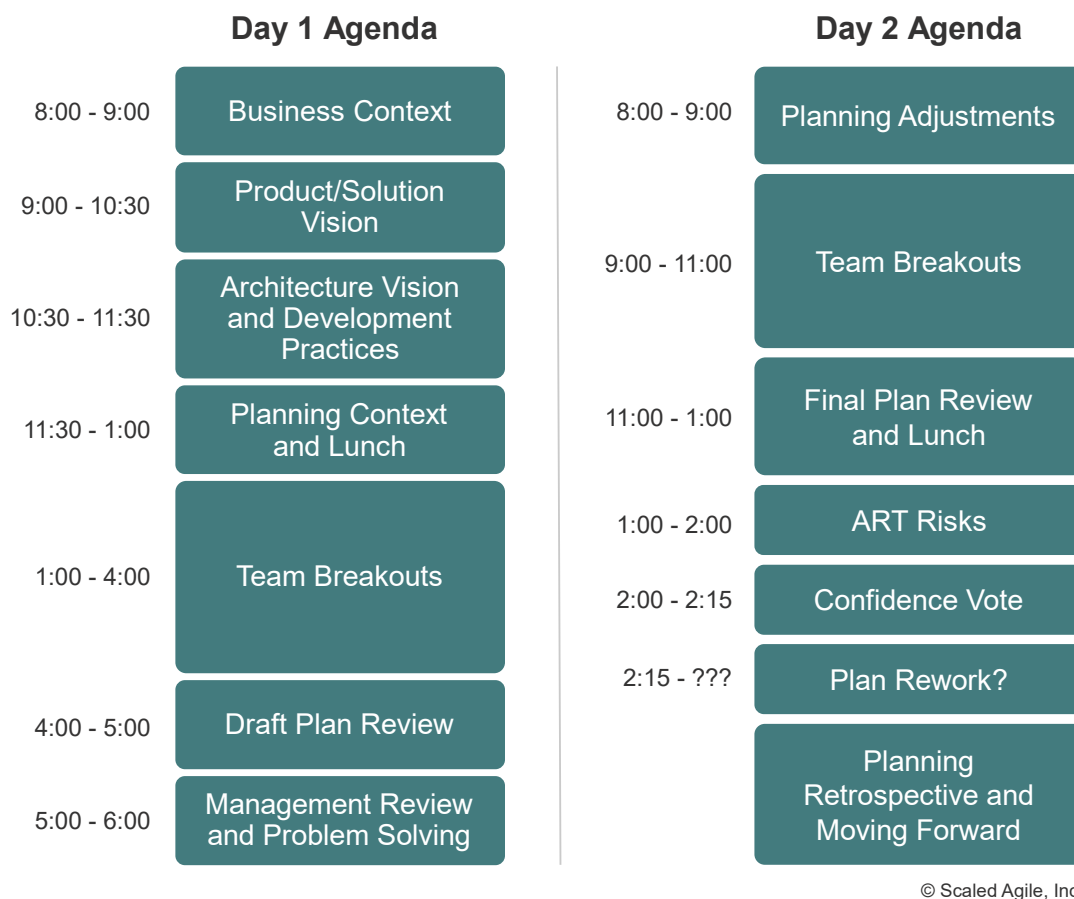


© Scaled Agile, Inc.

Kuva 4. Ongelmanratkaisutyöpajan vaiheet [16].

IP-toistokierroksen loppuksi tehtävä PI-suunnittelu on SAFen ydinidea. Julkaisu-
junan päällikön järjestämään tapahtumaan kutsutaan kaikki ART:n jäsenet.

Kuvassa 5 voidaan havaita tyypillinen esimerkki PI-suunnittelun asialistasta. Tä-
mäntapaista asialistaa voidaan käyttää, jos kaikki tiiminjäsenet sijaitsevat lähek-
käsillä aikavyöhykkeillä, mutta muussa tapauksessa suunnittelua voidaan ve-
nyttää kolmannelle päivälle. Osuuksien nimiä, järjestystä tai määrää voi muut-
taa tästä ohjelmasta, mutta tärkeintä on, että huomioon suunnittelussa otetaan
liiketoiminnan konteksti, tiekartta, visio ja tärkeimmät ominaisuudet ketterän toi-
mitusjunan kehitysjonolta. Suunnittelun tuloksena saadaan vähintään tavoitteet,
joihin ketterät tiimit voivat sitoutua (planned objectives) ja päivitetty julkaisu-
junan suunnittelutaulu (ART planning board). [21.]



Kuva 5. Esimerkki PI-suunnittelun agendasta [18].

Ensimmäisen päivän ohjelma alkaa esittelemällä liiketoiminnan konteksti (Business Context). Tässä osuudessa liiketoiminnan edustaja (Business Owner) tai organisaation johdon jäsen kertoo liiketoiminnan nykytilasta, jakaa portfolion vision ja kertoo, kuinka hyvin nykyiset ratkaisut palvelevat asiakkaita. [21.]

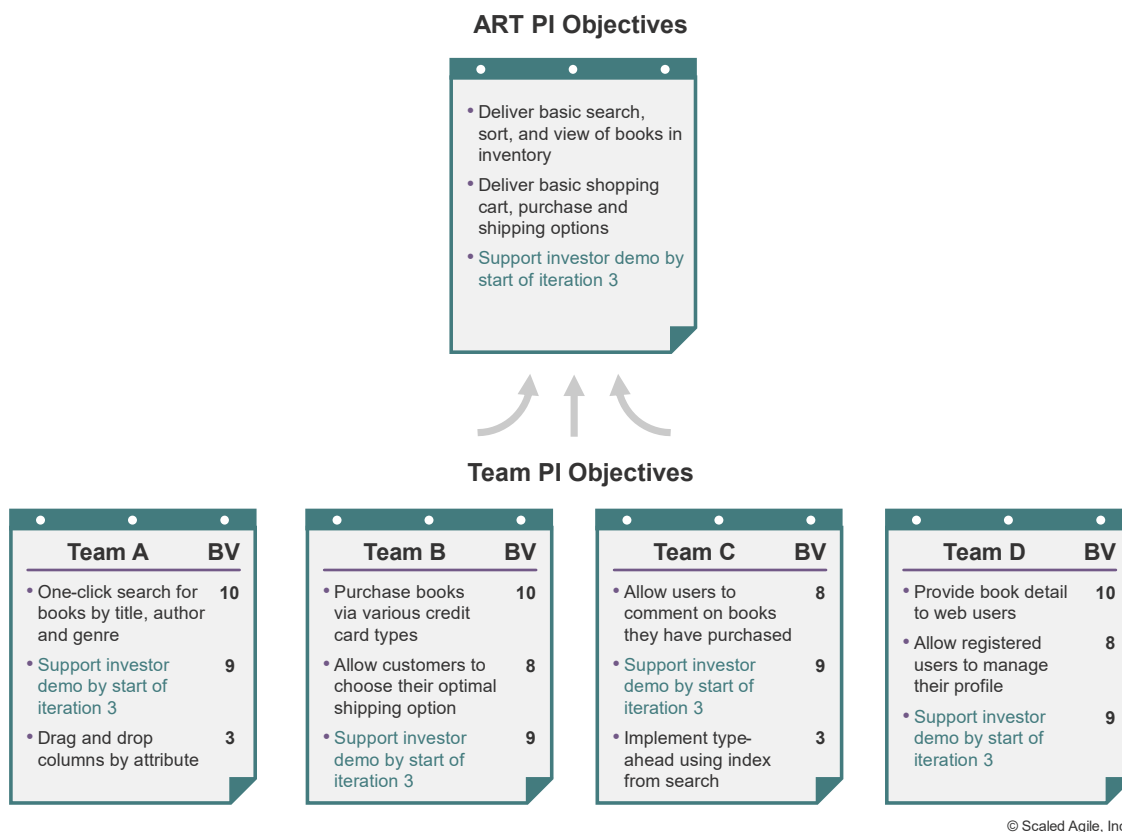
Tuotteen/ratkaisun visio (Product/Solution Vision) -osuudessa nykyinen visio viestitään esimerkiksi kertomalla tärkeimmistä tulevista ominaisuuksista. Arkkitehtuurinen visio ja kehityskäytännöt (Architecture Vision and Development) -osuus sisältää järjestelmäarkkitehdin esittämän arkkitehtuurisen vision. Seniorkehityspäällikkö voi myös kertoa seuraavan PI:n tulevista muutoksista kehityskäytäntöihin, kuten testiautomaatioon, DevOpsiin tai CI/CD-putkiin. [21.]

Suunnitelman konteksti ja lounas (Planning Context and Lunch) -osuudessa julkaisujunan päällikkö esittelee PI-suunnittelun prosessin ja tavoitteet. Jäljelle

jäävä aika käytetään lounastauolle. Tiimien breakoutit (Team Breakouts) -osuudessa tiimit kokoontuvat keskenään kartoittamaan, mitä riskejä heillä on ja mitä riippuvuuksia heillä on erityisesti muiden tiimien vastuulla oleviin ominaisuuksiin. Tiimit asettavat tavoitteet, mukaan lukien epävarmat tavoitteet, ART:n suunnittelutaululle. [21.]

Alustavan suunnitelman katselmoinnissa (Draft Plan Review) tiimit kertovat vuorotellen suunnitelmistaan pääasiat, kuten työkapasiteetin ja -kuormituksen, riskit ja riippuvuudet. Alustavan suunnitelman katselmointi on tiukasti aikataulutettu. Tämän jälkeen johdon katselmointi ja ongelmanratkaisu -osuudessa johdon jäsenet, kuten liiketoiminnan edustajat ja tuotehallinta, esittävät huolensa esimerkiksi suunnitelmien laajuudesta, resurssien riittävydestä ja riippuvuuksista. Sidosryhmät pysyvät kokouksessa pyrkien ratkaisemaan havaitsemansa ongelmat esimerkiksi muuttamalla suunnitelmien laajuutta, jotta suunnitelma on saatavissa. [21.]

Toinen suunnittelupäivä alkaa esittelemällä suunnitelman muutokset (Planning adjustments). Tässä osuudessa johto esittää edellisen päivän aikana luotuihin suunnitelmiin tehtävät muutokset esimerkiksi laajuuteen tai resursseihin. Tiimien breakoutit (Team breakouts) -osuudessa tiimit viimeistelevät suunnitelmansa ja tekevät tarvittavat muutokset johdon esittämien muutosten pohjalta. Liiketoiminnan edustaja asettaa liikearvopisteet tavoitteille. Kuvassa 6 voidaan havaita esimerkkejä tavoitteista ja niille annetuista liikearvopisteistä. [21.]



Kuva 6. Esimerkki PI-suunnittelussa syntyvistä tavoitteista tiimeille ja koko junalle. Tiimien tavoitteille on kullekin merkitty saatava liikearvo [21].

Kuva selventää PI-suunnittelussa syntyviä suunnitelmia: tiimi tekee omat suunnitelmat. Tiimien tavoitteet kootaan ja muotoillaan myös koko junan yhteisiksi tavoitteiksi. [21.]

Lopullisen suunnitelman katselmointi (Final plan review and lunch) -osuudessa tiimit esittävät suunnitelmansa jälleen yleisölle riskeineen ja esteineen. Jos liiketoiminnan edustaja hyväksyy suunnitelman, suunnitelma kootaan muiden tiimien suunnitelmien joukkoon yhteen näkymään, jotta kaikki näkevät suunnitelmien kokonaisuuden ja voivat ehdottaa muutoksia mahdollisiin huomaamiinsa epäkohtiin. [21.]

ART:n riskit -osuudessa tiimien esiintuomat riskit tutkitaan yksi kerrallaan ja luokitellaan johonkin seuraavista kategorioista:

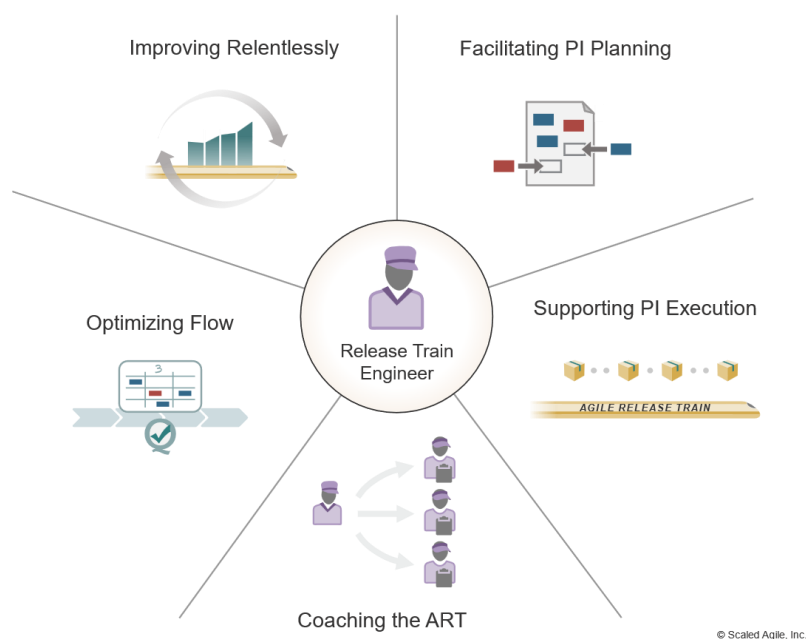
- Ratkaistu (Resolved) – riski todetaan ratkaistuksi keskustelun jälkeen
- omistettu (Owned) – jokin junan jäsen ottaa riskin vastuulle, sillä riskiä ei ole mahdollista ratkaista PI-suunnittelun aikana
- hyväksytty (Accepted) – riski on sellainen, joka täytyy ymmärtää ja hyväksyä, eikä se ole ratkaistavissa
- lievennetty (Mitigated) – riskin vaikutuksen vähentämiseksi tehdään suunnitelma.

Luottamusäänestys (Confidence vote) -osuudessa kaikki äänestävät siitä, kuinka luottavaisia he ovat suunnitelmaan asteikolla 1–5. Suunnitelman uudelleentyöstö (Plan rework) -osuudessa suunnitelmaa muokataan tarvittaessa, jos junan jäsenten luottamus suunnitelmaan on keskimäärin alle kolme. Lopuksi suunnitelman sujumisen tarkastelu ja jatkon suunnittelu (Planning retrospective and moving forward) -osuudessa käydään läpi mikä meni hyvin, mikä ei mennyt hyvin ja mitä täytyisi tehdä eri tavalla. [21.]

2.3 SAFen roolit

SAFessa on sekä omia erityisiä rooleja että muille viitekehyksille yhteisiä rooleja, kuten scrum master ja tuoteomistaja. Tässä luvussa käydään ketterän tiimin jäsenen kannalta tärkeimmät SAFen roolit.

Julkaisujunaa johtaa palvelevasti julkaisujunan päällikkö (RTE, Release Train Engineer), jonka tehtävänä on järjestää ja johtaa junan suunnittelu- ja kehitystahtumia ja ylläpitää julkaisujunan prosesseja. RTE on ketterien tiimien valmentaja, joka auttaa heitä tuottamaan tulosta. RTE seuraa junan käytäntöjen ja PI:n suoritusta. Hän muokkaa SAFen käytäntöjä vastaamaan organisaation tarpeisiin ja standardoivat ja dokumentoivat käytäntöjä. RTE:n tehtävät on esitetty kuvassa 7. [22.]



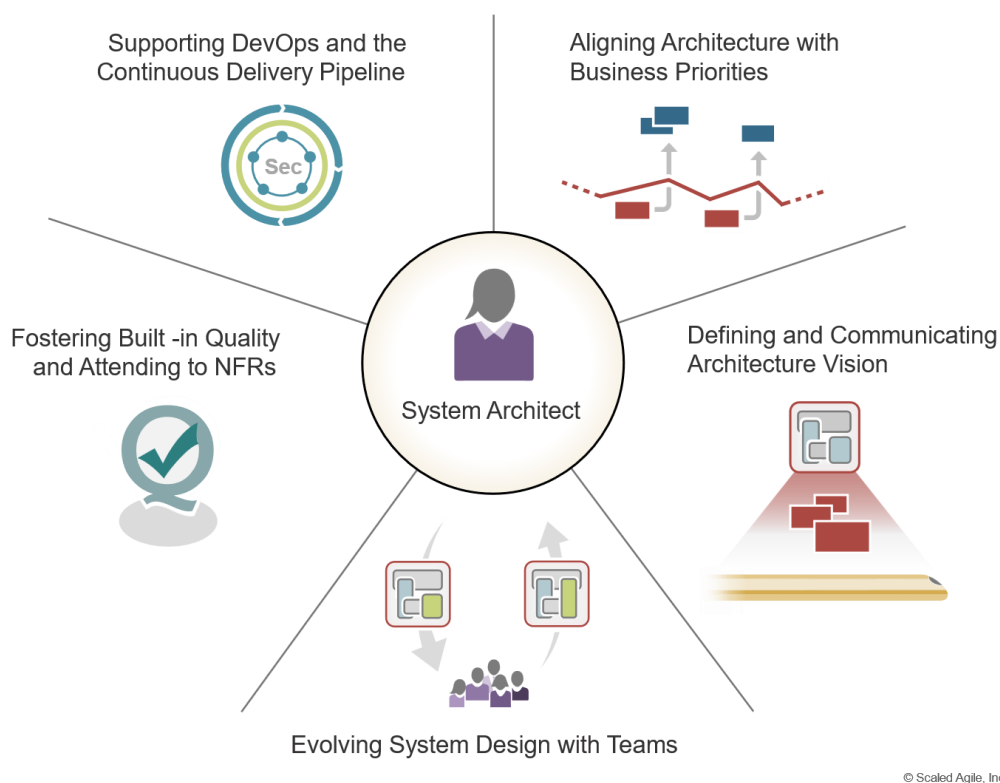
Kuva 7. RTE:n tehtävät [22].

Tuotehallinta (Product Management) on vastuussa asiakkaan tarpeiden täyttämisestä, julkaisujunan kehitysjonossa olevien ominaisuuksien hallinnasta ja prioriteetin määrittelystä sekä tuotestrategian, vision ja tiekartan kehittamisestä [23]. Tuotehallinta selvittää aktiivisesti, mistä asiakas saa tuotearvoa ja mitä tuotearvon toimittamiseen vaaditaan. Tuotehallinnan tehtävät on esitetty kuvassa 8. [24.]



Kuva 8. Tuotehallinnan tehtävät [24].

Järjestelmäarkkitehti (System Architect) määrittelee ja jakaa muille junan kehittämien tuotteiden teknisen ja arkkitehtuurisen vision. Järjestelmäarkkitehteja voi olla tarvittaessa useita. Arkkitehti varmistaa, että arkkitehtuuri tukee tuotehallinnan tunnistamia liiketoiminnan prioriteetteja. Järjestelmäarkkitehdin yksi tärkeistä tehtävistä on valmistaa juna uusia ominaisuuksia varten ennen niiden toteutuksen alkamista määrittämällä mahdollistajia (enablers) junan kehitysjoon. Järjestelmäarkkitehdin tehtävät on esitetty kuvassa 9. [25.]



Kuva 9. Järjestelmäarkkitehdin tehtävät [25].

Liiketoiminnan edustajat (Business Owners, BO) ovat lean-ketteriä junan johtajia [26]. He edustavat junan päivittäisessä toiminnassa liiketoiminnan ja asiakkaiden etua. Scaled Agile Inc -järjestö kuvaa liiketoiminnan edustajia näin:

Liiketoiminnan omistajat (BO) ovat pieni joukko sidosryhmiä, joilla on ensisijainen liiketoiminnallinen ja tekninen vastuu ketterän julkaisujunan (ART) kehittämisen ratkaisun hallinnoinnista, vaatimustenmukaisuudesta ja sijoitetun pääoman tuotosta (ROI). [5.]

Vaikka ketterässä kehityksessä ja SAFessakin korostetaan tiimien itsehallintointia, kaikkea johtamista ei ole SAFessa poistettu. Liiketoiminnan edustajat eivät kuitenkaan saa valvoa tiimejä tai antaa heille tehtäviä. Liiketoiminnan edustaja vie muutoksia läpi esimerkiksi johtamisen avulla. He ovat mukana suunnittelussa viemässä läpi lean-ketteriä käytäntöjä. [27.]

Lean-portfolioon hallintaa operoi pääasiassa lean-portfolioon hallinnasta vastaavat johtajat, mutta liiketoiminnan edustajat ovat myös usein mukana hallinnoinnissa [28]. Liiketoiminnan edustajat valvovat, että yksittäiset arvovirrat ja portfolio ovat

linjassa. He usein avustavat lean-portfolion hallintaa budjetoinnissa osoittamaan portfolion budjetin portfolion arvovirtoihin. [27.]

Liiketoiminnan edustajien tärkeimpiin työtehtäviin kuuluu portfolion kehitysjonon päivitys ja PI-suunnitteluun liittyvät tehtävät. PI-suunnittelussa liiketoiminnan edustaja muun muassa esittelee liiketoiminnan kontekstin ja katselmoi tiimien alustavat ja lopulliset suunnitelmat. Liiketoiminnan edustajan tehtävät on esitetty alueittain kuvassa 10. [27.]



Kuva 10. Liiketoiminnan edustajan tehtävät [27].

Ketterät tiimit ovat monitaitoisia, tyypillisesti kymmenen hengen tai alle kymmenen hengen itsehallinnoivia ja itse työnsä organisoivia tiimejä. Tiimien tehtävä on toteuttaa ratkaisuja, jotka vastaavat asiakkaiden sekä sidosryhmien odotuksia ja vaatimuksia. Työskentelemällä pieni toistokierros kerrallaan ketterä tiimi pyrkii oppimaan, saamaan usein palautetta ja mukautumaan palautteeseen. [29.]

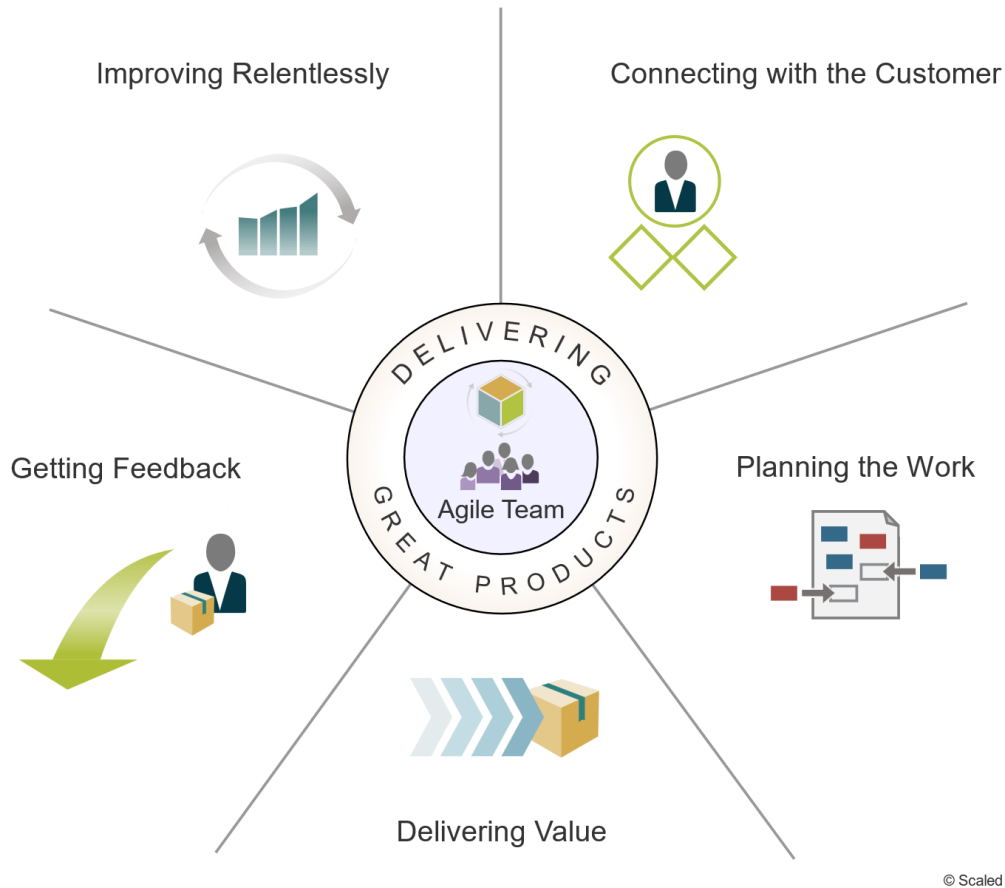
Ketterät tiimit sisältävät kaikki toiminnot, joita tarvitaan arvon toimittamiseen asiakkaalle. Ketterät tiimit siis määrittelevät ja suunnittelevat ominaisuudet, rakentavat suunnittelemansa ratkaisun, testaavat sen ja julkaisevat ominaisuudet asiakkaalle toistokierroksittain. [29.]

Tiimit voidaan organisoida eri tavoitteiden mukaan. Virtaan linjatut tiimit (stream aligned teams) keskittyvät loppukäyttäjän palvelemiseen ja kykenevät tuottamaan itsenäisesti arvoa asiakkaille alusta loppuun. Monimukaisten alijärjestelmien tiimit (complicated subsystem teams) hoitavat erityisosaamista tarvitsevia järjestelmiä vähentäen muiden tiimien kuormaa. Alustan tiimit (platform teams) hoitavat sovellusten palveluita ja rajapintoja virtaan linjattujen tiimien käytettäväksi. Mahdollistavat tiimit (enabling teams) taas tarjoavat työkaluja, palveluita ja väliaikaisesti asiantuntemustaan muille tiimeille. [29.]

Ketterissä tiimeissä on kaksi erikoisroolia: Scrum master/tiimivalmentaja ja tuoteomistaja (product owner). Scrum masterin tehtävä on muun muassa toteuttaa ja ylläpitää ketteriä menetelmiä, optimoida ja parantaa tiimin suorituskykyä sekä tehdä yhteistyötä RTE:n kanssa junan parantamiseksi. Hän valmistaa tiimin PI-suunnitteluun. Ennen suunnittelua scrum master varmistaa, että tiimi ymmärtää tulevat ominaisuudet, joita tuotehallinta ja liiketoiminnan edustajat ovat suunnitelleet. Hän auttaa tiimiä ja tuoteomistajaa tunnistamaan työtä tulevalle PI:lle. Scrum master myös luo toistuvat palaverit, kuten kehitysjonon jalostuksen (backlog refinement), tiimin päivittäispalaverin (team sync, daily), toistokierrosuunnittelun (team planning), retrospektiivin ja demot sekä usein johtaa niitä. [29; 30.]

Tuoteomistaja edistää tiekarttaa ja visiota sekä pitää yhteyttä asiakkaisiin. Nimensä mukaisesti hän myös edustaa asiakasta ja tuotteen parasta. Työskentelemällä asiakkaan ja tiimien kanssa he auttavat tiimiä määrittämään sellaisia tarinoita ja tehtäviä kehitysjonoon, jotka kohdistuvat asiakkaan tarpeisiin ja ylläpitävät tuotteen teknistä eheyttä. Tuoteomistaja kerää tiimin työtä avustavaa tietoa useista lähteistä. Hän tuntee asiakkaat ja heidän tarpeensa sekä yrityksen sisäiset sidosryhmät, kuten liiketoiminnan edustajat, Lean-portfolion hallinnan,

tuotehallinnan ja järjestelmäarkkitehdit. Tuoteomistaja tuntee myös muut tuoteomistajat ja tekee heidän kanssaan yhteistyötä. Kuvassa 11 esitetään ketterän tiimin toiminta-alueet. [29; 31.]



Kuva 11. Ketterän tiimin toiminnot. Ketterä tiimi keskustelelee asiakkaan kanssa, suunnittelee työnsä, toimittaa liikearvoa, saa palautetta ja parantaa sisukkaasti. [29].

2.4 Hyödyt

SAFesta on tutkittua hyötyä ohjelmistokehityksen prosessien parantamiseen [32, s. 85]. SAFe tarjoaa yksityiskohtaiset valmiit ohjeet ketterän kehityksen toteuttamiseen laajassa mittakaavassa [33]. SAFen periaatteissa mainittu läpinäkyvyys on käytännössä parantunut SAFea toteuttavien organisaatioiden toiminnassa. Kommunikaatio asiakkaiden kanssa ja tiimien välillä on parantunut. Suunnittelu on muuttunut tehokkaammaksi. Tämä on sen ansiota, että tiimit tietävät, minkä parissa muut tiimit työskentelevät ja heillä on tiedossa koko

organisaation tiekartta. SAFen tapahtumat osaltaan tehostavat tätä tiedon jakamista. [32, s. 85–86.]

SAFe on tarjonnut useille organisaatioille paremmin ennustettavia prosesseja. Ennustettavuus on parantunut ainakin tiekarttojen ja suunnittelutapahtumien myötä. Tiimit tietävät tulevat tehtävät kauaskin tulevaisuuteen, mikä helpottaa tiimien itsenäistä suunnittelua. [32, s. 86; 24.]

SAFe parantaa työskentelyn ketteryyttä käyttämällä ketterän kehityksen menetelmiä, kuten toistokierroksia, työn pilkkomista ja monitaitoisia tiimejä. SAFen tapahtumien yhteydessä pystyy huomaamaan esteet työlle ja reagoimaan niihin. Julkaisujunan kehitysjonossa olevien ominaisuuksien prioriteetin määrittely käyttämällä WSJF (Weighted Shortest Job First) -menetelmää [34] tekee tiimien itsenäisestä suunnittelusta helpompaa. WSJF-menetelmässä lasketaan kuinka paljon viivästyksestä aiheutuisi haittaa ja jaetaan se työn pituudella. [32, s. 65–69.]

SAFe auttaa varmistamaan, että organisaation eri tasot ovat linjassa liiketoiminnan tavoitteiden kanssa. Tehokas työn priorisointi, portfolion hallinta, suunnittelutapahtumat ja SAFen roolit ovat auttaneet tuomaan parannuksia yhteisen suunnan saavuttamiseen. SAFesta saatuja liiketoiminnallisia hyötyjä ovat nopeampi markkinoilletuontiaika, tiiviimpi julkaisuaikataulu, nopeampi tuotteiden ja ominaisuuksien toimitus ja parempi asiakastytyväisyys. Tuotteet on saatu toimitettua useammin ajallaan SAFen käyttöönottaneissa organisaatioissa. [32, s. 93; 35; 36.]

2.5 Haasteet

Vaikka SAFessa on useita todistettuja hyötyjä useille organisaatioille, siinä on myös haasteita ja haittapuolia. Samoja haasteita on mainittu useissa lähteissä. [32; 35; 36; 37]

Yksi haaste on vastarintaisuus muutoksille. Ciancarinin ym. [37, s. 179] mukaan vastarintaisuus johtuu SAFen monimutkaisuudesta ja jäykkyydestä, jonka vuoksi monet eivät pidä SAFEa ketteränä viitekehyksenä, vaan pitävät sitä kykenemättömänä mukautumaan nopeisiin vaatimusten muutoksiin. SAFEen siirtymistä on pidetty siirtymiseksi ketterästä kehityksestä takaisin kohti vesiputouksimallia [35, s. 347].

Ciancarinin ym. [37, s. 179] mukaan SAFessa on piirteitä, jotka vetoavat perinteistä organisaation laajuista ylhäältä-alas hierarkiaa kannattaviin johdon jäseniin. Scrum Alliancen johtokunnan jäsenen Denningin [38] mukaan SAFessa on riski, että organisaatio palautuu takaisin epätuottoisaan ja innovaatiota tukehduttavaan vertikaaliseen hallintorakenteeseen. SAFE vaatii paljon resursseja, motivoituneita työntekijöitä, koulutusta ja muutoksen mittausta, jotta varmistetaan, että SAFE aikaansaa parannusta [37, s. 178].

Monissa lähteissä SAFen joustavuutta pidettiin haasteellisena, sillä työ on suunniteltu pitkälle 10–12 viikon PI:lle, ja jos suunnitelmaan tarvitaankin suuria muutoksia, on suunnitelman muuttaminen hankalaa. [32, s. 89]

3 CMMI

3.1 CMMI ja sen soveltaminen

CMMI (Capability Maturity Model Integration) on prosessinkehittämiseen suunniteltu malli, joka sisältää tehokkaan prosessin ominaisuuksia. Se on alun perin kehitetty sovelluskehityksen prosessinparannukseen, mutta versiossa 1.3 siitä oli kolme versiota: CMMI-DEV (sovelluskehitys), CMMI-SVC (palvelut) ja CMMI-ACQ (hankinnat) [39]. CMMI-DEV on käytössä monella alalla, kuten ilmailussa, pankkitoiminnassa, ohjelmistoalalla, autonvalmistuksessa, tietokonelaitteistoissa ja tietoliikenteessä. Sen sisältämät käytännöt pystyvät tukemaan muun muassa projektinhallintaa, prosessinhallintaa, järjestelmäsuunnittelua sekä laitteistotekniikan ja ohjelmistotekniikan kehitystä ja ylläpitoa. [40.]

CMMI pohjautuu CMM:ään (Capability Maturity Model), jonka kehitys alkoi vuonna 1986. Kummankin mallin kehittäjänä toimi Carnegie Mellon yliopistossa sijaitseva Software Engineering Institute (SEI), jota sponsoroi Yhdysvaltain puolustusministeriö. Carnegie Mellon yliopisto perusti myöhemmin uuden organisaation, CMMI Instituten, jolle CMMI-tuotteet siirtyivät kokonaisuudessaan vuonna 2013. Vuonna 2016 kaupallinen tietohallintayritys ISACA hankki CMMI Instituten. [41.]

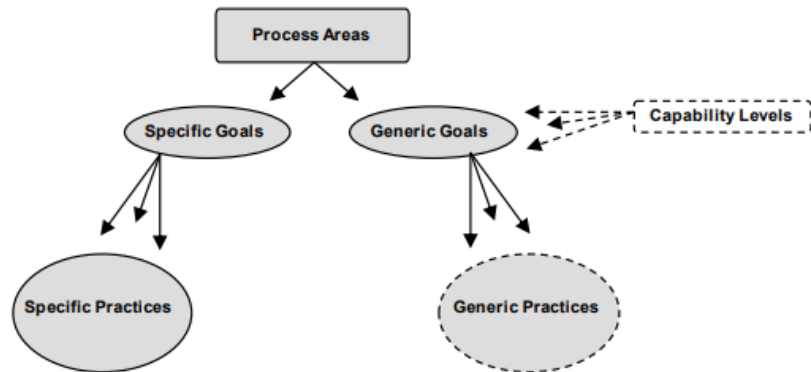
Uusin versio CMMI:stä on vuonna 2023 julkaistu 3.0 [42], mutta ainoastaan versio 1.3 ja sitä vanhemmat ovat saatavilla vapaasti Carnegie Mellon yliopiston verkkosivuilta [40]. Työssä hyödynnettiin tämän vuoksi CMMI-DEV versiota 1.3. CMMI:tä sovellettiin tässä työssä liittämällä Tarkasta ja mukauta -tapauksissa saadut palautteet CMMI:n prosessialueisiin. Tämän jälkeen on helpompi tunnistaa erityisen tärkeitä palautteita ja parannuskohteita.

3.2 Vaiheistetun ja jatkuvan kehittämisen mallit

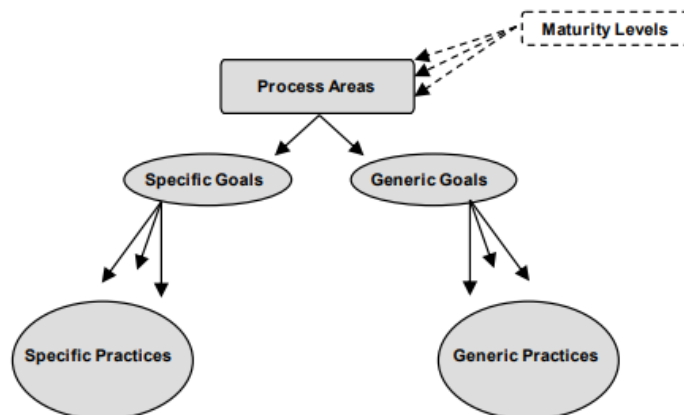
CMMI tarjoaa kaksi erilaista mallia prosessien parantamiseen. Vaiheistetun kehittämisen mallissa käytetään kypsyystasoa (maturity levels) organisaation kunkin prosessialueen tason määrittämiseen. Jatkuvan kehittämisen mallissa taas käytetään kyvykkyyksitasoja (capability levels) jokaisen yksittäisen prosessin tason määrittämiseen. [40, s. 21–23.]

Kummassakin mallissa on prosessialueita, joista jokainen sisältää yleiset tavoitteet (generic goals) ja kyseiselle prosessialueelle erityisiä tavoitteita (specific goals). Yleiset tavoitteet täytetään käyttämällä yleisiä käytäntöjä (generic practices) ja erityiset tavoitteet vastaavasti erityisillä käytännöillä (specific practices). Kuvassa 12 esitetään vaiheistetun kehittämisen ja jatkuvan kehittämisen mallien erot. [40, s. 22–23.]

Continuous Representation



Staged Representation



Kuva 12. Jatkuvan kehittämisen malli (ylhällä) ja vaiheistetun kehittämisen malli (alhaalla) [40, s. 22].

Jatkuvan kehittämisen malli sopii paremmin, jos halutaan parantaa yksittäisiä prosessialueita. Vaiheistetun kehittämisen malli sisältää valmiiksi määritellyt prosessialueet, joiden täytyy olla vakiintuneita, ennen kuin kehitetään seuraavan kypsyyden prosessialueita. Jos halutaan parantaa kattavasti koko organisaation yleistä suorituskykyä, kannattaa valita vaiheistetun kehittämisen malli. Koska tässä työssä halutaan kehittää vain tunnistettuja heikkouksia yksittäisissä prosessialueissa, jatkuvan kehittämisen malli sopii paremmin. [43, s. 15–17.]

3.3 Prosessialueet

Prosessialueet ovat kokoelma toisiinsa liittyviä käytäntöjä, jotka yhdessä toteutettuna aikaansaavat jonkin kyvykkyyden (capability) käytettäessä jatkuvaa (continuous) mallia tai kypsyyden (maturity) käytettäessä vaiheistetun kehittämisen mallia. CMMI-DEV -mallissa on 22 eri prosessialuetta. Prosessialueet jaetaan neljään eri kategoriaan taulukon 1 mukaisesti. [43, s. 25–29.]

Taulukko 1. CMMI-DEV:in prosessialueet [43, s. 27–28].

Prosessinhallinta (Process Management)	Projektinhallinta (Project management)	Tekniikka (Engineering)	Tukitoiminnot (Support)
<ul style="list-style-type: none"> • Organizational Process Definition (OPD) • Organizational Process Focus (OPF) • Organizational Performance Management (OPM) • Organizational Process Performance (OPP) • Organizational Training (OT) 	<ul style="list-style-type: none"> • Integrated Project Management (IPM) • Project Monitoring and Control (PMC) • Project Planning (PP) • Quantitative Project Management (QPM) • Requirements Management (REQM) • Risk Management (RSKM) • Supplier Agreement 	<ul style="list-style-type: none"> • Product Integration (PI) • Requirements Development (RD) • Technical Solution (TS) • Validation (VAL) • Verification (VER) 	<ul style="list-style-type: none"> • Causal Analysis and Resolution (CAR) • Configuration Management (CM) • Decision Analysis and Resolution (DAR) • Measurement and Analysis (MA) • Process and Product Quality Assurance (PPQA)

	Management (SAM)		
--	---------------------	--	--

Taulukosta voidaan havaita, että CMMI-mallin prosessialueet kattavat useita SAFen prosesseja, kuten tuotteen integraation, projektinsuunnittelun ja mittauksen ja analyysin.

3.4 CMMI ketterässä kehityksessä

CMMI on vanha malli. Ketterän kehityksen menetelmät taas kehittyvät jatkuvasti. Tässä luvussa tarkastellaan, onko CMMI edelleen sopiva ketterän kehityksen apukeinona sekä onko CMMI ristiriidassa ketterän kehityksen kanssa.

CMMI on kokoelma abstrakteja prosesseja, joiden toteuttamiseen on esitetty käytäntöjä. Nämä käytännöt ovat kuitenkin suuntaa antavia, ja tärkeintä CMMI-mallissa on prosessien toteutuminen, eikä se, miten ne toteutetaan. CMMI kertoo mitä pitäisi tehdä, kun taas ketterät menetelmät keskittyvät yksityiskohtaisemmin siihen, kuinka ohjelmistoa kehitetään ja miten kehittäminen pitäisi tehdä. Näin ollen CMMI ja ketterä kehitys eivät ole suoranaisesti ristiriidassa. [44, s. 2.]

CMMI:n ja ketterän kehityksen yhteiskäyttö organisaatiossa on tavallista ja niiden yhteiskäyttö on lisääntynyt [44, s. 1]. CMMI:n prosessit voi toteuttaa vähintään tasolle 2 tai 3 käyttämällä ketteriä menetelmiä ja korkeammalle tasolle käyttämällä ketterien menetelmien lisäksi muita menetelmiä. [45, s. 21.]

CMMI:n ja ketterän kehityksen yhdistäminen voi parantaa muun muassa ymmärtämystä sovelluskehityksen prosesseista ja vaatimusten määrittelyä [45, s. 31]. Vikojen tunnistamisen parantumisen kautta myös laatu voi parantua [45, s. 32].

Toisaalta on huomionarvoista, että CMMI:n täysimittainen toteuttaminen organisaatiossa vaatisi huomattavia resursseja, ja CMMI:n ohjaus ja vastuu ovat osittain ristiriidassa ketterän kehityksen kanssa [46, s. 237]. Tämän työn

tekemiseen näistä ongelmista ei ole haittaa, sillä kaikkia prosessialueita ei oteta huomioon arvioinnissa.

3.5 Prosessien arviointi

Carnegie Mellon yliopisto on kehittänyt prosessinarviointia standardoivan ja kuvaavan menetelmän Standard CMMI Appraisal Method for Process Improvement eli SCAMPI:n. SCAMPI:sta on 3 eri versiota A, B ja C, joista A on kattavin menetelmä monine käytäntöineen, suurineen arviointitiimeineen ja josta saadaan tulokseksi luokitus. C-menetelmä taas ei vaadi tiimiä, sisältää vähemmän käytäntöjä ja vaiheita sekä on nopein ja edullisin arviointimenetelmä. SCAMPI C -menetelmässä tietolähteinä voi käyttää haastatteluja tai ainoastaan dokumenttien katselmointia. Tähän työhön paras valinta näiden seikkojen vuoksi on SCAMPI C -menetelmä. [47, s. 48.]

SCAMPI-menetelmät on määritetty menetelmien määrittelydokumentissa (Method Definition Document, MDD). MDD:n osassa 2 on määritelty kuhunkin SCAMPI-menetelmään sisältyvät prosessivaiheet. [48.] SCAMPI B:n ja C:n vaiheita vertaillaan taulukossa 2.

Taulukko 2. SCAMPI:n prosessivaiheiden vaatimukset [47, s. 83–99]. S=sovelletaan, V=valinnainen.

Prosessivaatimus	SCAMPI B	SCAMPI C
1.1 Analyze Requirements		
1.1.1. Determine Appraisal Objectives	S	S
1.1.2. Determine Appraisal Constraints	S	S
1.1.3. Determine Appraisal Scope	S	S
1.1.4. Determine Outputs	S	S
1.1.5. Obtain Commitment to Appraisal Input	S	S
1.2. Develop Appraisal Plan		

1.2.1. Tailor Method	S	S
1.2.2. Identify Needed Resources	S	S
1.2.3. Determine Cost and Schedule	S	S
1.2.4. Plan and Manage Logistics	S	S
1.2.5. Document and Manage Risks	S	S
1.2.6. Obtain Commitment to Appraisal Plan	S	S
1.3 Select and Prepare Team		
1.3.1. Identify Team Leader	S	S
1.3.2. Select Team Members	S	V
1.3.3. Prepare Team	S	V
1.4. Obtain and Analyze Initial Objective Evidence		
1.4.1. Prepare Participants	S	S
1.4.2. Administer Instruments	V	V
1.4.3. Obtain Initial Objective Evidence	V	V
1.4.4. Inventory Objective Evidence	S	V
1.5. Prepare for Collection of Objective Evidence		
1.5.1. Perform Readiness Review	S	S
1.5.2. Prepare Data Collection Plan	S	S
1.5.3. Replan Data Collection	V	V
2.1. Examine Objective Evidence		
2.1.1. Examine Objective Evidence from Instruments	V	V
2.1.2. Examine Objective Evidence from Presentations	V	V
2.1.3. Examine Objective Evidence from Documents	S	V
2.1.4. Examine Objective Evidence from Interviews	S	V
2.2. Verify and Validate Objective Evidence		
2.2.1. Verify Objective Evidence	S	S
2.2.2. Characterize Implementation of Model Practices	S	S
2.2.3. Validate Practice Implementation Gaps	S	V

2.3. Document Objective Evidence		
2.3.1. Take/Review/Tag Notes	S	V
2.3.2. Record Presence/Absence of Objective Evidence	S	S
2.3.3. Document Practice Implementation Gaps	S	V
2.3.4. Review and Update the Data Collection Plan	S	V
2.4. Generate Appraisal Results		
2.4.4. Document Appraisal Results	S	S
3.1. Deliver Appraisal Results		
3.1.1. Present Final Findings	S	S
3.1.2. Conduct Executive Session(s)	V	V
3.1.3. Plan for Next Steps	S	S
3.2. Package and Archive Appraisal Assets		
3.2.1. Collect Lessons Learned	S	S
3.2.2. Generate Appraisal Record	S	S
3.2.3. Provide Appraisal Feedback to CMMI Steward	S	S
3.2.4. Archive and/or Dispose of Key Artifacts	S	S

Taulukosta voidaan havaita, että SCAMPI C:n vaatimukset saavutettiin tätä insinööriä tehdessä. Työtä tehdessä on määritetty, mitä tutkitaan, eli Tarkasta ja mukautu -palautteita. Ennen insinööriä on määritelty aikataulu työlle ja se kuinka paljon työhön saa käyttää työaika eli kuinka paljon työn tekeminen maksaa itselle ja työnantajalle. Tässä raportissa on muun muassa esitelty työn tulokset, mahdollinen jatkokäyttö tuloksille ja opitut asiat tutkimustavoista. Lisäksi taulukosta voidaan havaita, että SCAMPI B -menetelmä on hiukan kattavampi, sillä se vaatii monta arvioijaa ja siinä on pakollista tutkia tavoitetodisteita dokumenteista ja haastatteluista.

4 Menetelmät

Tutkimus tehtiin empiirisenä tutkimuksena käyttämällä sekä määrällisiä että laadullisia menetelmiä. Tutkimusaineistona käytettiin SAFen ja CMMI:n teorian osalta verkkoaineistoa ja kirjallisuutta sekä palautteiden ja palaute-ehdotusten keräämiseen CGI:n sisäistä Atlassian Confluence -materiaalia, kirjallisuutta ja CMMI-mallia.

Tutkimusprosessissa tehtiin tiedonhaun rajaus ja suunnitelma sekä tutkittiin teoriaa SAFe-viitekehystä ja CMMI-mallista. Confluencesta löytyvästä Tarkasta ja mukauta -palautteista sekä Microsoft Teamsissä esiintyneistä palautteista etsittiin toistuvia ongelmakohtia. Kun jokin aihealue toistui vähintään kaksi kertaa, tutkittiin, mihin CMMI:n prosessialueeseen aihe liittyi. Nämä prosessialueet lisättiin arvioitaviin prosesseihin. SCAMPI-prosessinarviointimenetelmää käyttämällä arvioitiin prosessien nykytilaa. SCAMPI:n tavoitetodisteina (objective evidence) käytettiin Tarkasta ja mukauta -palautteita sekä Confluence-materiaalia. CMMI-mallia käyttämällä suoritettiin kuiluanalyysi. Lopuksi kerättiin ratkaisuehdotuksia palautteista, kirjallisuudesta ja CMMI-mallista.

Tutkimus rajattiin nykyisestä Confluence-ympäristöstä löytyvän Tarkasta- ja mukauta -palautteen analysointiin. Teorian osalta tutkimus rajoitettiin keskeisiin käsitteisiin ja käytäntöihin. Käsiteltävät palautteet rajoitettiin sellaisiin, jotka ovat toistuneet vähintään kahdesti ja joiden tiedettiin olevan edelleen ongelmana ja uusien toistumiskerta oli vähintään PI 40.

Ensimmäinen kriteeri palautteiden analysoinnissa oli niiden toistomäärä. Toinen kriteeri palautteiden tärkeysjärjestyksen määrittämiseen oli se, mikä palaute antaa eniten parannusta prosessin sujuvuuteen vähimmällä työpanoksella. Kolmas kriteeri palautteiden tärkeysjärjestyksen määrittämisessä oli niihin liittyvän prosessialueen kypsyysluokitus. Matalamman kypsyysasteen prosessialueeseen liittyvällä aineistolla oli suurempi painoarvo.

5 Tulokset

5.1 Havainnot

Palautteet liittyivät laajasti eri CMMI-prosessialueisiin, mutta joitakin PI:stä toiseen toistuvia palautteita tuli esille. Taulukossa 3 havaitaan 6 eri kategoriaa, joista löydettiin rajaukset täyttäviä palautteita.

Taulukko 3. Tarkasta ja mukauta -palautteiden kategoriat, CMMI-prosessialueet ja prosessialueiden alin kypsyyssuokitus. Taulukko on järjestetty prioriteettijärjestykseen ylimmän ollessa tärkein parannuskohde ja alimman vähiten tärkein.

Palautteen aihe	PI:t	CMMI-prosessialue	CMMI-kypsyyssuokitus
Testiympäristöjen ongelmat	26, 28, 34, 42, 43, 44	Verification (VER)	3
Yllättävät muutokset vaatimuksiin tai kapasiteettiin (yllättävät muutokset, epävarmuus)	32, 33, 34, 40	Requirements Management (REQM), Project Planning (PP)	2
Suunnitellut ominaisuudet eivät valmistu, suunnitelmat eivät ole realistisia	36, 39, 40	Project Planning (PP)	2
Liikaa vikoja tuotantoympäristössä	36, 43	Verification (VER)	3
BO:t ei osallistu Tarkasta ja mukauta -tapahtumiin	29, 40	Measurement and Analysis (MA)	2
Tapahtumat pitäisi olla englanniksi tai suomea osaamattomat täytyisi ottaa huomioon	29, 30, 44 (Teams)	Project Planning (PP)	2

5.2 Ratkaisuehdotukset

Tässä luvussa esitetään ratkaisuehdotuksia palautteiden ongelmiin. Ratkaisuehdotukset ongelmiin ovat SAFen käytännöistä, palautteista itsestään ja yleisistä hyväksi havaituista käytännöistä.

Testiympäristöjen ongelmat

Yksi selkeimmistä palautteissa esille tulleista ongelmista oli testiympäristöjen epävakaus. Palautteiden mukaan testiympäristöjen tavallisin ongelma oli koonnin (build) epäonnistuminen ja siitä johtuva testiympäristön tai sen osan kaatuminen.

Testiympäristöjen ongelmien ehkäisemiseen on käytäntöjä, joiden noudattaminen kannattaa varmistaa. Testiympäristöjen seuraaminen auttaa ehkäisemään ongelmia, jotka johtuvat testipalvelimien ylikuormituksesta tai ongelmista ohjelmien suorituskyvyssä. Esimerkiksi James Turnbullin kirja *The Art of Monitoring* kertoo suorituskyvyn seuraamiseen hyödyllisistä käytännöistä ja työkaluista [49]. Testiympäristön tilan palauttaminen virhetilasta edelliseen toimivaan tilaan kannattaa tehdä mahdollisimman sujuvaksi (rollback strategy).

Selvää on, että testiympäristöjen ongelmat ovat toistuneet pitkään ja ne aiheuttavat työtuntien menetystä. Tämän vuoksi ongelman ratkaiseminen arvioitiin tärkeäksi, vaikka todennäköisesti ongelmien ratkaiseminen vie paljon resursseja. Nykyinen Jenkins-automaatiopalvelinratkaisu voi olla liian vaikea ylläpitää, jolloin yksi ratkaisu voisi olla käyttää GitLab-versionhallintatyökalun CI/CD-ratkaisuja. GitLabin käyttäminen myös CI/CD-ratkaisuihin voi selkeyttää määrittämiä. Lisäksi aikataulutettujen julkaisuputkien luominen ja versionhallinnan yhdistäminen julkaisuputkiin on saumatonta [50].

Koontien epäonnistumisen syy ei tullut palautteissa esille. Palautteissa ei myöskään kerrottu kovin tarkkaan käytännön seurauksia testiympäristöjen ongelmista. Palautteissa ei tullut esille yhtäkään ratkaisuehdotusta. Näiden syiden vuoksi voitaisiin luoda uusi mahdollistaja kehitysjonoon, jossa tutkitaan juurisyy

testiympäristön ongelmille, keksitään paras ratkaisu juurisyille ja toteutetaan ratkaisu, oli se sitten migraatio GitLab CI/CD -ratkaisuun tai nykyisen julkaisu-putken parannus [51]. Tätä varten kannattaa arkkitehtuurisen kiitotien (architectural runway) mukaisesti luoda uusi väliaikainen tiimi, jossa on järjestelmäarkkitehti tuoteomistajana, scrum master ja tiimin jäseniä esim. järjestelmätiimistä. [52.]

Kun testiympäristössä on tehty parannuksia, testiympäristö kannattaa dokumentoida huolella. Dokumentoinnin lähteinä CMMI:n prosessialue VAL:in mukaan on [40, s. 398]:

- Tuotteiden ja tuotekomponenttien vaatimukset testiympäristöltä
- standardit
- asiakkaiden vaatimuskriteerit
- testiympäristön suorituskyky
- testiympäristön suorituskyvyn sallittu vaihtelu.

Dokumentoinnin tuloksena voisi olla esimerkiksi validointimenettelyt, validointikriteerit sekä menetelmät testiympäristön ylläpitoon, perehdytykseen ja tukeen [40, s. 398].

Yllättävät muutokset

Epävarmuus ja yllättävät muutokset ovat yleisiä haasteita ohjelmistokehityksessä [53, s. 4–7]. Myös palautteissa yleinen tyytymättömyyden lähde oli yllättävät muutokset ja epävarmuudet. Epäarvostetut muutokset liittyivät joko yllättäviin tiimikokoonpanon tai kapasiteetin muutoksiin, joista tiimit eivät kokeneet saaneensa tarpeeksi nopeasti tietoa. On kuitenkin tapoja vähentää epävarmuuksien vaikutusta. Ohjelmiston vaatimuksia suunnitellessa käyttäjien kanssa kannattaa pitää yllä suoraa ja avointa kommunikaatiota [53, s. 24]. Suunnitelmassa täytyy aina ottaa huomioon muutoksien mahdollisuus. Tämä on yksi ketterän kehityksen tärkeimmistä periaatteista [53, s. 29–32].

Jossain aiheeseen liittyvistä palautteista tuli esille, että muutokset liittyivät kilpailutuksiin. Koska kilpailutukset lisäävät epävarmuutta, täytyisi niiden aikana tapahtuviin budjetointi- tai ominaisuusmuutoksiin varautua paremmin. Epävarmasti toteutuvat ominaisuudet voi tunnistaa esimerkiksi sen perusteella, minkä tyyppisiä ominaisuuksia jouduttiin aiemmin palautteiden antamisen aikaan perumaan suunnitelmasta tai millaisia ominaisuuksia jouduttiin lisäämään suunnitelmaan kilpailutusten tuloksien vuoksi. Jos tunnistetaan epävarmasti toteutuva ominaisuus, voi olla, ettei sen toteuttamista kannata aloittaa, ennen kuin kilpailutuksen tulos on selvillä. [53, s. 23–28.]

Tiimeille kannattaa suunnittelun aikana kertoa, mitkä ominaisuudet ovat epävarmoja. Siten riskit voidaan kirjata ylös PI-suunnitelmaan. Tiimin jäsenille kannattaa kertoa, keiltä ovat yhteishenkilöitä, joilta voi kysyä vaatimuksiin ja aikatauluihin liittyviä kysymyksiä. Tiimin jäsenien tulee olla herkästi yhteydessä näihin yhteishenkilöihin, jos he huomaavat, että jokin ominaisuus ei ole mahdollista toteuttaa annetussa aikataulussa ja tilanne pystytään selvittämään yhteistyössä asiakkaiden kanssa. [53, s. 23–28.]

Suunnitellut ominaisuudet eivät toteutuneet

Kolmas toistuva palautteiden aihe oli suunnitelmien puutteellinen toteutuminen huolimatta korkeasta luottamusarvosanasta PI-suunnittelussa. Yksi tapa parantaa tai hallita ongelmaa olisi ominaisuuksien pilkkominen pienemmiksi. Tiimin on helpompi suunnitella pienempiä tarinoita pienemmistä ominaisuuksista, sillä ne ovat helpommin hahmotettavissa. Kun ominaisuudet ovat pienempiä, ne ovat helpommin toteutettavissa yhden PI:n aikana. [54.]

Kehittäjien WIP-määrää kannattaa rajoittaa, jotta tiimin jäsenet eivät joudu vaihtamaan usein töiden välillä (context switching). Tämä vähentää tuottavuutta ja aiheuttaa töiden jäämistä kesken. [55.]

SAFen Tarkasta ja mukauta -tapahtuman tärkeimpiä tarkoituksia on tunnistaa ongelmia ja selvittää niiden juurisyy. Ongelmanratkaisutyöpajoissa täytyisikin

selvittää, miksi ominaisuuksia ei saatu valmiiksi, ja suorittaa ratkaisemiseen aivoriivi, jonka jälkeen parannukset lisätään junan kehitysjonoon. [16.]

Liikaa vikoja tuotantoympäristössä

Yksi tärkeimmistä tavoista vähentää vikoja (defects) on kirjoittaa mahdollisimman selkeitä hyväksymisvaatimuksia ja käyttäjätarinoita. Kun nämä ovat mahdollisimman selkeitä, hyväksymistestit on helppo kirjoittaa. Kun taas hyväksymistestit ovat selkeitä ja kattavia, viat voi huomata jo testiympäristössä ennen niiden päätymistä tuotantoon. [56.]

Sen jälkeen, kun vaatimukset ovat selkeitä, koodin katselmointiin kannattaa kiinnittää huomiota. Tekoäly voi löytää koodia katselmoidessa ongelmia, jotka olisi voineet jäädä ihmissilmältä huomaamatta. GitLabissa toimiva itseisännöity tuote on Codium PR-Agent [57]. Aiemmista testiympäristön tai tuotannon vioista voisi katsoa, onko vioilla yhteistä tekijää, kuten koskevatko ne samaa sovelluksen osaa, ja keskittyä tämän osan prosessien parantamiseen. PI43:n aikana kerättyyn palautteeseen oli myös kommentoitu, että hot fixeille eli normaalin julkaisuaikataulun ulkopuolella tehtäville kiireisille korjauksille voisi tehdä juurisyyanalyysin, jonka jälkeen kirjoitettaisiin parannustehtäviä.

Liiketoiminnan edustajat eivät osallistu Tarkasta ja mukauta -tapahtumaan

Liiketoiminnan edustajat ovat yksi oleellisimmista osallistujista Tarkasta ja mukauta -tapahtumaan. Liiketoiminnan edustajien tehtävä tapahtumassa on määrittää toteutuneen liikearvon osuus suunnitellusta liikearvosta. Liiketoiminnan edustaja voi saada tapahtumassa tietoa töiden etenemisestä ja auttaa poistamaan esteitä, jotka ovat tiimien hallinnan ulkopuolella. Lisäksi liiketoiminnan edustajan kuuluu osallistua mahdolliseen ongelmanratkaisutyöpajaan. Näistä syistä liiketoiminnan edustajien on suositeltavaa osallistua SAFen tapahtumiin. [16; 27]

Tapahtumat pitäisi olla englanniksi tai suomea osaamattomat täytyisi ottaa huomioon

Kommunikaatio on tärkeää ohjelmistokehityksessä. Varsinkin IT-alalla toimivassa suuressa kansainvälisessä yrityksessä kommunikointi täytyy usein, jopa pääasiassa, tehdä kansainväliseksi kieleksi muodostuneella englannin kielellä. Erinäisistä syistä kuitenkin kaikki eivät haluaisi puhua englantia. Syynä tähän on Aichhornin ja Puckin [58] mukaan usein vieraan kielen pelko. Kielitaito ei kuitenkaan parane välttelemällä vieraan kielen puhumista vaan vieraan kielen puhumisen välttely pitää yllä kielimuureja. Englannin kielen välttely ja kielen vaihtaminen äidinkielelle aiheuttaa haittaa tiedonjaolle ja yrityksen sisäiseen verkostoitumiseen. [58.]

Ihmisten pelkoa vieraan kielen puhumiselle vähentää tuomitsematon ilmapiiri. Göran Lindahl, ABB:n entinen toimitusjohtaja, sanoi MIT Sloan Management Review -lehden haastattelussa vuonna 2001, että heidän yrityksensä kieli on ”huono englantia”. Tällä Lindahl tarkoitti, ettei kenenkään tarvitse pelätä idean esittämistä, koska heidän englannin kielensä ei ole täydellinen. Tällaiset lausunnot voivat vähentää ihmisten pelkoa tai epämukavuutta vieraan kielen käyttämisessä. [58.]

6 Tuloksien arviointi ja pohdinta

Testiympäristöjen ongelmat olivat mainittu eniten Tarkasta ja mukauta -tapahtuman aikana kerätyissä palautteissa, sillä se esiintyi peräti kuudella eri PI:llä. Tästä voidaan päätellä, että ongelmaa pidetään merkittävänä, ja se on jatkunut pitkään. Testiympäristöjen jatkuva integraatio on isossa osassa ohjelmistokehitystä ja SAFeakin. Toimivien testiympäristöjen merkitys on erityisen iso sosiaali- ja terveydenhuollon ohjelmistokehityksessä, sillä rajapinnat ja data ovat erityisen monimutkaisia ja vaikeita jäljitellä paikallisessa ympäristössä. Tämän vuoksi testiympäristöjen ongelmat voivat helposti näissä töissä jopa estää työt täysin. Testiympäristö on monimutkainen kokoelma eri applikaatioita, joten ongelmien selvittäminen on luultavasti resursseja vievää. Pitkällä aikavälillä on

kuitenkin oletettavaa, että testiympäristöjen parantamiseen käytetty aika mak-saisi itsensä takaisin vähentyneinä hukattuina työtunteina.

Muita palautteiden kohteita, kuin testiympäristön ongelmia ei pidetty yhtä han-kalina korjata. Muutoksista tehdään vähemmän yllättäviä käyttämällä avointa kommunikaatiota asiakkaan ja kehittäjien välillä ja mahdollisuuksien mukaan keräämällä enemmän tietoa ennen ominaisuuksien suunnittelua. Ominaisuuksien pilkkominen pienemmiksi ja selkeämmiksi kokonaisuuksiksi helpottaa kir-joittamaan niistä helpommin selkeitä tarinoita ja tekee suunnitelmien ja vaati-musten muuttamisesta yksinkertaisempaa. Tuotantoympäristön viat arvioitiin vasta neljänneksi tärkeimmäksi, sillä aiempien palautekohteiden korjaamisen uskottiin auttavan vähentämään tuotantovikoja, ja tämä kohde oli mainittu vain kaksi kertaa.

Suorittamatta jääneitä tavoitteita ei CMMI-mallista löytynyt valituista prosessi-alueista. Kaikki muut tavoitteet saivat arvioinnissa arvosanan 3 (määritelty) paitsi vahvistusprosessialueen VER erityinen tavoite SP 1.2 ”perusta vahvistus-ympäristö” sai kyvykkyyssuokituksen 1 (suoritettu), sillä testiympäristöjen doku-mentointia pidettiin puutteellisena. Kaikki prosessialueiden arvioinnit ovat liit-teessä 1. Jatkotutkimuksena voisi tutkia eroja CMMI 1.3:n ja uusimman CMMI 3.0 -mallin välillä etenkin käsitellyissä VER, REQM, PP ja MA -prosessialueissa sekä mitä puutteita mallin noudattamisessa löytyisi käyttämällä tätä uusinta mal-lia.

Työmenetelmissä oli sekä hyviä että huonoja puolia. SAFen teorian tutkiminen oli hyödyllistä joidenkin ratkaisuehdotuksien kehittämiseen. CMMI-mallin kyp-syysalueet helpottivat palautteiden tärkeysjärjestyksen määrittämistä. Toisaalta jatkuvan kehittämisen mallin käytöstä saatu hyöty sekä palautteisiin liittyvien prosessialueiden arvioinnin hyöty oli vähäinen. Lähes kaikki käytännöt ja tavoit-teet täyttyivät, paitsi vahvistusympäristön perustamiseen liittyvä.

Valituissa prosessialueissa esiintyvät käytännöt ovat vakiintuneita ohjelmistoke-hityksen käytäntöjä. Hyöty prosessialueiden analysoinnille jatkuvan mallin

avulla voisi olla suurempi, jos parannuskohteena olisi uudempi projekti ja halutaan varmistaa, että kaikki hyödylliset käytännöt halutaan toteuttaa. Toisaalta CMMI-mallista löytyi VAL-prosessialueesta hyödyllisiä esimerkkejä testiympäristön dokumentointiin. Jos työ suoritettiin uudestaan, kannattaisi ottaa selvää uudemman CMMI-mallin käyttöönotosta. Työn kirjallisuustutkimustulokset ja tutkimusmenetelmät ovat hyödynnettävissä SAFesta, prosessienparannuksesta ja CMMI-mallin hyödyntämisestä kiinnostuneille.

Lähteet

- 1 CGI Yrityksenä. Verkkoaineisto. CGI Inc. <<https://www.cgi.com/fi/fi/cgi-yrityksena>>. Luettu 5.11.2024.
- 2 Lean Thinking and Practice. Verkkoaineisto. Lean Enterprise Institute. <<https://www.lean.org/lexicon-terms/lean-thinking-and-practice/>>. Luettu 14.6.2024.
- 3 SAFe 6.0. 2022. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/safe/>>. Päivitetty 13.3.2022. Luettu 24.7.2024.
- 4 Knaster, Richard & Leffingwell, Dean. 2018. SAFe 4.5 distilled: applying the Scaled Agile Framework for lean enterprises. E-kirja. O'Reilly Media.
- 5 SAFe Glossary. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/glossary?lang=fi>>. Luettu 7.6.2024.
- 6 Lean-Agile Mindset. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/lean-agile-mindset>>. Päivitetty 25.5.2023. Luettu 5.11.2024.
- 7 SAFe Lean-Agile Principles. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/safe-lean-agile-principles/>>. Päivitetty 4.5.2023.
- 8 Lean-Agile Leadership. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/lean-agile-leadership/>>. Päivitetty 27.5.2023. Luettu 5.11.2024.
- 9 Core Values. 2021. Verkkoaineisto. Scaled Agile Inc. <<https://v5.scaledagileframework.com/safe-core-values/>>. Päivitetty 10.2.2021. Luettu 7.6.2024
- 10 Core Values. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/safe-core-values/>>. Päivitetty 10.10.2023. Luettu 7.6.2024.
- 11 Agile Product Delivery. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/agile-product-delivery/>>. Päivitetty 11.10.2023. Luettu 7.6.2024.
- 12 System Demo. 2022. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/system-demo/>>. Päivitetty 6.12.2022. Luettu 14.6.2024.

- 13 Agile Release Train. 2022. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/agile-release-train>>. Päivitetty 24.10.2022. Luettu 7.6.2024.
- 14 Planning Interval. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/planning-interval/>>. Päivitetty 7.2.2023. Luettu 9.6.2024.
- 15 Innovation and Planning Iteration. 2022. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/innovation-and-planning-iteration/>>. Päivitetty 6.2.2022. Luettu 28.6.2024.
- 16 Inspect and Adapt. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/inspect-and-adapt/>>. Päivitetty 22.1.2023. Luettu 28.6.2024.
- 17 Lewis, Sarah. What is a fishbone diagram? Verkkoaineisto. Techtarget. <<https://www.techtarget.com/whatis/definition/fishbone-diagram>>. Luettu 14.7.2024.
- 18 5 Whys. Verkkoaineisto. Lean Enterprise Institute. <<https://www.lean.org/lexicon-terms/5-whys/>>. Luettu 14.7.2024.
- 19 Pareto Chart. Verkkoaineisto. Lean Enterprise Institute. <<https://www.lean.org/lexicon-terms/pareto-chart/>>. Luettu 15.7.2024.
- 20 What is a Sprint Retrospective? Verkkoaineisto. Scrum.org. <<https://www.scrum.org/resources/what-is-a-sprint-retrospective>>. Luettu 28.6.2024.
- 21 PI Planning. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/pi-planning/>>. 19.3.2023. Luettu 14.7.2024.
- 22 Release Train Engineer. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/release-train-engineer/>>. Päivitetty 12.10.2023. Luettu 7.6.2024.
- 23 ART and Solution Train Backlogs. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/art-and-solution-train-backlogs/>>. 9.10.2023. Luettu 23.7.2024.
- 24 Product Management. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/product-management/>>. Päivitetty 4.7.2023. Luettu 16.8.2024.
- 25 System Architect. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/system-architect/>>. Päivitetty 5.7.2023. Luettu 11.8.2024

- 26 Lean-Agile leadership. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/lean-agile-leadership/>>. Päivitetty 27.5.2023. Luettu 16.8.2024.
- 27 Business Owners. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/business-owners/>>. Päivitetty 3.7.2023. Luettu 16.8.2024.
- 28 Lean Portfolio Management. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/lean-portfolio-management/>>. Päivitetty 11.10.2023. Luettu 17.8.2024.
- 29 Agile Teams. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/agile-teams/>>. Päivitetty 22.8.2023. Luettu 20.8.2024.
- 30 Scrum Master/Team Coach. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/scrum-master-team-coach/>>. Päivitetty 12.10.2023. Luettu 20.8.2024.
- 31 Product Owner. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/product-owner/>>. Päivitetty 12.10.2023. Luettu 20.8.2024.
- 32 Järvinen, Matti. 2023. The Benefits and Challenges of Scaled Agile Framework in the IT Industry – Case Study: Company X. Master’s thesis. Arcada University of Applied Sciences. Theseus-tietokanta.
- 33 Implementation Roadmap. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/implementation-roadmap/>>. Päivitetty 4.5.2023. Luettu 31.7.2024.
- 34 Weighted Shortest Job First. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/wsjf/>>. Päivitetty 9.10.2023. Luettu 2.8.2024.
- 35 Putta, Abheeshta; Uludağ, Omer & Paasivaara, Maria. 2018 Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review. Product-Focused Software Process Improvement. PROFES 2018. Lecture Notes in Computer Science, vol 1127. s. 334–351.
- 36 Putta, Abheeshta; Uludağ, Omer & Paasivaara, Maria. 2021. Benefits and Challenges of Adopting SAFe – An Empirical Survey. Agile Processes in Software Engineering and Extreme Programming. XP 2021. Lecture Notes in Business Information Processing, vol 419. s. 172–187.

- 37 Ciancarini, Paolo; Kruglov, Artem; Pedrycz, Witod; Salikhov Dilshat & Succi, Giancarlo. 2022. Issues in the Adoption of the Scaled Agile Framework. 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). s. 175–184.
- 38 Denning, Stephen. 2015. Agile: it's time to put it to use to manage business complexity. *Strategy & Leadership*, vol. 43, no. 5. s. 10–17.
- 39 What is the difference between CMMI, CMMI-DEV, CMMI-SVC, CMMI-SPM (formerly CMMI-ACQ), People CMM, and DMM? 2020. Verkkoaineisto. ISACA. <<https://support.isaca.org/s/article/What-is-the-difference-between-CMMI-CMMI-DEV-CMMI-SVC-CMMI-SPM-formerly-CMMI-ACQ-People-CMM-and-DMM-1598331745508>>. Päivitetty 9.10.2020. Luettu 5.8.2024.
- 40 CMMI for Development, Version 1.3. 2010. Verkkoaineisto. Software Engineering Institute. <https://insights.sei.cmu.edu/documents/853/2010_005_001_15287.pdf>. Luettu 6.8.2024.
- 41 About ISACA. Verkkoaineisto. ISACA. <<https://cmmiinstitute.com/company>>. Luettu 25.7.2024.
- 42 New Update to CMMI Performance Solutions. Verkkoaineisto. ISACA. <<https://cmmiinstitute.com/products/cmml/content-release>>. Luettu 25.7.2024.
- 43 Chaudhary, Mukund & Abhishek Chopra. 2017. CMMI for Development. E-kirja. Apress.
- 44 Santana Furtado Soares, Felipe & de Lemos Meira, Silvio Romero. An Agile Strategy for Implementing CMMI Project Management Practices in Software Organizations. 2015 10th Iberian Conference on Information Systems and Technologies (CISTI).
- 45 Selleri Silva, Fernando; Santana Furtado Soares, Felipe; Lima Peres, Angela; Monteiro de Azevedo, Ivanildo; Paula L.F. Vasconcelos, Ana; Kenji Kamei, Fernando & de Lemos Meira, Silvio Romero. 2014. Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, Volume 58. s. 20–43.
- 46 Astridita, Anggia; Raharjo, Teguh & Fitriani, Anita Nur. 2024. Perceived Benefits and Challenges of Implementing CMMI on Agile Project Management: A Systematic Literature Review. *International Journal of Advanced Computer Science & Applications*, Vol. 15, No. 1. s. 232–239.
- 47 Handbook for Conducting Standard CMMI Appraisal Method for Process Improvement (SCAMPI) B and C Appraisals, Version 1.1. 2005. Verkkoaineisto. Software Engineering Institute. <https://insights.sei.cmu.edu/documents/1612/2005_002_001_14438.pdf>. 2005. Luettu 7.9.2024.

- 48 Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document. 2011. Verkkoaineisto. Software Engineering Institute. <https://isights.sei.cmu.edu/documents/1618/2011_002_001_15311.pdf>. 2011. Luettu 7.9.2024.
- 49 Turnbull, James. 2016. The Art of Monitoring. E-kirja. O'Reilly Media.
- 50 Levan, Michael. 2020. 5 advantages of GitLab CI/CD pipelines. Verkkoaineisto. <<https://www.techtarget.com/searchsoftwarequality/video/5-advantages-of-GitLab-CI-CD-pipelines>>. 2020. Luettu 8.9.2024.
- 51 Enablers. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/enablers/>>. Päivitetty 13.10.2023. Luettu 14.9.2024.
- 52 Architectural Runway. 2023. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/architectural-runway/>>. Päivitetty 9.1.2023. Luettu 14.9.2024.
- 53 Ylikotila, Arttu. 2023. Uncertainty in software development – a threat and a possibility. M.Sc. Thesis. University of Tampere, Faculty of Technology and Communication Sciences. Trepo-arkisto.
- 54 Spence, Ian. Right-Sizing Features for SAFe Planning Intervals. Verkkoaineisto. Scaled Agile Inc. <<https://scaledagileframework.com/right-sizing-features-for-safe-program-increments/>>. Luettu 22.9.2024.
- 55 Radigan, Dan. Putting the 'flow' back in workflow with WIP limits. Verkkoaineisto. Atlassian. <<https://www.atlassian.com/agile/kanban/wip-limits>>. Luettu 22.9.2024.
- 56 Cohn, Mike. 2004. User Stories Applied: For Agile Software Development. E-kirja. Addison-Wesley Professional.
- 57 Data Privacy. Verkkoaineisto. CodiumAI. <https://pr-agent-docs.codium.ai/overview/data_privacy/>. Luettu 27.9.2024.
- 58 Aichhorn, Nathalie & Puck, Jonas. 2017. "I just don't feel comfortable speaking English": Foreign language anxiety as a catalyst for spoken-language barriers in MNCs. International Business Review, vol. 26, no. 4. s. 749-763.

Prosessialueiden kyvykkyyssluokitukset

Prosessialue	SP 1.1	SP 1.2	SP 1.3	SP 1.4	SP 1.5	SG 1	SP 2.1	SP 2.2	SP 2.3	SP 2.4	SP 2.5	SP 2.6	SP 2.7	SG 2	SP 3.1	SP 3.2	SP 3.3	SG 3	Prosessialueen kyvykkyyss	
Verification (VER)	3	1	3			1	3	3	3					3	3	3		3	1	
Requirements Management (REQM)	3	3	3	3	3	3	3	3	3					3	3	3		3	3	
Project Planning (PP)	3	3	3	3		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Measurement and Analysis (MA)	3	3	3	3		3	3	3	3	3				3					3	

