



Scrum - Teorian ja käytännön kuilu

Tuomas Väkiparta

OPINNÄYTETYÖ
Elokuu 2024

Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

VÄKIPARTA, TUOMAS:
Scrum – Teorian ja käytännön kuilu

Opinnäytetyö 64 sivua
Elokuu 2024

Opinnäytetyössä pyrittiin selvittämään, miten käytetyimmän ketterän kehityksen menetelmän, Scrumin, teoriasta poiketaan työelämän todellisissa Scrum-projekteissa. Opinnäytetyön tavoitteena oli kirkastaa Scrumin teoreettisen lähtökohdan ja työelämän sovelluksien kuilua näkyvämmäksi.

Opinnäytetyön tarkoituksena oli selvittää Scrumin teorian ja käytännön eroja kuvailevan kirjallisuuskatsauksen, työelämän Scrum-projektin seurannan pohjalta tehtyjen havaintojen sekä viiden työelämän edustajan haastattelujen avulla. Työelämän projektin havainnointi ja haastattelut tehtiin suomalaisessa, keski-suudessa IT-konsultointiyrityksessä keväällä ja kesällä 2024. Teoriaa ja käytäntöä vertaillaan tutkielmassa Scrumin rooli-, tapahtuma- ja tuotostasoilla.

Opinnäytetyön tuloksien pohjalta vaikuttaisi siltä, että työelämässä Scrumin teoriasta poiketaan usein. Scrumin roolit muun muassa jäävät toisinaan kokonaan jakamatta, projekteissa voi olla liian monta saman roolin edustajaa tai sitten vastuut menevät roolien välillä ristiin. Toisaalta Scrumin tapahtumat eivät aina noudata ohjepituuksia, eivätkä toteudu Scrumin osoittamalla tavalla, jos yli-päättään toteutuvat. Lisäksi myös tuotoksien kohdalla toisinaan poiketaan teoriasta: kehitysjonot eivät ole aina käytössä, kehitysjonot voivat olla heikosti priorisoituja ja sidottuja tuotteen tavoitteeseen tai työmääräarviot saattavat puuttua kokonaan.

Työelämässä Scrumia jalkauttavat Scrum masterit voivat hyötyä opinnäytetyön tuloksista tiedostaessaan paremmin, millaisista lähtökohdista Scrum-projektin jäsenet mahdollisesti tulevat Scrum-projektiin. Toisaalta myös Scrum-projekteihin uutena tulevat voivat saada tutkielman tuloksista hyvän katsauksen mahdollisiin eroihin.

Opinnäytetyön havaintoja voidaan käyttää pohjana mahdollisessa jatko-tutkimuksessa selvittämään, miksi opinnäytetyössä löydettyjä poikkeamia esiintyy ja ovatko poikkeamat soveltajien kannalta edullisia vai haitallisia. Tämä tieto puolestaan voi auttaa joko Scrumin teoriaa mukautumaan paremmin työelämän tarpeisiin tai työelämän soveltajia korjaamaan toimintaansa vastaamaan paremmin teoriaa.

Asiasanat: scrum, rooli, tapahtuma, tuotos

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

VÄKIPARTA, TUOMAS:
Scrum – The Gap Between Theory and Practice

Bachelor's thesis 64 pages
August 2024

Scrum is the most widely used agile development method in the IT industry as of 2024. However, there are as many variations of Scrum as there are practitioners, but it is uncertain whether these deviations are justifiable.

The purpose of this bachelor's thesis was to shed light on these variations between the Scrum theory and practice. These findings, in turn, can serve as a basis for enhancing the Scrum theory and help practitioners better align their practices with the theory.

The topic of this thesis was researched through a descriptive literature review on the topic, supported by observations made from shadowing a real-life Scrum project and interviews with five representatives from the IT industry. The project shadowing and interviews were conducted in a mid-sized Finnish IT consulting company in the spring and summer of 2024.

Based on the findings of this thesis, deviations from the Scrum events, roles, and artifacts are quite prevalent in many applications of Scrum. Some of these variations might be well-founded due to budget and time constraints whereas others may simply be blatant abuse of the Scrum methodology without justification. A more detailed cost-benefit analysis of these variations, however, requires further research on the topic.

Key words: scrum, role, event, artefact

SISÄLLYS

1	JOHDANTO	6
2	MENETELMÄT JA AINEISTOT	8
3	SCRUM – ERÄS KETTERÄN KEHITTÄMISEN MENETELMÄ.....	9
	3.1 Scrumin taustat.....	9
	3.2 Scrumin teorian lyhyt kuvaus	10
	3.3 Scrum-viitekehyksen soveltaminen taustayrityksessä	13
4	SCRUMIN TEORIAN JA KÄYTÄNNÖN MAHDOLLISET EROT	16
	4.1 Tutkimuskysymykset	16
	4.2 Hypoteesit	16
5	SCRUMIN TEORIAN JA KÄYTÄNNÖN HAVAITTUJA EROAVAISUUKSIA.....	18
	5.1 Erot rooleissa.....	18
	5.1.1 Kehittäjät ja Scrum-tiimin koko	19
	5.1.2 Tuoteomistaja	23
	5.1.3 Scrum master	27
	5.2 Erot tapahtumissa	31
	5.2.1 Sprintti.....	33
	5.2.2 Sprintin suunnittelu.....	36
	5.2.3 Päivittäispalaveri	38
	5.2.4 Sprintin katselmointi	42
	5.2.5 Sprintin retrospektiivi	44
	5.3 Erot tuotoksissa	45
	5.3.1 Tuotteen kehitysjono	46
	5.3.2 Sprintin kehitysjono	49
	5.3.3 Inkrementit ja yksittäiset itemit.....	52
6	TULOKSET JA ANALYSOINTI	56
7	POHDINTA	61
	7.1 Tutkimuksen rajoitteet ja heikkoudet.....	61
	7.2 Mahdollinen jatkotutkimus.....	62
	LÄHTEET	63

LYHENTEET JA TERMIT

Business expert	Liiketoiminnan asiantuntija taustayrityksen asiakkaan puolella. Tukee tuoteomistajaa käyttäjätarinoiden luonnissa.
CSM	"Customer Success Manager" eli asiakkuudesta vastaava päällikkö taustayrityksen projekteissa
Jira	Atlassianin tarjoama projektihallinnan työkalu, jota käytettiin taustayrityksessä.
Lead consultant	Taustayrityksessä käytetty titteli projekteissa päävastuussa olevasta konsultista. Kts. Salesforce consultant.
Lead developer	Taustayrityksessä käytetty titteli projektissa päävastuussa olevasta ohjelmistokehittäjästä. Kts. Salesforce developer.
Solution architect	Ratkaisuarkkitehti taustayrityksen asiakkaan puolelta. Vastaa integraatioarkkitehtuurista ja tämän yhteensopivuudesta Salesforce-arkkitehtuurin kanssa.
Salesforce architect	Salesforce-arkkitehti taustayrityksessä. Vastaa Salesforceen päälle rakennettavan kehityksen arkkitehtuurista ja parhaiden käytänteiden noudattamisesta Salesforce-kehityksessä.
Salesforce consultant	Salesforce-konsultti taustayrityksessä. Vastaa Salesforceen päälle tehtävästä designista ja konfiguraatioista käytännöntasolla.
Salesforce developer	Salesforce-kehittäjä taustayrityksessä. Vastaa Salesforce-alustalle tehtävistä räätälöidyistä ratkaisuista.

1 JOHDANTO

Scrum on yksi monista ketterän ohjelmistokehityksen projektimenetelmistä. Scrumin juuret ovat 1986 Harvard Business Review -lehden artikkelissa, jossa Hirotaka Takeuchi ja Ikujiro Nonaka ensimmäisinä esittivät termin viitaten rugbyjoukkueen saumattomaan ja ketterään yhteispeliin. Tästä muutama vuosi myöhemmin Jeff Sutherland ja Ken Schwaber jalostivat mallin todellisen viitekehityksen, jota he ovat tähän päivään asti kehittäneet vastaamaan paremmin työelämän tarpeita.

Digital.ai:n vuoden 2023 ”17th State of Agile” -kyselytutkimuksen raportista käy ilmi, että Scrum on pitänyt pintansa käytetyimpänä ketterän kehityksen menetelmänä jo vuodesta 2006 lähtien, kun ensimmäinen vastaa tutkimus teetettiin (Digital.ai 2023, 11). Käytetyimpänä ketteränä menetelmänä voisi olla perusteltua selvittää, kuinka viitekehystä toteutetaan käytännössä. Yhtenä motivaationa selvitykselle voidaan pitää Sutherlandin ja Schwaberin julkaiseman virallisen Scrum-oppaan loppusanoissa mainitsemaa varoitusta (2020, 13), jossa todetaan että ”Vaikka Scrumin osittainen toteuttaminen onkin mahdollista, se ei ole Scrumia. Scrum on olemassa vain kokonaisuutena”.

Työn tavoitteena on tehdä näkyväksi Scrumin teorian ja käytännön eroja tämänhetkisessä työelämässä. Löydettyjen erojen pohjalta olisi kenties mahdollista selvittää erojen hyötyjä ja haittoja, minkä pohjalta Scrumin teoriaa voitaisiin tarvittaessa korjata vastaamaan paremmin työelämän tarpeita tai päinvastoin työelämän soveltajat voisivat ohjata toimintaansa lähemmäksi teoriaa. Tämän tutkielman tuloksista voivat hyötyä niin taustayrityksen Scrum-projektimallin kanssa tekemisissä olevat työntekijät kuin yleisestikin ohjelmistoalalla ketterien projektien parissa työskentelevät. Etenkin Scrumin teorian jalkauttamisesta käytäntöön vastaavat Scrum masterit voivat hyötyä mahdollisista löydöksistä sovittaessaan teoriaa organisaatioonsa. Edellisten lisäksi myös ohjelmistoalalle uutena tulevat, esimerkiksi vastavalmistuneet opiskelijat, voivat hyödyntää tutkielman löydöksiä helpottamaan integroitumistaan työelämään.

Opinnäytetyön tarkoituksena on selvittää kolmella tavalla, miten Scrumin teoriasta mahdollisesti poiketaan työelämän ohjelmistoprojekteissa. Ensimmäkin opinnäytetyötä varten selvitetään kirjallisuuskatsauksen avulla, millaista aiempaa tutkimusta aiheesta on tehty ja millaisia mahdollisia eroja näissä tutkimuksissa on havaittu. Työelämälähtöisyyttä opinnäytetyöhön tuo yhden keskisuuren, suomalaisen IT-konsultointialan yrityksen Scrumilla toteutetun asiakasprojektin seurannan pohjalta tehdyt havainnot keväältä ja kesältä 2024. Kirjallisuuskatsauksen ja asiakasprojektin havainnoinnin lisäksi opinnäytetyötä varten on haastateltu viittä edellä mainitun taustayrityksen työntekijää Scrumin teoriaan liittyen.

Taustayritys on Salesforce-konsultointiin erikoistunut keskisuuri suomalainen yritys. Salesforce on markkinaosuudeltaan maailman suurin CRM-ratkaisujen palveluntarjoaja maailmassa (Salesforce 2024). Salesforce tarjoaa SaaS-ratkaisuja lisenssipohjaisesti ja opinnäytetyön taustayritys toimii Salesforceen partneriyrityksenä tarjoten Salesforcea käyttäville asiakasyrityksille tukea ympäristöjen käyttöönotoissa, ylläpidossa ja kehitystyössä. Asiakasprojektien ketterissä toteutuksissa taustayrityksessä on käytössään ensisijaisesti kaksi projektituotannon mallia, kanban ja Scrum.

2 MENETELMÄT JA AINEISTOT

Opinnäytetyötä varten tehtiin kuvaileva kirjallisuuskatsaus, jonka avulla selvitettiin aiempaa tutkimusta aiheeseen liittyen. Kirjallisuuskatsauksen tutkimukset olivat pitkälti toteutettu joko suuremman otoksen kyselytutkimuksina, esimerkiksi webkyselyinä, tai sitten yhden tai useamman yrityksen sisällä tehtyinä haastatteluina tai Scrum-tiimien seurantajaksoina case-tyyppisesti.

Kirjallisuuskatsauksen lisäksi opinnäytetyötä varten seurattiin yhtä opinnäytetyön taustayrityksen asiakkuutta, jota hallinnoitiin Scrumia soveltaen. Seurattu taustayrityksen asiakas on suomalainen, kuudesta tytäryhtiöstä koostuva teollisuus-eristys- ja telinekonserni. Asiakas oli ostanut seuratulta taustayritykseltä konsultointi- ja kehityspalveluita parantaakseen omaa Salesforce-ympäristöään, jonka konfigurointi oli tätä ennen ollut pitkälti yhden asiakasyrityksessä toimineen henkilön vastuulla. Yrityksen konsernirakenne ja pitkälle kehittyneet toimitusprosessit tekivät ympäristöstä kohtalaisen monimutkaisen, minkä vuoksi suuremmat hankkeet vaativat yhden henkilön lisäksi suurempaa kehitysbudjettia. Tässä kohtaa opinnäytetyön taustayritys tuli mukaan kehitystyöhön.

Kevään ja kesän 2024 aikana opinnäytetyötä varten havainnoitiin sekä taustayrityksen sisäisiä että asiakasyrityksen kanssa käytyjä ulkoisia palaverieja. Tänä aikana yrityksen Salesforce-ympäristöön toteutettiin Scrumia hyödyntäen sprinttimuotoisesti kolmea eri lopputuotetta erikseen ja osin päällekkäin. Seurannan aikana työn alla olivat työsuunnittelu-, tuntikirjaus- ja tarjoushallintakomponentit. Opinnäytetyössä havainnoinnin painopiste säilytettiin itse Scrum-prosessissa.

Scrumin teorian ja käytännön erojen selvittämiseksi taustayrityksestä otettiin myös viisi Scrum-projekteissa mukana ollutta työelämän edustajaa. Haastattelut tehtiin pääsääntöisesti yrityksen kotitoimistolla, mutta kaksi haastattelusta tehtiin etäyhteydellä. Haastateltavilta kysyttiin aluksi tausoittavia kysymyksiä, joilla kartoitettiin haastateltavien historiaa IT-alalta. Tämän jälkeen haastateltavilta kysyttiin aihepiireittäin heidän käytännön kokemuksiinsa Scrum-projektien roolituksista, tapahtumista ja tuotoksista. Kaikista haastattelusta on tehty muistiinpanot. Lisäksi neljä viidestä haastattelusta on nauhoitettu ja litteroitu.

3 SCRUM – ERÄS KETTERÄN KEHITTÄMISEN MENETELMÄ

3.1 Scrumin taustat

Scrum on lähtöisin vuonna 1986 julkaistusta Harvard Business Review -lehden artikkelista ”The New New Product Development Game”, jossa professorit Hirotaka Takeuchi ja Ikujiro Nonaka esittelivät ensimmäistä kertaa Scrumin ajatuksen (Salimi, n.d). Tämän ketterän mallin tavoitteena oli vastata vaiheittaisen tekemisen tuomiin ongelmiin, jotka olivat yhä näkyvämpiä maailmassa, jossa yrityksiltä vaaditaan korkean laadun, matalien tuotantokustannuksien ja tuotedifferentiaation lisäksi nopeutta tuoda uusia tuotteita markkinoille. (Takeuchi & Nonaka 1986.)

Nopeasti muuttuvassa ja digitalisoituvassa ympäristössä vesiputousmallin kaltaiset staattiset projektimallit eivät enää vaikuttaneet riittävän. ”Scrum” terminä on lainattu rugbyista. Nimi viittaa Takeuchin ja Nonakan mallin visioon ketterästä rugbyjoukkueesta, joka etenee kohti tavoitettaan koko kentän laajuudella vuorotellen pallonhaltijaa. Takeuchi ja Nonaka viittaavat artikkelissaan perinteisempään, vaiheittaiseen tekemiseen ”viestijuoksuna”, jossa tuotteen kehitys tapahtuu vaiheittain funktio kerrallaan.

Artikkeliaan varten Takeuchi ja Nonaka seurasivat monikansallisia yrityksiä, kuten Fuji-Xeroxia, Canonia, Hondaa, NEC:tä, Epsonia, Brotheria, 3M:ää, Xeroxia ja Hewlett-Packardia. Esimerkkiyrityksiä seuratessaan Takeuchi ja Nonaka havaitsivat, että näissä yrityksissä oli vaiheittaisen, ylhäältäpäin johdetun tuotekehitysmallin sijasta käytössä päällekkäin toimivia, käsin poimittuja asiantuntijatiimejä. Näillä ketterillä työryhmillä oli tarjottu vapaus ratkaista ongelmia parhaaksi näkemällään tavalla, kustannus- ja aikarajoitteet huomioiden. Takeuchi ja Nonaka eivät näin olleen siis vielä luoneet lopullista Scrum-viitekehystä, jota tänä päivänä monissa ohjelmistoprojekteissa käytetään, vaan ennemminkin esittelivät termin ja joukon havaintoja, jotka yhdistivät ketterästi toimivia yrityksiä.

Vuonna 1993 biometriikan tohtori Jeff Sutherland sovelsi ja kehitti Scrum-prosessia taustayrityksessään Easel Corporationissa yhdistäen Takeuchin ja Nonakan havaintoja olio-ohjelmoinnin, empiirisen prosessin hallinnan, iteratiivisen kehityksen teorioihin yhdessä tuottavuutta tutkineiden tutkimuksien tuloksiin. Tästä kaksi vuotta myöhemmin ohjelmistoalalla toiminut kehittäjä Ken Schwaber tuotti ensimmäisen Scrumia käsitelleen julkaisunsa keskusteltuaan pitkään Sutherlandin kanssa aiheesta. Tästä alkaen Schwaber ja Sutherland ovat säännöllisesti julkaisseet Scrumia käsitteleviä artikkeleita ja teoksia, joista kenties tunnetuimpia on useammalle kielelle käännetty ”Scrum-opas”. (Salimi, n.d.) Täten varsinainen Scrum-viitekehys syntyi Schwaberin ja Sutherlandin toimesta soveltaen Takeuchin ja Nonakan ajatuksia ja havaintoja.

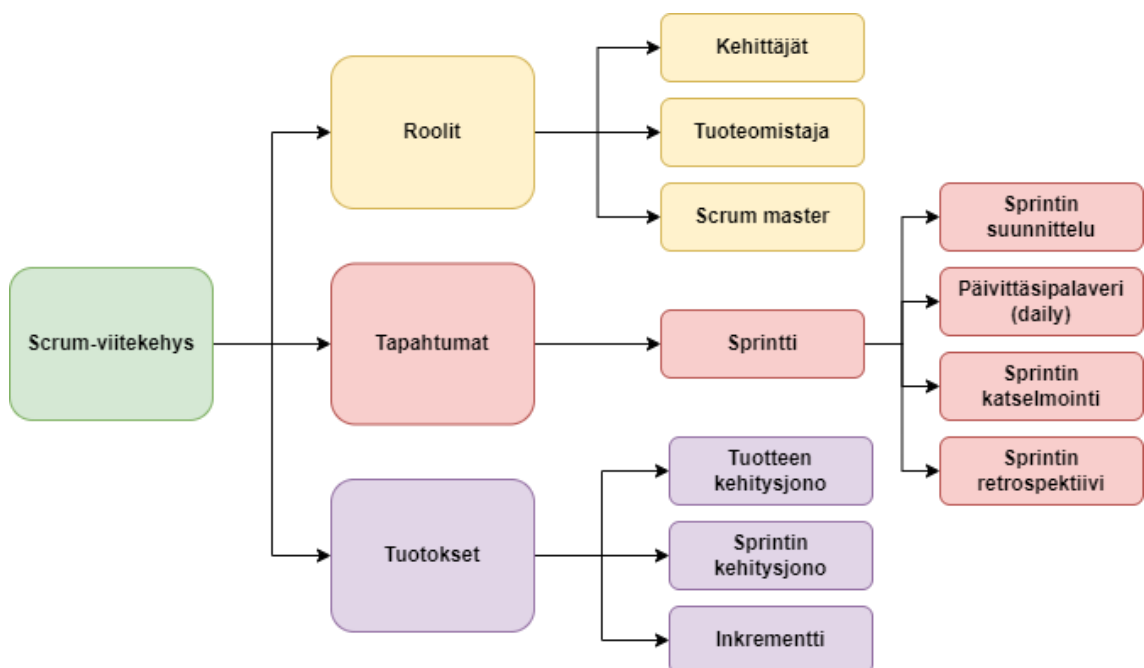
Schwaberin ja Sutherlandin julkaisema Scrumin määrittelevä Scrum-opas julkaistiin ensimmäisen kerran vuonna 2010, minkä jälkeen opasta on säännöllisesti päivitetty vastaamaan paremmin käytännön tarpeita (Sutherland & Schwaber 2020, 1). Yhdessä Pete Deemerin, Gabrielle Benefieldin, Craig Larmanin ja Bas Vodden luoman ”Scrum Primer” -pohjamateriaalin kanssa Scrum-opas on yleisesti tunnistettu tutkimuksissa Scrumin teorian perustavanlaatuisiksi pohjaksi (Masood, Hoda & Blincoe 2022, 1580). Myös tässä tutkielmassa Scrumin teorian pääasiallisina lähteinä toimivat Schwaberin ja Sutherlandin vuonna 2020 päivittämä Scrum-opas sekä Deemerin ym. vuonna 2012 julkaisema toinen versio Scrum Primer -oppaasta.

3.2 Scrumin teorian lyhyt kuvaus

Scrum on tarkoituksella luotu kevyeksi viitekehyyksi, jonka toteutukseen käyttäjille on jätetty paljon vapauksia, sillä malli luottaa ennen kaikkea soveltajien kollektiiviseen älyyn ja kokemuksiin (Sutherland & Schwaber 2020, 3). Tämä saattaakin osittain selittää Scrumin suosiota ketterissä projekteissa, sillä viitekehys tarjoaa laveat raamit, joiden ympärille projektitiimi voidaan rakentaa vaihtelevissakin organisaatiokulttuureissa. Tässä voi myös piillä Scrumin yksi haasteista, sillä opas ei tarjoa kuitenkaan konkreettisia askeleita projektin hallintaan esimerkiksi tehtävien erittelyn ja estimoinnin, priorisoinnin, sprintin tavoitteen määrittelemisen, tehtävien jalostamisen tai työnjaon suhteen (Masood ym. 2022, 1579). Scrum-viitekehyyksessä ei annakaan yksityiskohtaisia ohjeita

sääntöjä projektin toteutuksen suhteen, vaan viitekehys ennen kaikkea ohjaa tiimin yhteistä toimintaa ja vuorovaikutusta (Sutherland & Schwaber 2020, 3).

Scrumin taustateoria voidaan jakaa karkeasti kolmeen kattokäsitteeseen, joita ovat roolit, tapahtumat ja tuotokset. Tätä havainnollistaa opinnäytetyötä varten laatimani kuvan 1 mukainen kaavio, joka noudattelee pitkälti Sutherlandin ja Schwaberin Scrum-oppaan kappalerakenteen mukaista kuvausta Scrumin teoriasta. Tässä opinnäytetyössä Scrumin teorian ja käytännön vertailu tullaan tekemään kuvan 1 mukaista runkoa noudattaen aloittaen rooleista, siirtyen tapahtumiin ja lopuksi vertaillen tuotoksia. Kuvassa 1 näkyviä käsitteitä avataan tarkemmin vielä jokaisen vertailevan kappaleen alussa.

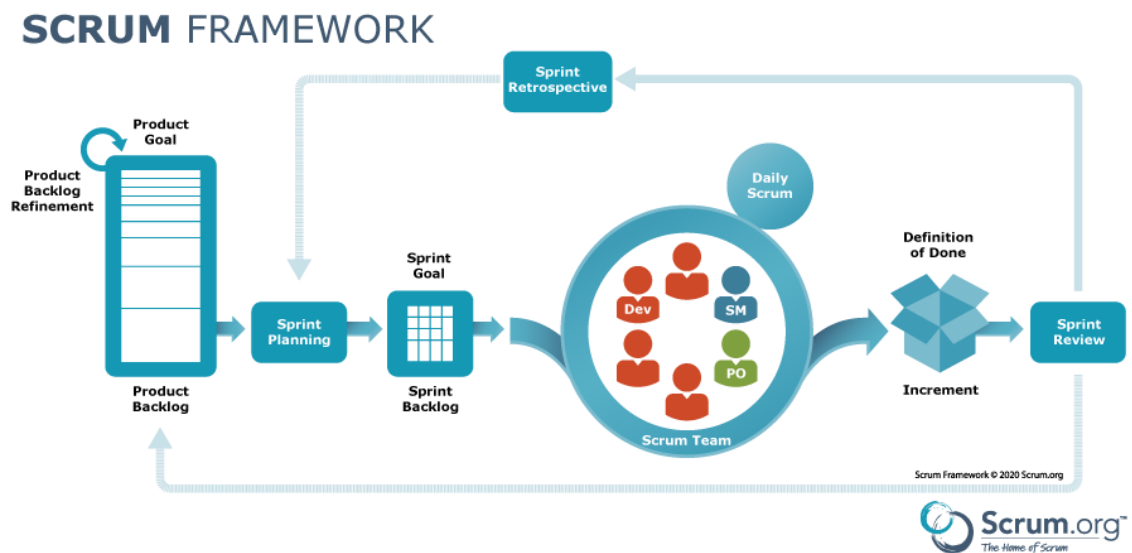


Kuva 1. Eräs hahmotelma Scrumista noudatellen Sutherlandin ja Schwaberin vuoden 2020 Scrum-oppaan rakennetta.

Rooleja Scrumissa on kolme: kehittäjä, tuoteomistaja ja Scrum master. Kehittäjällä ei nimestään huolimatta tarkoiteta pelkästään ohjelmistokehittäjiä, vaan nimitys on lähinnä peruja Scrumin ohjelmistoalataustasta. Kehittäjät voidaan nähdä Scrum-projektin tuottavana voimana, jotka vastaavat lopputuotoksen valmiiksi saattamisesta. Kehittäjien vastuulla on sprintin suunnitelman laatiminen sprintin kehitysjonosta, laadukkaan työn tekeminen valmiin määritelmään nojautuen ja omien suunnitelmien sopeuttaminen sprintin tavoitteeseen pääsemiseksi (Sutherland & Schwaber 2020, 5). Tuoteomistajan

roolissa toimiva yksi henkilö vastaa puolestaan siitä, että Scrum-tiimin tekemisestä saadaan irti suurin mahdollinen lisäarvo viestimällä tuotteen tavoitteet selkeästi ja läpinäkyvästi (Sutherland & Schwaber 2020, 6). Kehittäjien ja tuoteomistajan tukena toimii Scrum master, jonka vastuulla on huolehtia Scrum-tiimiä hidastavien esteiden poistamisesta, työrauhan säilymisestä ja kaikkien Scrumin tapahtumien toteutumisesta (Sutherland & Schwaber 2020, 6).

Roolituksien lisäksi Scrum-opas tarjoa viitekehyykselle viisi tapahtumaa, jotka ovat sprintti, sprintin suunnittelu, päivittäispalaveri, sprintin katselmointi ja sprintin retrospektiivi. Kuvassa 1 muut tapahtumat ovat hierarkkisesti sprinttiä alempana, sillä sprintti itsessään tapahtuma, jonka ympärille muut tapahtumat rakentuvat (Sutherland & Schwaber 2020, 3). Sutherland ja Schwaber toteavat (2020, 7), että ”Sprintit ovat Scrumin pulssi”, mikä useimmissa sprinttiä kuvaavissa kaavioissa näkyekin sprintin keskeisenä asemointina, kuten kuvassa 2.



Kuva 2. Eräs kuvaus Scrum-prosessin rakenteesta lainattuna Ken Schwaberin perustamalta Scrum.org -sivustolta. Kuvassa toimijoiden roolit on lyhennetty tittelein ”Dev”, ”SM” ja ”PO”, jotka ovat suomeksi kehittäjä, Scrum master ja tuoteomistaja vastaavasti.

Kuvassa 2 yhtä sprinttiä kuvastaa keskellä näkyvä kehä, jonka sisällä aiemmin esitellyt roolit ovat. Sprintti on maksimissaan kuukauden mittainen pyrhähdys, jonka aikana toteutetaan kaikki tarpeellinen tuotteen tavoitteen saavuttamiseksi (Sutherland & Schwaber 2020, 7). Onnistuakseen sprintti vaatii kuitenkin pohjalle sprintin suunnittelutapahtuman, jonka aikana tiimi määrittelee, miksi sprintti

toteutetaan, mitä sprintin aikana tehdään ja miten tavoitteeseen päästään (Sutherland & Scwhaber 2020, 7). Kuvassa 2 nähdään myös sprintin aikana pidettävät päivittäispalaverit. Päivittäispalaveri on sprintin aikana joka päivä pidettävä 15 minuutin mittainen kehittäjille varattu kokous, jossa kehittäjät jakavat senhetkisen statusensa työn etenemisestä ja mahdollisista työn etenemisen esteistä (Sutherland & Schwaber 2020, 9). Sprintin päätyttyä toiseksi viimeinen tapahtuma on sprintin katselmointi, jossa Scrum-tiimi esittelee keskeisille sidosryhmille, mitä sprintin aikana saatiin valmiiksi, mitä mahdollisia muutoksia tuotteen toimintaympäristössä on tapahtunut ja miten toimintaa tulisi mahdollisesti mukauttaa (Sutherland & Schwaber 2020, 9). Viimeisenä tapahtumana osana sprinttiä on sprintin retrospektiivi, jossa Scrum-tiimi sisäisesti tarkastelee omia prosessejaan ja tekemistään tarkoituksenaan kehittää yhteisiä toimintatapojaan (Sutherland & Schwaber 2020, 10). Viime kädessä Scrum-viitekehyksen tehtävänä on roolituksien ja tapahtumien kautta maksimoida tiimin tuottavuutta, jotta tehdystä työstä saataisiin mahdollisimman suuri lisäarvo.

Tämä lisäarvo konkretisoituu Scrumin tuotoksissa eli artefakteissa, joita ovat tuotteen kehitysjojo, sprintin kehitysjojo sekä inkrementti. Tuotokset ovat työn ja luodun arvon ilmentymiä (Sutherland & Schwaber 2020, 10). Karkeimpana tuotoksen tasona on tuotteen kehitysjojo. Tuotteen kehitysjojo vähitellen järjestettävä ja jalostuva lista kaikesta työstä, joka mahdollisesti toteutetaan tuotteen kehittämiseksi (Sutherland & Schwaber 2020, 11). Tuotteen kehitysjojonosta poimitaan aina kullekin sprintille tehtävät työt Scrumin toiseen tuotoksen tasoon, sprintin kehitysjojnoon, joka kuvaa sprintin aikana tehtäviä töitä. Sprintin kehitysjojonon tarkentuu sprintin tekemisen aikana ja sen tulisi olla tarpeeksi konkreettinen, jotta sprintin aikana syntyvän lisäarvon seuraaminen olisi mahdollista (Sutherland & Schwaber 2020, 11). Yksittäinen inkrementti on selkeä, valmiin määritelmää noudattava askel kohti sprintin tavoitteita (Sutherland & Schwaber 2020, 12).

3.3 Scrum-viitekehyksen soveltaminen taustayrityksessä

Kuvassa 3 on taustayrityksen luvalla lainattu kuvankaappaus opetusmateriaaleista, joissa yrityksen uusille työntekijöille esitellään yrityksessä käytetyt toimitusmallit: kanban ja Scrum. Kyseisellä kuvan 3 kalvolla lyhyesti esitellään ja

motivoidaan Scrumin käyttö. Vastaavanlainen lyhyt esitys Scrumista löytyy myös kuvasta 4, joka on ollut osa yrityksen vanhaa, asiakasodotuksien hallintaa käsittelevää koulutuspakettia. Yrityksen työntekijöiden lisäksi on tärkeää, että myös yrityksen asiakkailla on selkeä käsitys yrityksen toimintatavoista. Näin jokaisella ketterässä projektissa mukana olevalle on selvää, missä tahdissa, keiden toimesta ja miten tuote toimitetaan.

Operating Model - Scrum

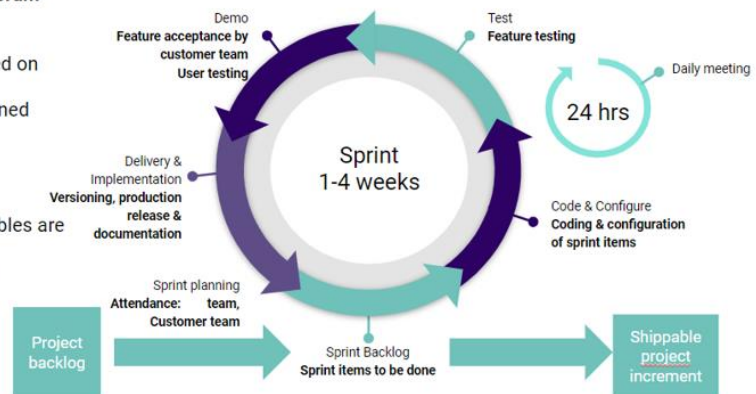
Project will be implemented with Agile Scrum

Scrum model:

- User stories and tasks are managed on the backlog
- Each sprint is time boxed and planned before implementation

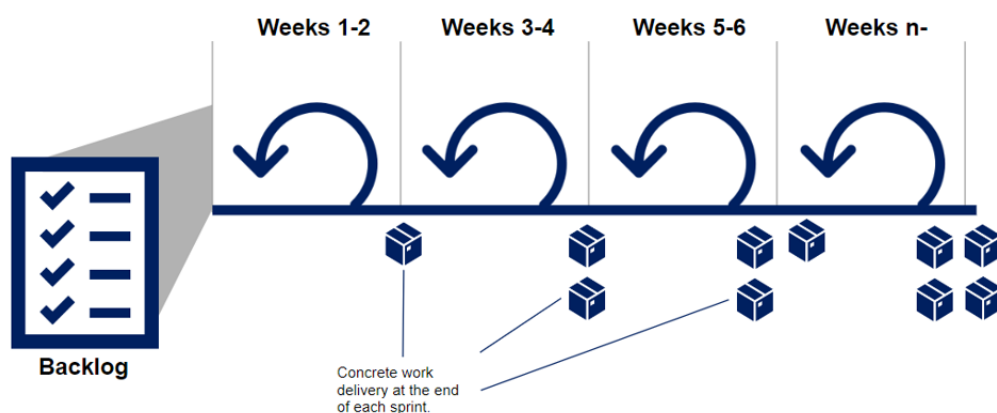
Why Scrum and not Kanban?

- Delivery time is long
- Project scope is large and deliverables are complex
- Team is larger and requires more management



More in PM trainings!

Kuva 3. Taustayrityksen luvalla lainattu kuvankaappaus yrityksen uusille työntekijöille esitettävästä materiaalista. Tämän dian tarkoituksena on lyhyesti esitellä uudelle työntekijälle Scrumin periaatteet ja motivoida mallin käyttö. Yrityksen nimi häivytetty kuvasta.



Kuva 4. Taustayrityksen luvalla lainattu kuvankaappaus uuden työntekijän koulutusmateriaaleista. Kalvo on osa koulutusta, jossa käsitellään asiakkaiden odotuksien hallintaa. Yrityksen nimi häivytetty kuvasta.

Taustayrityksessä Scrumia on monien haastateltavien mukaan sovellettu jonkin verran teoriasta. Vaikka taustayrityksen Scrum onkin sovellettu versio, ei malli

ole kuitenkaan niin pitkälle sovellettu, ettei sitä voisi käyttää tämän opinnäytetyön pohjana. Sovellettu Scrum noudattelee kuitenkin virallisesti pitkälti alkuperäistä Scrumia, kuten kuvista 3 ja 4 havaitaan. Lisäksi taustayritys ylläpitää työntekijöidensä osaamista Scrumista tarjoamalla työntekijöilleen mahdollisuuksia virallisten Scrum-sertifiointien suorittamiseen. Toisaalta opinnäytetyön tarkoituksena onkin tehdä havaintoja, miten Scrumia mahdollisesti sovelletaan teoriasta poikkeavin tavoin, jolloin täysin puhtasoppista mallia soveltavan yrityksen seuraamisesta tuskin olisi saatavilla lisäarvoa opinnäytetyöhön.

4 SCRUMIN TEORIAN JA KÄYTÄNNÖN MAHDOLLISET EROT

4.1 Tutkimuskysymykset

Tämän opinnäytetyön tavoitteena on vastata kysymykseen, millä eri tavoin edellisessä luvussa esitetystä teoreettisesta Scrum-viitekehystä mahdollisesti poiketaan työelämän ohjelmistoprojekteissa käytännössä. Selkeyden ja kokonaisuuden hallitsemisen nimissä eroavaisuuksia teoriassa ja käytännössä tarkastellaan Scrumin roolituksien, tapahtumien ja tuotoksien kautta. Tutkimuskysymykset voidaan siis pilkkoa tarkemmin alla olevalla tavalla.

1. Miten Scrumin rooleista mahdollisesti poiketaan käytännössä?
 - a. Miten kehittäjän rooli ja tiimin koko mahdollisesti eroavat teoriasta?
 - b. Miten tuoteomistajan rooli mahdollisesti eroaa teoriasta?
 - c. Miten Scrum masterin rooli mahdollisesti eroaa teoriasta?

2. Miten Scrumin tapahtumista mahdollisesti poiketaan työelämässä?
 - a. Miten sprinttien käytännöt mahdollisesti eroavat teoriasta?
 - b. Miten sprintin suunnittelu, päivittäispalaverit, sprintin katselmointi ja sprintin retrospektiivi mahdollisesti eroavat teoriasta?

3. Miten Scrumin tuotokset mahdollisesti poikkeavat teoriasta?
 - a. Miten tuotteen kehitysjonoa hyödynnetään käytännössä?
 - b. Miten sprintin kehitysjonoa hyödynnetään käytännössä?
 - c. Miten inkrementtejä ja yksittäisiä tehtäväkuvauksia hyödynnetään käytännössä?

4.2 Hypoteesit

Vaikka Scrum-viitekehys onkin melko kevyt, uskoisin, että käytännössä teoriasta kuitenkin poiketaan. Oma hypoteesini on, että ennen kaikkea roolituksissa ja tapahtumissa olisi eniten poikkeamia. Tämä siitä syystä, että nämä saattanevat vaikuttaa eniten yrityksessä jo oleviin prosesseihin ja toimintatapoihin. Tuotoksien kohdalla poikkeamia tuskin on merkittävästi. Suurin poikkeama on

todennäköisesti valmiin määritelmien puutteellisuudessa, mikä saattanee johtua aika- ja budjettirajoitteista.

Roolien kohdalla uskoisin tapahtuvan jonkinasteista päällekkäisyyttä ja vastuiden sekoittumista. Tuoteomistajan vastuut saattavat olla laadun varmistamisen puolesta usein selkeästi määriteltynä nimetyille henkilölle, mutta esimerkiksi Scrum masterin tehtävät saattavat olla löyhästi sidottuna Scrumin teoriaan, jos nimettyä Scrum masteria ylipäättäen on. Tähän saattavat vaikuttaa yrityksen omat kokemukset Scrum masterin tarpeellisuudesta ja kenties myös Scrumin ulkopuoliset tittelit ja roolit saattavat sekoittaa tilannetta. Tuoteomistajan merkitys on Scrumissa korvaamaton, samoin kehittäjien, mutta kenties Scrum master saatetaan nähdä projektipäälliköiden ja muiden asiakasvastaavien ohella tarpeettomana.

Tapahtumissa on kenties huomattavia eroavaisuuksia, sillä niille tarjotaan oppaassa selkeitä aikamääritelmiä: muun muassa päivittäispalaveri on 15 minuuttia pitkä ja pidetään joka päivä, sprintin suunnittelu on korkeintaan kahdeksan tunnin mittainen sprintin ollessa kuukauden mittainen ja sprintin tulisi olla maksimissaan kuukauden mittainen. Näiden määreiden noudattaminen jo valmiiksi hektisessä työympäristössä voi tuoda omat lisähaasteensa. Esimerkiksi 15 minuutin palaverin sovittaminen useampia projekteja suorittavien kehittäjien kalenteriin voi olla haastavaa ja lisäksi kallista. Mahdollisesti kaikkia tapahtumia ei siis edes järjestetä vastoin Scrumin virallista teoriaa. Saattaa myös olla, että tapahtumien protokolla ei noudata täysin teoriaa.

5 SCRUMIN TEORIAN JA KÄYTÄNNÖN HAVAITTUJA EROAVAISUUKSIA

Tässä luvussa on esitelty varsinaiset havainnot Scrumin teorian ja käytännön eroista, jotka ovat selvinneet kirjallisuuskatsauksen, asiakasprojektin seurannan ja haastattelujen pohjalta. Tarkastelu aloitetaan rooleista, minkä jälkeen siirrytään tapahtumiin ja lopuksi vertaillaan vielä tuotoksien eroja teoriassa ja käytännössä. Rooleja, tapahtumia ja tuotoksia käsittelevien kappaleiden alussa on vielä esitelty keskeisimpiä havaintoja ja taustoja seuratulle projektille sekä haastatteluille. Haastattelujen kysymykset löytyvät jokaisen kappaleen alusta.

5.1 Erot rooleissa

Opinnäytetyötä varten seuratussa asiakasprojektissa oli seurannan aikana ensisijaisesti mukana seitsemän henkilöä: yksi tuoteomistaja, yksi projektia hallinnoiva CSM ja kuusi kehittäjää, joista yksi oli samaan aikaan aiemmin mainittu projektin tuoteomistaja. Jäljempänä on mahdollista huomata, että etenkin kehittäjän rooli saattaakin usein mennä päällekkäin muiden roolien kanssa, mikä ei sinänsä riko Scrumin teoriaa, mutta ei tosin vastaa täysin suositeltua linjaa.

Opinnäytetyötä varten haastatelluilta henkilöiltä kysyttiin rooleihin liittyen alla olevat kysymykset. Kysymyksien avulla oli tarkoituksena löytää kohtia, joissa voitaisiin löytää selkeitä eroja teorian ja käytännön välillä. Näin ollen kysymyksissä korostuu roolien vastuut, päällekkäisyydet ja puuttumiset.

- Missä rooleissa olet toiminut?
- Jos toimit tällä hetkellä tai olet toiminut Scrum-projektissa, missä roolissa toimit? Mitkä vastuut kuuluvat roolisi alle?
- Oletko toiminut projektissa, jossa on useampi kuin yksi tuoteomistaja? Entä Scrum master?
- Onko ollut projekteja, joista on puuttunut tuoteomistaja tai Scrum master?
- Minkä kokoisissa Scrum-tiimeissä olet toiminut?

- Onko yleistä, että projektitiimeissä tapahtuu muutoksia? Miten roolit ovat mukautuneet, jos projektitiimissä on tapahtunut muutoksia projektin edetessä?
- Oletko ollut mukana projektissa, jossa on ollut päällekkäisiä rooleja? Mitkä roolit ovat menneet päällekkäin?
- Oletko toiminut projektissa, jossa useampi Scrum-tiimi on toiminut samassa projektissa?

5.1.1 Kehittäjät ja Scrum-tiimin koko

Kehittäjän roolissa toimivat luovat Scrumin aikana kaiken käyttökelpoisen sisällön kullekin inkrementille. Kehittäjien vastuulla on kullekin sprintille tehtävän sprintin kehitysjonon määrittely toteutettavalle tasolle valmiin määritelmää noudattaen. Useimmiten kehittäjiltä vaaditaan monipuolisia taitoja. (Sutherland & Schwaber 2020, 5.) Kehittäjät ovat siis kunkin Scrum-projektin varsinaisen toteutuksen luojat. Jäljempänä tuoteomistajaa ja Scrum masteria käsittelevissä kappaleissa tullaan kuitenkin huomaamaan, että toisinaan kehittäjät saavat myös päällekkäisiä rooleja tuoteomistajan tai Scrum masterin kengissä.

Operating Model

Roles in the project

Below is a list of roles in the project

Role	Organization	Description and Contact details
Product owner	Acme Corp	Responsible of creating and prioritizing the user stories chosen to realisation.
Business Expert	Acme Corp	Supports creating user stories
Solution Architect	Acme Corp	Defines integration architecture with Salesforce architect
Developer	Acme Corp	Design and develops integrations
External system development	Acme Corp	Mobile / Customer Portal development
Salesforce architect	Taustayritys	Defines Salesforce architecture, make sure solution is aligned with Salesforce best practices
Salesforce consultant	Taustayritys	Responsible for Salesforce design and configuration
Salesforce developer	Taustayritys	Responsible for Salesforce custom development

Kuva 5. Taustayrityksen roolikuvaukset projektille. Yrityksen nimi häivytetty.

Scrumissa jokainen tiimin kehittäjä on ”tiimin jäsen”, eikä ryhmässä käytetä kiinteitä asiantuntijatitteleitä, kuten liiketoiminnan analyttikkoa, tietokannan pääkäyttäjää (DBA), team leadia, käyttöliittymäsuunnittelijaa tai ohjelmoijaa (Deemer, Benefield, Larman & Vodde 2012, 4). Tästä poiketen taustayrityksessä

kuitenkin käytetään Scrum-projekteissa erillisiä titteleitä, kuten projektien aloitukseen tarkoitettusta kuvan 5 taustamateriaalista ilmenee. Kiinnostavasti Scrumin rooleista ainoastaan tuoteomistaja ja kehittäjä ovat erikseen mainittu, eikä Scrum masteria ole listauksessa lainkaan.

Myös esimerkiksi Masood ym. ovat havainneet haastatteluissaan, että Scrum-tiimin jäsenet saattavat tavanomaisesti koostua alansa asiantuntijoista, jotka ovat roolittaneet itsensä muutamainkin tarkempiin tehtäviin, esimerkiksi front-end ja back-end tehtäviin aikaisemman kokeneisuutensa pohjalta (Masood ym. 2022, 1584). Osa Dieboldin, Ostbergin, Wagnerin ja Zendlerin haastattelemista tiimeistä ilmaisi, että asiantuntijatiimit saattavat kokea syvällisen ymmärryksen lisäävän tiimin tehokkuutta. Toisaalta taas yleisempää osaamista suosivat tiimit kokivat monialaisuuden vähentävän ulkopuolisten tekijöiden tarvetta ja tasapainottavan vastuuta paremmin. (Diebold, Ostberg, Wagner & Zandler 2015, 6.)

Asiantuntijatiimiä ja monialaisuutta ei tulisikaan näin ollen välttämättä hahmottaa toisensa pois sulkevin vaihtoehtoina, vaan pikemminkin ääripäinä jatkumolle, jonka välille moni Scrum-tiimi oletettavasti asettuu. Vaikka Scrum-tiimin tulisikin olla riittävän monialainen pärjätäkseen ilman jatkuvaa ulkopuolelta haettavaa erityisosaamista, laajoissa projekteissa tuskin on mahdollista luoda kaiken osaamisen kattavaa tiimiä, joka olisi lisäksi vielä tarpeeksi pieni ollakseen ketterä.

Ketterässä kehittämisessä pienemmät kehitystiimit ovatkin suositeltuja, eikä Scrum ole tässä poikkeus. Scrum-oppaan mukaan tiimien tulisi olla pieniä, enintään kymmenen henkilön suuruisia (Sutherland & Schwaber 2020, 5). Scrum Primer ottaa tarkemmin kantaa todeten Scrum-tiimin koostuvan viidestä yhdeksään henkilöstä (Deemer ym. 2012, 4). Jos projektissa on yksi erikseen nimetty Scrum master ja yksi tuoteomistaja, tarkoittaisi tämä heidän lisäksi korkeintaan kahdeksaa kehittäjää. Toisinaan kuitenkin näistä rajoja venytään suuntaan ja toiseen. Esimerkiksi Diebold ym. havaitsivat tutkimuksissaan, että joissakin yrityksissä on toiminut vain kahden hengen Scrum tiimejä, kun taas toisissa on ollut 10 tai enemmän jäsentä (Diebold ym. 2015, 6). Suurempiakin, jopa yli 16 hengen, Scrum-tiimejä on löydetty Paulkin teettämässä kyselytutkimuksessa (Paulk, 2013, 31).

Seuratessa asiakasprojektissa toimi kevään ja alkukesän 2024 aikana kuusi kehittäjää. Projektin "lead consultant" -nimikkeellä toimiva taustayrityksen kehittäjä vastasi Salesforce-ympäristön deklaratiiivisesta kehittämisestä ja designista. Projektin "lead developer" vastasi taas räätälöityjen, kustomoidusti koodaten rakennetuista komponenteista. Kolmantena kehittäjänä projektissa toimi asiakkaan tuoteomistaja, joka vastasi pienemmistä ylläpidollisista kehitystehtävistä. Projektissa oli edellisten lisäksi kaksi ohjelmistokehittäjää ja yksi konsultti pienemmällä allokatiolla tekemällä rajatumpia ratkaisuja. Käytännössä projektitiimissä kehittäjien määrä pysyi siis viitetasossa.

Opinnäytetyön haastatteluissa selvisi, että toisinaan Scrum-projekteihin tulee mukaan niin sanottuja "vierailevia tähtiä", jotka toteuttavat pienen osatoteutuksen, minkä jälkeen he poistuvat projektista seuraavan sprintin alkaessa. Haastateltavista jokainen mainitsi taustayrityksen Scrum-tiimien asettuvan yhden ja kymmenen henkilön välille, mikä on ainakin ylärajan puolesta Scrumin viitearvoissa. Haastateltavien vastauksista voitiin haarukoida keskimääräisen Scrum-tiimin kehittäjien määrän asettuvan noin kahden ja viiden hengen välille.

Haastateltavien mukaan Scrum-tiimit ovat lähtökohtaisesti olleet pieniä asiakkuuksien pienen koon seurauksena – projektit eivät ole olleet suuren mittakaavan hankkeita, joissa tarvittaisiin suurta kehitysnopeutta, kuten eräs haastateltavista totesi. Tällöin myös jokaiselle kehittäjälle riittää tekemistä, kuten toinen haastateltavista totesi. Osa haastateltavista mainitsi kuitenkin tilanteista, joissa taustayritys on palvellut asiakastaan "yhden miehen Scrum-tiimillä", jossa yksi kokeneempi konsultti on toiminut samaan aikaan kehittäjänä, Scrum masterina ja tuoteomistajana.

On ilmiselvää, että edellä havaitussa yhdenmiehen Scrumin tapauksessa Scrumin roolituksen teoria on jo äärimmilleen sovellettu. Pienissä asiakkuuksissa tämä voi tosin olla perusteltua, jos toteutuksessa on pysyvä matalassa budjetissa tai toteutettava tuote on todella yksinkertainen. Toisaalta voidaan pohtia, onko tämä Scrumia ylipäätään päällekkäin menevien roolien ja äärimmäisen pieneksi muodostetun "kehitystiimin" rikkoessa Scrumin teoriaa näin räikeästi.

Toisaalta taustayrityksessä kehitystiimien pientä kokoa selittää työskentelymalli, jossa työntekijöillä on lähtökohtaisesti useampi asiakasprojekti samaan aikaan työn alla. Tämä on tosin vastoin Scrumin suosituksia. Scrum Primerissa todetaan Scrum-tiimien olevan tuottavimmillaan, kun tiimin jäsenet ovat täysin sitoutuneita yhteisen projektiin ja tiimit pysyvät vakaina (Deemer ym. 2012, 4).

Tällaista tuottavuuden laskua oli havaittavissa Hassani-Alaouin, Cameronin ja Giannellian case-tutkimuksessa, jossa jopa kahdeksassa yhdestätoista tiimistä kehitystiimin jäsenillä oli sprintin ulkopuolisia, häiriöitä aiheuttavia velvoitteita. Monet vastaajista nimesivät nämä häiriötekijät suurimmaksi viivästytekijäksi tuotteen toimituksessa (Hassani-Alaoui, Cameron & Giannellia 2020, 6261.) Lahden, Tuovisen, Mikkosen ja Capillan teknisen velan ja prosessivelan välistä suhdetta selvittävässä case-tutkimuksessa havaittiin, että tutkimuksen kohteena olleessa startupyrityksessä yhtenä isona häiriötekijänä oli Scrum-prosessin ohelle otettu toimintatapa reagoida välittömästi asiakkaiden tukipyyntöihin (Lahti, Tuovinen, Mikkonen & Capilla 2022, 319). Työrauhan rikkoutuminen yhtenä merkittävänä syynä vaikuttaisi siis olevan Scrum-tiimin jäsenten velvoittaminen useampaan tehtävään samanaikaisesti. Tämä toimintatapa voi olla peruja yrityksen aikaisemmista toimintamalleista ajalta ennen Scrumia.

Edellisestä Scrum Primerin vakauteen tähtäävästä suosituksesta poiketen opinnäytetyötä varten tehdyissä haastatteluissa selvisi lisäksi, että monesti Scrum-tiimeissä tapahtuu henkilöstönvaihdoksia – varsinkin aikajänteen kasvaessa. Useampi haastateltava mainitsi jo aiemmin esitellyt ”vierailevat tähdet”, jotka tulevat projektiin tuottamaan erityistä osaamista vaativan osatoteutuksen. Tässä tosin toistuu jo Dieboldin ym. haastatteluissa ilmennyt tasapainottelu asiantuntijatiimien ja moniosaajista koostuvien tiimien välillä: asiantuntijatiimi saattanee päästä syvemmälle ymmärryksen tasolle, mutta toisaalta tiimin vakaus voi kärsiä, kun tiimiin tarvitaan laajempaa osaamista.

Osa opinnäytetyötä varten haastateltavista mainitsi, että skaalautuvuus ja joustavuus on osa asiakkaalle myytävää kokonaisuutta: Asiakkaalle ei myydä pelkästään yhtä kiinteää tiimiä ja sen osaamista, vaan kyvykkyyttä koko taustayrityksen työvoimasta. Kyse on jälleen kerran joustavuuden ja vakauden välillä tasapainottelusta.

Myös Hassani-Alaouin ym. kvalitatiivisessa case-tutkimuksessa ilmeni Scrum-tiimien skaalautumista projektien kasvaessa (Hassani-Alaoui ym. 2020, 6262). Toisaalta samassa tutkimuksessa havaittiin, että toisinaan Scrum-tiimin kasvaminen liian isoksi hoidettiin yksinkertaisesti jakamalla tiimi kahtia, mikä erään haastateltavan mielestä johtaa helposti siiloutumiseen ja näkyvyyden rajoittumiseen (Hassani-Alaoui ym. 2020, 6262).

Edellä mainittu tapa jakaa Scrum-tiimi kylmästi kahtia ei vaikuttaisi olevan kovin yleinen toimintatapa, mutta toimintatavan mainitut seuraukset tuskin yllättävät ketään. Yleisestikin tiivis ja siiloutumaton yhteistyö lienevät syitä, joiden takia Scrum-viitekehityksessä suositaan pieniä ja vakaita tiimejä.

5.1.2 Tuoteomistaja

Scrumin teoriassa tuoteomistaja vastaa siitä, että Scrum-tiimin työstä saadaan suurin mahdollinen lisäarvo. Tuoteomistajan yksi tärkeimmistä tehtävistä on tuotteen kehitysjonon hallitseminen. Käytännössä tämä tarkoittaa, että tuoteomistajan on kyettävä viestimään tuotteen tavoite ja sisältö selkeästi tiimin sisällä, priorisoimaan kehitysjonon sisältö sekä varmistaa sisällön läpinäkyvyys ja ymmärrettävyys. Tuoteomistaja on näistä tehtävistä vastuussa, vaikka hän voikin delegoida tehtäviä eteenpäin. Tuoteomistajalla on oltava yrityksen sisällä mandaatti toiminnalleen ja tähän tehtävään osoitetaan vain yksi henkilö. (Sutherland & Schwaber 2020, 6.) Tiivistetysti voidaan todeta, että tuoteomistaja on yksi henkilö, joka toimii laadunvarmistajana, siltana Scrum-tiimin ja sidosryhmien välillä sekä tuotteen kehitysjonon omistajana.

Scrumin teoriassa todetaan (Sutherland & Schwaber 2020, 6), että ”Tuoteomistaja voi itse tehdä yllä mainitut työt tai delegoida niitä muille. Joka tapauksessa tuoteomistaja vastaa näistä töistä”. Toisaalta myöhemmin todetaan (Sutherland & Schwaber 2020, 6), että ”Tuoteomistaja on yksi henkilö, ei komitea”. Vastuu esitellyistä tehtävistä on siis aina tuoteomistajalla, vaikka hän jakaisikin tehtäviään eteenpäin. Masood ym. haastatteluissa selvisi, että joillakin tuoteomistajilla oli takanaan tukitiimi jakamassa tuoteomistajan vastuita (Masood ym. 2022, 1583). On mahdollista pohtia, että missä vaiheessa tuoteomistaja muuttuu ”yhdestä henkilöstä komiteaksi”. Vaikka vastuu säilyisikin muodollisesti

tuoteomistajalla, tapahtuuko näin käytännössä vai siirtyykö myös vastuu hiljaisesti delegoidulle henkilölle?

Toisaalta Masood ym. havaitsivat, että tuoteomistaja saattaa myös tehtävien delegoimisen lisäksi ottaa työpöydälleen lisävastuita ja toimia myös kehittäjän roolissa (Masood ym. 2022, 1583). Paulkin kyselytutkimuksen perusteella vaikuttaisi jopa, että pelkästään tuoteomistajan roolissa toimiminen olisi enemmän poikkeus kuin sääntö. Tässä tutkimuksessa selvisi, että tuoteomistaja saattaa toimia kehitystiimin sisällä, minkä voidaan nähdä haittaavan liiketoiminnan tarpeiden ja teknisen toteutuksen kuromista umpeen. (Paulk 2013, 32.) Tämä on ymmärrettävää, sillä liian tekninen tuoteomistaja saattaa kehitystyöhön keskittyessään unohtaa liiketoiminnan ja loppukäyttäjien tarpeet.

Seuratun projektin tuoteomistaja toimi edellä esitetyn kaltaisessa tuoteomistajan ja kehittäjän kaksoisroolissa. Kyseinen tuoteomistaja oli aiemmin vastannut asiakkaan Salesforce-ympäristön kehittämisestä, minkä seurauksena tuoteomistajan käsitys sekä yrityksen liiketoiminnasta että tuotteen teknisistä ominaisuuksista oli kohtalaisen vahva. Projektin seuraamisen aikana tuoteomistaja oli yhä mukana kehitystyössä tehden pienempiä muutoksia yhdessä alaisuudessaan toimivan assistentin kanssa. Seuratussa projektissa tuoteomistaja oli hyvin selvillä yrityksen liiketoiminnasta, jolloin loppukäyttäjien tarpeetkin tulivat esille.

Yksi tuoteomistaja tärkeimmistä vastuista on huolehtia tuotteen kehitysjonon täsmällisyydestä, selkeydestä ja järjestämisestä. Näin ollen tämän ydintehtävän delegoiminen toiselle henkilölle voitaisiin jo nähdä jonkinasteisena rikkomuksena Scrumin teoriaa vastaan. Toisaalta Sverrisdottirin ym. haastattelututkimuksessa kaksi viidestä haastatellusta tuoteomistajasta totesi, että projekteissa tuotteen kehitysjonon hallitseminen on joko Scrum masterin tai erillisen projektipäällikön vastuulla (Sverrisdottir ym. 2014, 265).

Myös Hassani-Alaoui ym. case-tutkimuksessa ilmeni, että huomattavan monessa Scrum-tiimissä tuoteomistaja ei ole ainoa henkilö, joka vastaa tuotteen kehitysjonosta. Joitakin mainittuja syitä olivat mm. tuoteomistajan kokemattomuus,

tehtävän jakaminen erillisen liiketoiminnan analyytikon kanssa ja tuoteomistajan muut kiireet. (Hassani-Alaoui ym. 2020, 6261.)

Voi kenties olla hyvin perusteltua, että tuotteen kehitysjonosta ottaa vastuun kokeneempi tiimin jäsen tilanteissa, joissa asiakkaan puolelta tulevalla tuoteomistajalla välttämättä ole aikaa tai osaamista, saati edes kiinnostusta, tuotteen kehitysjonon hallintaan. Lähtökohtaisesti kuitenkin Scrumin teorian mukainen lähestymistapa tällaisiin tilanteisiin olisi edellyttää Scrum masterilta vahvempaa tukea tuoteomistajalle kehitysjonon menestyksekkääseen hallintaan.

Vuoden 2020 Scrum-oppaassa ei oteta kantaa siihen, tulisiko tuoteomistajan olla asiakkaan puolelta vai toimitusyrityksen sisäisesti. Palveluita tarjoavien yritysten kohdalla on kuitenkin hyvin mahdollista, että tuoteomistajaksi on valittu asiakkaan puolelta edustaja (Masood ym. 2022, 1583). On myös havaittu tapauksia, joissa asiakkaalta tulevan tuoteomistajan lisäksi projektissa on ollut myös toimitusyrityksen sisäinen tuoteomistaja (Diebold ym. 2015, 6). Myös opinnäytetyötä varten tehdyissä haastatteluissa selvisi, että toisinaan taustayrityksessä on sekä asiakkaan että taustayrityksen puolelta sisäisesti järjestetty kaksi tuoteomistajaa. Tässä tapauksessa voitaisiin jo selkeästi todeta eräänlaisen komitean muodostuneen, eikä käytäntö enää vastaisi Scrumin teoriaa.

Toisaalta tällainen sisäisen ja ulkoisen tuoteomistajan malli varmasti edistää asiakkaan ja toimitusyrityksen välistä luottamusta molempien ollessa sitoutuneita tuotteen tavoitteisiin ja niiden viestimiseen kehitystiimille. Muutenkin viestintä asiakkaan ja toimittavan yrityksen välillä voi kenties parantua tällä mallilla. Lisäksi saatetaan välttää aiemmin esitelty tilanne liian teknisestä tai liian liiketoimintalähtöisestä tuoteomistajasta asiakkaalta tulevan, ulkoisen tuoteomistajan vastatessa enemmän liiketoiminnan tavoitteista ja toimitusyrityksen, sisäisen tuoteomistajan ollessa enemmän vastuussa teknisestä toteutuksesta.

Toisaalta teoriaa rikkovien käytäntöjen kohdalla voidaan todeta, että kaksi tuoteomistajaa on kenties pienempi paha kuin ei yhtään tuoteomistajaa. Esimerkiksi Dieboldin ym. teettämässä haastatteluissa ilmeni, että joissakin tapauksissa tuoteomistajaa ei ole ollut lainkaan, vaan vaatimukset on saatu suoraan sidosryhmien edustajilta tai tuotepäälliköiltä (Diebold ym. 2015, 6). Myös

Elorannan, Koskimiehen ja Mikkosen haastatteluissa ilmeni, että seitsemällä tiimillä kahdeksastatoista puuttui kokonaan tuoteomistaja (Eloranta, Koskimies & Mikkonen 2016, 197).

Edellisten lisäksi opinnäytetyötä varten tehdyissä haastatteluissa ilmeni, että taustayrityksen työntekijät ovat olleet mukana Scrum-projekteissa, joissa on ollut useampi tuoteomistaja, mutta lisäksi on ollut projekteja, joista tuoteomistaja on puuttunut kokonaan. Edellä esitetyn sisäisen ja ulkoisen tuoteomistajan lisäksi yksi haastateltavista kertoi olleensa projektitiimissä, jossa oli peräti neljä tuoteomistajaa, joista kukin edusti omaa liiketoiminnan yksikköänsä. Tässä tapauksessa kehitystyötä on tehty samalle alustalle, mutta ratkaisut on rakennettu eri liiketoiminnoille, erilaisilla näkyvyyksillä ja eri sovelluksille. Samanlaista tuoteomistajien hajauttamista saman tuotteen eri funktioiden välillä havaittiin Paasivaaran, Durasiewiczin ja Lasseniuksen yhtä monikansallista Scrum-tiimiä seuranneessa case-tutkimuksessa (Paasivaaran, Durasiewiczin & Lassenius 2008, 532).

Tällaisissa tapauksissa useamman tuoteomistajan käyttäminen on tuskin rikkomus Scrumin teoriaan, kunhan tuoteomistajia palvelevat tiimit kykenevät erottamaan vastuunsa ja velvoitteensa eri tuoteomistajien alaisuudessa. Saattaa jopa olla, että kaikkien liiketoimintojen kehittämistä edistävän yksittäisen tuoteomistajan käyttäminen olisi liki mahdotonta velvoitteiden kasvaessa liian suuriksi yhdelle henkilölle. Tällöin saatettaisiin lopulta päätyä aiemmin esiteltyyn tilanteeseen, jossa tuoteomistaja lopulta delegoi tuotteen kehitysjonon hallinnan eteenpäin ja tuoteomistajuus on käytännössä siirtynyt toiselle henkilölle.

Tuoteomistajan puuttumisen syyksi opinnäytetyötä varten tehtyjen haastattelujen vastauksissa mainittiin esimerkiksi pienet asiakkaat, jotka haluavat perustason CRM-järjestelmäratkaisun, mutta eivät välttämättä tiedä itsekään, mitä haluavat. Toisaalta asiakkaalle ei välttämättä ole alusta alkaen selvää, että vaaditaan nimenomaan yksi henkilö, joka vastaa roolista. Näissä tapauksissa haastateltavat mainitsivat tuoteomistajan kuitenkin usein löytyvän asiakkaalta jossakin vaiheessa. Jos tuoteomistajaa ei ole löytynyt asiakkaan, on tuoteomistaja tuotu mukaan tiimiin vähintäänkin taustayrityksen sisältä.

Scrum-oppaassa todetaan, että tuoteomistajan tehtävien hoitamista varten tuoteomistajan päätöksillä on oltava koko organisaation kunnioitus (Sutherland & Schwaber 2020, 6). Sinänsä vaatimus vaikuttaa paperilla yksinkertaiselta, mutta todellisuudessa seuratun projektin kohdalla tämän kohdan täyttymisen kriittisyys tuli ilmeisen selväksi. Seuratussa projektissa oli havaittavissa jonkinasteista tuoteomistajan kunnioituksen ja auktoriteetin puutetta, sillä virallisesti Scrum-tiimin ulkopuolella projektin tuoteomistajalla ei ollut valta-asemaa esimerkiksi vaatia loppukäyttäjää suorittamaan edistetyn tuotteen testauksia. Käytännössä tuoteomistajan oli käytettävä testauspyyntönsä loppukäyttäjille liiketoiminnan johdossa, joka puolestaan vaati loppukäyttäjää suorittamaan testaukset.

Scrum-tiimin ulkoisen byrokratian lisäksi tuoteomistajan tehtävää haittasi selvä kunnioituksen puute siinä, että loppukäyttäjät suhtautuivat tuoteomistajan mukaan toisinaan melko passiivisesti sisäisiin demoihin. Loppukäyttäjät eivät esimerkiksi hyödyntäneet täysin kehittäjänä toimivan tuoteomistajan demoja ja testauksia tai reagoineet demojen jälkeisiin palautepyyntöihin. Tuoteomistajalla ei toisin sanoen ollut täyttä taustaorganisaationsa tukea, vaikka Scrum-tiimin sisällä tuoteomistajan arvo tunnustettiin.

5.1.3 Scrum master

Scrum master vastaa Scrumin jalkauttamisesta projektitiimiin käytännössä. Scrum master vastaa tiimin tehokkuudesta kannustamalla tiimin jäseniä itseohjautuvuuteen, järjestämällä tiimille työrauhan, poistamalla etenemisesteet ja huolehtimalla, että kaikki Scrumin tapahtumat järjestetään. Lisäksi Scrum master toimii tuoteomistajana tukena auttamalla kehitysjonon hallinnassa ja tavoitteiden viestinnässä kehitystiimille. Tarvittaessa Scrum master auttaa myös tuoteomistajaa edistämällä Scrum-tiimin ja ulkoisten sidosryhmien yhteistyötä. Organisaation sisällä Scrum master huolehtii Scrum viitekehyksen oikeaoppisuudesta ja neuvomalla organisaationsa jäseniä Scrumin käytännön toteutuksessa. (Sutherland & Schwaber 2020, 6–7.) Etenkin Scrum masterin kyky varmistaa kaikkien Scrumin tapahtumien toteutuminen vaikuttaisi Kadenicin, Koumaditisin ja Junker-Jensenin teettämän tutkimuksen perusteella olevan merkittävässä roolissa Scrumin onnistumisen kannalta (Kadenic, Koumaidtis, Junker-Jensen 2023, 10).

Scrum Primerissa erikseen todetaan, ettei Scrum master ole projektipäällikkö, vaan tiimiänsä tukeva palvelija. Tarkennuksena Deemer ym. toteavat, ettei Scrum-tiimissä ole lainkaan projektipäällikköä, vaan nämä tehtävät on jaettu muiden roolien kesken – erityisesti tuoteomistajan ja kehittäjän tiimin. Tarvittaessa Scrum masterin roolin voi ottaa hoitaakseen projektin entinen tai nykyinen projektipäällikkö, mutta tällöin kyseisen henkilön on kyettävä erottamaan perustavanlaatuiset erot projektipäällikön ja Scrum masterin rooleissa (Deemer ym. 2012, 5.)

Vaikka roolien eroavaisuudet olisivatkin Scrum masterin ja projektipäällikön virkaa samanaikaisesti toteuttavalla henkilöllä tiedossa, voi roolien yhteensovittaminen olla silti haastavaa. Esimerkiksi Nollin, Razzakin, Bassin ja Beechamin haastatteluissa ilmeni, että osa tällaisessa roolissa toimivista henkilöistä koki ristiriidan paineen. Haastateltavat kertoivat, että toisaalta Scrum masterin roolissa on kannustettava tiimin itseohjautuvuuteen, mutta samalla tiimiin on jalkautettava asiakkaan puolelta tuleva tulospaine projektipäällikön roolissa (Noll, Razzak, Bass & Beecham 2017, 317). Edellä mainittu tavoitteiden ristiriita avaa selkeämmin Scrum Primerin vaatimusta, ettei Scrum-projekteissa tulisi olla projektipäällikköä missään muodossa. Nollin ym. tekemässä kirjallisuuskatsauksessa oli viitteitä siitä, että Scrum masterilla on hyvin usein myös projektipäällikön tai vastaavan roolin tehtäviä vastuullaan (Noll ym. 2017, 316).

Paulkin kyselytutkimuksessa on vastaavasti selvinnyt, että useimmiten projekteissa on mukana erillinen projektipäällikkö tai Scrum masterin ja projektipäällikön roolia samanaikaisesti täyttävä henkilö. On tosin käyty kiivasta keskustelua siitä, onko projektipäälliköllä virkaa minkäänlaisissa ketterissä projekteissa, joiden perusluonteeseen kuuluu tiimiläisten voimaannuttaminen ja itseohjautuvuuteen kannustaminen. Vain kaksi viidestä Paulkin teettämän kyselyn vastaajasta ilmaisi, että heillä on lähtökohtaisesti projekteissa erikseen nimetty Scrum master. (Paulk 2013, 31–32.)

Paulkin ja Nollin havaintojen pohjalta vaikuttaisi siltä, että näin toimiessa yhdelle henkilölle kasaantuu liikaa vastuuta ja Scrum-tiimin autonomia kärsii. Projektipäällikön ja Scrum masterin roolin yhdistäminen voitaisiin siis nähdä perusteettomana rikkomuksena Scrumin teoriasta. Kenties yhtenä etuna näin

toimiessa voitaisiin nähdä Scrum masterin vahvempi mandaatti ja kyky viedä kehitettäviä kohteita eteenpäin tiimin autonomian ja ketteryuden kustannuksella.

Dieboldin ym. teettämässä haastatteluissa on selvinnyt, että myös lujatahtoisille ja kokeneille kehittäjille on annettu Scrum masterin rooli heidän vahvan teknisen ymmärryksensä takia tämän mahdollisesti lisätessä Scrum masterin auktoriteettia ja hyväksyntää Scrum-tiimin sisällä (Diebold ym. 2015, 6–7). Myös Ereizin ja Mušićin seuraamissa asiakasprojekteissa oli havaittavissa tilanne, jossa kehittäjä otti Scrum masterin roolin toimittaessaan tiimipäällikön virkaa (Ereiz & Mušić 2019, 326). Kehittäjien joukosta nousevan Scrum masterin malli kenties myös vahvistaa Scrum-tiimin kehittäjien ymmärrystä itse mallista sekä tätä kautta vahvistaa heidän sitoutumistaan Scrumin toimintatapoihin.

Scrum master ei aina ole yksi nimetty henkilö, kuten kaksoisroolissa toimiva projektipäällikön ja Scrum masterin yhteen sulauma. Esimerkiksi Masood ym. haastattelemat henkilöt kertoivat tapauksista, joissa Scrum-projekteissa on käytetty kiertojärjestelmää, jossa kokeneet tiimin jäsenet vuorottelevat Scrum masterin roolissa (Masood ym. 2022, 1583). Tällainen toimintatapa ei ole täysin vastoin Scrumin teoriaa, vaikka suositus olisikin, että projektitiimissä olisi yksi pääasiallinen Scrum master (Deemer ym. 2012, 5). Toisinaan Scrum master saattaa tosin olla myös jaettu useamman projektitiimin kesken, jolloin Scrum master jakaa keskittymisensä useamman projektin välillä (Masood ym. 2022, 1583; Diebold ym. 2015, 6). Kiertävä Scrum masterin rooli ja Scrum masterin jakaminen useamman tiimin kesken todennäköisesti helpottavat roolin täyttämistä, mutta samalla tiimin vakaus saattanee kärsiä.

Siinä missä kehittäjän ja Scrum masterin roolin sekoittuminen ei ole pahin rikkomus Scrumin teoriasta, on Scrum masterin ja tuoteomistajan roolin sekoittuminen vahvasti vastoin mallin teoriaa. Scrum Primer on Scrum-opasta tarkempi määrittäessään, ettei Scrum master voi missään tapauksessa olla sama henkilö kuin tuoteomistaja, sillä näin toimittaessa on vaarana projektin mikromanagementi kyseisen henkilön toimesta (Deemer ym. 2012, 5). Tässä vaiheessa viimeistään on selvää, miksi kehittäjiä käsitelleessä kappaleessa esitellyt ”yhden miehen Scrum-tiimit”, joissa yksi konsultti hoitaa kaikkia Scrumin rooleja samanaikaisesti, menevät vahvasti vastoin teoriaa.

Moniroolituksien ja muiden poikkeavien käytäntöjen lisäksi on myös mahdollista, ettei Scrum masteria ole lainkaan. Esimerkiksi Masoodin ym. haastatteluissa mainittiin tapauksista, joissa Scrum-tiimi on kokenut Scrum masterin olevan tarpeeton tiimin kokeneisuuden vuoksi (Masood ym. 2022, 1583). Toisaalta Dieboldin ym. tutkimuksen perusteella vaikutti siltä, että myös halu välttää kustannuksia oli yksi syy Scrum masterin puuttumiselle (Diebold ym. 2015, 7). Toisaalta yksi Ereizin ja Mušićin seuraamista yrityksistä lähtökohtaisesti suosii erikseen nimettyä Scrum masteria, mutta halu välttää kustannuksia saattaa tulla myös asiakkaan puolelta, jolloin Scrum masterin rooli yhdistetään esimerkiksi liiketoiminta-analyytikon rooliin (Ereiz & Mušić 2019, 326).

Scrum masterin roolille ei siis välttämättä nähdä aina rahallista arvoa, jolloin se joko yhdistetään toiseen rooliin tai jätetään kokonaan täyttämättä. Tällä tavoin säästetään luonnollisesti kustannuksissa lyhyellä aikavälillä, mutta pidemmällä välillä kustannukset saattavat nousta, kun Scrumia toteutetaan epäjohton mukaisesti tai toteuttajien kannalta epäedullisesti. Ilman Scrum masteria tuoteomistaja saattaa esimerkiksi jäädä täysin yksin tuotteen kehitysjonon kanssa ja osa tapahtumista voi jäädä kokonaan pitämättä.

Myös kaikki opinnäytetyötä varten haastatelluista taustayrityksen työntekijöistä kertoivat olleensa mukana projekteissa, joissa Scrum master on puuttunut. Yksi haastateltavista avasi Scrum masterin toisinaan katoavan taustalle projektin vanhentuuessa. Haastateltava kertoi, kuinka projekti saattaa lähteä ”juhlallisesti” käyntiin selkeillä toimintamalleilla ja rooleilla. Lopulta tiimin itsenäistyessä ja toimintatapojen vakiintuessa asiakkuuden hallinnasta vastaavan päällikön ja Scrum masterin tarve puuttua toimintatapoihin on kuitenkin koettu tarpeettomaksi. Haastateltava koki, että prosessien hallinnasta oppikirjamaisesti voisi olla hyötyä, mutta ylimääräiselle työlle ei ole koettu pakottavaa tarvetta.

Jotkut haastateltavista mainitsivat myös olleensa tilanteissa, joissa projektissa on ollut useampi kuin yksi Scrum master. Haastateltavien vastauksista ilmeni erääksi syyksi useammalle Scrum masterille kehittäjät, jotka ovat hallinnoineet oman kehityskohteensa lisäksi myös tuoteomistajan kanssa tuotteen kehitysjonoa, fasilitoineet asiakastapaamisia ja hoitaneet muitakin Scrum masterin tehtäviä. Tällöin kehittäjä on käytännössä katsoen kahta roolia toteuttava Scrum

master. Kun projektissa on useampi samanlainen tuplaroolissa toimiva kehittäjä, syntyy projektiin luonnollisesti useampi Scrum master. Suurin osa haastateltavista kuitenkin totesi olleensa lähtökohtaisesti projekteissa, joissa on ollut vain yksi Scrum master.

Kuten kehittäjiä ja tiimin kokoa käsittelevän luvun kuvassa 5 ilmeni, taustayrityksessä käytetään hieman sovellettua roolitusta. Kuvasta voidaan havaita, että Scrum masterin roolia ei ole erikseen määriteltynä. Opinnäytetyötä varten seuratussa projektissa oli virallisesti yksi CSM-nimikkeellä toimiva henkilö, joka voitaisiin mieltää Scrum masteriksi. Käytännössä kyseisen päällikön vastuisiin kuului muun muassa projektin budjetin seuranta, aikataulutukset sekä yhteistyön fasilitointi sidosryhmien välillä ja sprinttikokonaisuuksien hallinta Jira-alustalla. Aivan teorian varsinaisessa mielessä projektin Scrum masterina asiakkuuspäällikkö ei kuitenkaan toiminut. Tämä tuli erityisen selväksi tapaamisissa, joissa virallinen projektia vetävä päällikkö ei ollut paikalla, jolloin projektin ”lead consultant” ja ”lead developer” -nimikkeillä toimivat kehittäjät ottivat näkyvämmiin Scrum masterin vastuita alleen tukien tuoteomistajaa ja muuta kehitystiimiä sprinttien suunnittelussa ja tuotteen kehitysjonon jalostamisessa. Näin ollen taustayrityksessä Scrum masterin rooli vaikutti olevan sisään leivottuna muihin rooleihin, kuten kuva 5 antaa olettaa.

Scrumin roolien upottaminen osaksi jo olemassa olevia projektitiimin rooleja, jos Scrum masteria ja projektipäällikköä ei lasketa, ei välttämättä estä Scrumin onnistunutta toteutusta. Esimerkiksi Kadenicin ym. kyselytutkimuksessa selvisi, ettei roolien orjallisen toteuttamisen ja onnistuneen Scrum-sovelluksen välillä vaikuttaisi olevan merkittävää riippuvuutta, vaikka näkevätkin roolien oikeaoppisen täyttämisen olevan Scrumin onnistumisen kannalta kuitenkin hyödyllistä (Kadenic ym. 2023, 10). Oppikirjaimaisen, roolien pakottamisen sijasta varmaan tärkeämpää onkin, että Scrumissa esitetyt roolien vastuut ovat vähintäänkin jollakin määrättyllä henkilöllä – mieluiten tietysti Scrumin ohjaamalla tavalla.

5.2 Erot tapahtumissa

Kuvasta 6 löytyy taustayrityksen virallisesti määrittelemät tapahtumat Scrum-projekteille. Asiakkaalle toimintamallin esittelevästä materiaalista voidaan

havaita, että kaikki Scrumiin tapahtumat löytyvät ainakin virallisesti taustayrityksen Scrum-projekteista.

Operating model - Scrum Meetings

Meeting	When	Length	Location	Agenda
Project kickoff	At the beginning of the project		Face to face preferred	<ul style="list-style-type: none"> ➤ Introductions ➤ Project offer and targets ➤ Project practices and delivery model ➤ Team and roles ➤ Risks ➤ Other topics
Dailys	Daily	15 min	Online / face to face	<ul style="list-style-type: none"> ➤ What I have completed since last daily? ➤ What will I do before the next? ➤ Are there any impediments?
Weekly / Bi-weekly	Weekly or Bi-weekly	1 hour	Online / face to face	<ul style="list-style-type: none"> ➤ Project and sprint status ➤ Any issues or concerns? ➤ Any escalation requirements?
Backlog Refining	Every second week or as required	1 hour	Online / face to face	<ul style="list-style-type: none"> ➤ Review all new user stories, design and acceptance criteria
Sprint planning	Every second week	1-3 hours	Face to face preferred	<ul style="list-style-type: none"> ➤ Agree the scope of the sprint based on customer's priorities. ➤ Plan sprint backlog items and create related tasks
Demo & Sprint Review	Every second week	1-2 hours	Online / face to face	<ul style="list-style-type: none"> ➤ What has been done in past two weeks? ➤ developers present current status of the work. ➤ Present Demo of features to customer ➤ Customer approves user stories
Retro	Once a Month / Once a Sprint	0,5-1 hour	Face to face preferred	<ul style="list-style-type: none"> ➤ What went well? ➤ Is there issues in the process? ➤ Is there something we can improve in next sprints? ➤ How are people feeling of the progress? ➤ Actions
Workshops	Depends of the project			
Project closing	At the end of the project	1-2 hours	Face to face preferred	<ul style="list-style-type: none"> ➤ Project overview ➤ Project achievements and guarantee time ➤ Next projects ➤ Other topics

Kuva 6. Taustaorganisaatiossa määritellyt tapahtumat Scrum-projektille. Taustayrityksen nimi häivytetty.

Scrumin tapahtumien teorian ja käytänteiden erojen vertailemiseksi opinnäyteytötä varten haastatelluille henkilöille esitettiin alla olevat kysymykset. Kysymyksillä haluttiin ensisijaisesti selvittää, miten usein Scrumin tapahtumia järjestetään, jos järjestetään, ja mitä tapahtumissa oikeastaan tapahtuu. Sprintin suunnittelusessiota koskeva kolmanneksi viimeinen kysymys on täydennetty haastattelujen jälkeen erillisellä kyselyllä haastateltavilta.

- Millaisia kokouksia teillä on ollut Scrum-projekteissa?
- Minkä pituisissa sprinteissä olet ollut mukana?
- Oletko ollut mukana sprintissä, joka on keskeytetty? Kuka teki lopullisen päätöksen keskeytyksestä?
- Kuinka monta kertaa viikossa projektissa on tyypillisesti pidetty päivittäispalavereita (daily) ja minkä pituisia ne tyypillisesti ovat olleet?
- Mitä tyypillisessä sprintin suunnittelusessiossa on tapahtunut ja miten tämä tapahtuma on yleensä edennyt?
- Mitä tyypillisessä sprintin katselmoinnissa on yleensä tapahtunut (Katselmointi eli ei siis retro, vaan toiseksi viimeinen tapahtuma, jossa

katsotaan, mitä sprintin aikana on saatu valmiiksi ja mahdollisesti suunnitellaan tulevaa sprinttiä)?

- Mikä on yleensä ollut sprintin retrospektiivin tarkoitus? Onko niitä ylipäättäen pidetty?

5.2.1 Sprintti

Sprintti on eittämättä Scrumin tärkein tapahtuma, jonka ympärille kaikki muut tapahtumat ja prosessit rakentuvat. Sprinttien tavoitteena on saada valmiiksi tuotteen osakokonaisuus eli inkrementti – virallisemmin ”potentially shippable product increment” eli vapaasti suomentaen ”mahdollisesti tuotantoon vietävä tuotelisäys”. Sprinttien maksimipituus on määritelty neljään viikkoon (Sutherland & Schwaber 2020, 7; Deemer ym. 2012, 3). Kaikilla opinnäytetyössä käytetyillä menetelmillä vaikuttaisi siltä, että sprinttien pituudet noudattelevat todella läheisesti Scrumin teoriaa.

Paulkin kyselytutkimuksessa selvisi, että selkeä enemmistö vastanneista kertoi työskentelevän kahden viikon mittaisissa sprinteissä, ja yli neljän viikon sprinteissä työskennelleet olivat selkeässä vähemmistössä (Paulk 2013, 32–33). Samoin myös Dieboldin ym. teettämässä haastattelututkimuksessa oli mahdollista havaita, että suurin osa vastaajista oli työskennellyt yleisimmin kahden viikon sprinteissä, eikä yksikään vastaajista ilmoittanut työskennelleensä yli neljän viikon sprinteissä (Diebold ym. 2015, 5).

Taustaorganisaatiossa sprinttien ohjepituudeksi on määritetty 1–4 viikkoa, kuten teoriaosuuden kuvasta 3 käy ilmi. Lähes kaikki opinnäytetyöhön haastatelluista henkilöistä totesivat työskennelleensä taustaorganisaatiossa 2–4 viikon sprinteissä. Jokainen haastateltavista totesi kahden viikon sprintin olevan ylivoimaisesti yleisin sprintin pituus projekteissa. Kahta viikkoa pidemmät sprintit olivat haastateltavien mukaan poikkeuksellisia, samoin kuin yhden viikon sprintit. Ainoastaan yksi haastateltavista kertoi olleensa mukana projektissa, jossa oli käytetty yli kuukauden mittaisia sprinttejä. Opinnäytetyötä varten seuratus asiakasprojektin sprintit olivat myös lähes poikkeuksetta aina kahden viikon mittaisia. Ainoastaan poikkeustapauksissa, esimerkiksi pääsiäislomien aikaan, sprintti saatettiin määritellä kahden viikon sijasta kolmen viikon mittaiseksi.

Myös Dieboldin ym. haastatteluissa selvisi, että sprintin pituuden muutokset olivat harvinaisia ja lähinnä yleisistä vapaista johtuvia (Diebold ym. 2015, 7). Sprinttien pituudenhan tulisi lähtökohtaisesti olla vakio (Sutherland & Schwaber 2020, 7; Deemer ym. 2012, 19). Lisäksi Elorannan ym. haastatteluissa selvisi, että ainoastaan yksi tiimi käytti vaihtelevan mittaisia sprinttejä siten, että sprinttiä pidennettiin työn vaatiessa odotettua pidemmän aikaa. (Eloranta ym. 2016, 196.) Tämä on tosin vahvasti Scrumin virallisen teorian kanssa ristiriidassa. Teorian mukaan sprintit ovat aikarajoitettuja, eikä sprinttiä ikinä venytetä riippumatta siitä, onko sprintin aikana suunniteltu työ valmis vai ei (Deemer ym. 2012, 3). Venyttämällä voidaan kenties saada joitakin keskeneräisiä toteutuksen osia valmiiksi, mutta hintana on prosessin epävakaus ja asiakkailta saatavan palautteen ja uusien kehitysehdotuksien vastaanottamisen viivästyminen.

Scrum primer suosittaa, että joka sprintin päätteeksi tuote olisi mahdollisuuksien mukaan valmiina käyttöön ottoon uusilla lisäyksillä, eikä erillisiä julkaisusprinttejä tarvittaisi (Deemer ym. 2012, 15). Tosin esimerkiksi Dieboldin ym. haastatteluissa ainakin yksi yritys ilmoitti erikseen käyttävänsä kahta erillistä sprinttiä, joista toinen oli ”normaali” neljän viikon sprintti, jota seurasi kahden viikon sprintti erinäiselle siistimis- ja dokumentaatiotyölle (Diebold ym. 2015, 7). Myös Elorannan ym. tutkimuksessa paljastui tapauksia, joissa systeemitestaukset toteutetaan joka toisessa sprintissä erillisen testaustiimin toimesta (Eloranta ym. 2016, 196). Samanlaista seuraavassa sprintissä testausta ilmeni myös Wróbelin, Przałan ja Weichbrothin haastattelututkimuksessa, jossa selkeä enemmistö vastaajista totesi tiimensä testaavan ratkaisuja seuraavassa sprintissä (Wróbel, Prząła ja Weichbroth 2023, 816). Toisaalta Lahden ym. case-tutkimuksessa seuratussa yrityksessä ei vaadittu testauksia lainkaan, mikä puolestaan johti haasteisiin teknisen velan lisäksi prosessivelan muodossa (Lahti ym. 2022, 320).

Testauskäytänteiden osalta sovelletut sprintit ovat kuitenkin lähempänä puhdasta Scrumia, kuin tilanteet, joissa sprinttejä ei käytetä lainkaan. Sprinttien ollessa Scrumin pulssi, voidaan tosin pohtia, onko kyseessä enää lainkaan Scrum-projekti, jos sprinttejä ei käytetä. Tästä huolimatta esimerkiksi Lahden ym. case-tutkimuksessa sekä Elorannan ym. selvityksessä paljastui ”Scrum-tiimejä”, jotka väittävät käyttävänsä Scrumia, vaikka eivät sprinttejä käytäkään (Lahti ym. 2022, 319–320; Eloranta ym. 2016, 196).

Opinnäytetyötä varten seuratussa projektissa oli mahdollista havaita vastaavanlaisia sprinttejä, joiden lopussa tuote ei olisi vielä ollut julkaisukelpoinen tehdyillä lisäyksillä. Samalla projektin dokumentaatiota saatettiin täydentää toteuttavasta sprintistä erillään myöhemmässä sprintissä. Sprinteillä toteutettiin asiakasyritykselle muun muassa työaikasuunnittelukomponenttia. Tähän komponenttiin tehtävät muutokset eivät aina olleet kelpoisia lopulliseen julkaisuun. Osittain syy tähän piili testauskäytänteissä, sillä asiakasyrityksen loppukäyttäjät eivät olleet kovin aktiivisia ja innokkaita testaamaan inkrementaalaisesti kehitettävää tuotetta. Sen sijaan loppukäyttäjät odottivat ketterästä toimitusmallista huolimatta vesiputousmallin mukaisesti ennemmin valmista tuotetta projektin loppuksi.

Myös Elorannan ym. tutkimuksessa oli mahdollista havaita loppukäyttäjien ja asiakkaiden mukana tulevia haasteita Scrumin sprinttitoteutuksiin. Haastatteluissa yksi Scrumia käyttänyt tiimi ilmoitti käyttävänsä neljän viikon sprinttejä, joiden sisällä kehitystiimi teki kehitystyötä kahden viikon sprinteillä, sillä asiakas ei ollut halukas osallistumaan palavereihin kahden viikon välein (Eloranta ym. 2016, 196). Tämä on varsin mielenkiintoinen lähestymistapa Scrumin sprintteihin, vaikkei täysin puhtasoppinen. Scrumin toiseksi viimeinen tapahtuma, sprintin katselmointi, nimittäin vaatisi loppukäyttäjiltä ja sidosryhmiltä palautteen, jonka perusteella kehitystiimi ohjaa toimintaansa oikeaan suuntaan.

Sprintin keskeyttäminen on mahdollista sprintin tavoitteen muuttuessa merkityksettömäksi, mutta ainoastaan projektin tuoteomistajalla on valta keskeyttää sprintti. (Sutherland & Schwaber 2020, 8). Opinnäytetyötä varten tehtyjen haastattelujen kohdalla ainoastaan yksi haastateltavista oli ollut mukana sprintillä, joka oli keskeytetty. Keskeytys oli tehty yhteisymmärryksessä asiakkaan puolelta tulleen tuoteomistajan ja muiden Scrum-tiimin jäsenten kesken. Haastateltava tarkensi, että käytännössä projektin tekeminen ei kuitenkaan keskeytynyt, vaan projektissa luovuttiin kesken sprintin Scrumin sprinttirakenteesta, kun malli oli havaittu projektin etenemisen kannalta sopimattomaksi. Seuratussa asiakasprojektissa yhtäkään sprinttiä ei keskeytetty.

5.2.2 Sprintin suunnittelu

Sprintin suunnittelutapahtuma käynnistää sprintin ja tapahtuman tarkoituksena on täsmentää sprintin aikana tehtävät työt. Sprintin suunnittelu alkaa tuoteomistajavetoisesti: tuoteomistaja täsmentää aluksi kehittäjille sprintin tavoitteen ja merkityksen sidosryhmille. Tämän selvityksen pohjalta kehittäjät katsovat yhdessä tuoteomistajan kanssa tuotteen kehitysjonosta, mitä tehtäviä sprintille nostetaan. Lopuksi kehittäjien vastuulla on selvittää, miten valitut tehtävät tullaan toteuttamaan. (Sutherland & Schwaber 2020, 8–9.) Scrum Primer jakaa yllä mainitut Scrum-oppaan kohdat kahteen osaan: tavoite ja tehtävät kuuluvat sprintin suunnittelun ensimmäiseen osaan, kun taas viimeisin toteutustapojen suunnittelu jaetaan sprintin suunnittelun toiseen osaan. (Deemer ym. 2012, 8–9.) Käytännössä kummankin oppaan teoreettinen lähtökohta sprintin suunnittelutapahtumaan on kuitenkin sama. Sprintin suunnittelu vastaa sprintin aikana tehtävän työn osalta kysymyksiin miksi, mitä ja miten.

Neljän viikon sprinteille enimmäispituudeksi on määritelty kahdeksan tuntia, mutta lyhyemmille sprinteille sprintin suunnittelutapahtumalle varattua aika on vastaavasti pienempi (Sutherland & Schwaber 2020, 9). Esimerkiksi kahden viikon sprintille suunnittelun enimmäispituus olisi neljä tuntia.

Dieboldin ym. Haastattelututkimuksessa sprintin suunnittelutapahtuman keston vaihtelu oli huomattavaa. Kaksi haastateltavista kertoi esimerkiksi käyttävänsä neljän viikon sprinttejä, mutta ensimmäinen mainitsi sprintin suunnittelun kestävän maksimiajan eli kahdeksan tuntia, kun taas toinen mainitsi suunnittelun kestävän vain puolituntia. Myös osa kahden viikon sprinttejä käyttävistä mainitsi ylittävänsä sprintin suunnittelulle varatun aikaikkunan. (Diebold ym. 2015, 7.)

Yhtenä syynä sprintin suunnittelusessioiden venymiseen saattaa olla selkeän agendan puuttuminen. Esimerkiksi Mortadan, Ayasin ja Hebigin. case- ja haastattelututkimuksessa kumpikaan seuratuista tiimeistä ei käyttänyt virallista agendaa suunnittelutilaisuuksissaan ja saman totesi lähes kaksi kolmesta kyselyyn vastanneista. Merkittävä osa kyselyyn vastanneista totesi sprintin suunnittelutilaisuuksiensa usein venyvän, kolmella tiimillä jopa 16 tunnin mittaisiksi. Lisäksi Mortada ym. havaitsivat molemmissa seurantatiimeissä sekä

kyselyvastauksissa, että osassa tiimejä sprintin suunnittelussa ei lainkaan lasketa mukaan Scrum-tiimin tulevalla sprintillä käytettävissä olevia resursseja Scrum-tiimin äärimmäisen itseohjautuvuuden seurauksena (Mortada, Ayas & Hebig 2020, 75–76).

Vaikka tarkan ohjelmarungon puuttuminen ja resurssien laskematta jättäminen sellaisenaan eivät ole rikkomus Scrumin teoriaa vastaan, tämä tuskin on kovin edullista Scrum-tiimille eikä välttämättä edistä työpanoksien optimaalista käyttöä.

Seuratessa asiakasprojektissa sprintin suunnittelusessiot olivat yhdistetty osaksi sprintin katselmointisessiota yhdeksi tapahtumaksi, jolle oli annettu nimitys ”demo ja sprintin suunnittelu”. Yhdistelmä tapahtuma järjestettiin aina kahden viikon välein kahden tunnin sessiona, joista ensimmäinen tunti tai puolitoista tuntia oli varattu sprintin katselmointiosuudelle. Katselmointia seurasi aina välittömästi noin puolen tunnin tai tunnin mittainen sprintin suunnittelusessio. Seuratessa projektissa pysyttiin siis aikaraameissa, eikä tapahtumien yhdistely ole merkittävä poikkeama teoriasta.

Ainakin seuratussa projektissa lyhyet sprintin suunnittelusessiot heijastelevat taustarytyksen sovellettua Scrumia, jossa pitkän aikavälin projekteissa Scrumiin yhdistetään jatkuvan kehityksen elementtejä. Sen sijaan, että jokaisessa suunnittelutapahtumassa projektitiimi vastaisi tarkasti kaikkiin suunnittelutapahtuman kysymyksiin – miksi, mitä ja miten – oli seuratussa projektitiimin sovellettu suunnittelutapahtuma huomattavasti käytännönläheisempi. Sprintin suunnittelutapahtumassa tiimi lähinnä valitsi tuotteen kehitysjonosta seuraavalle sprintille kehitettävät kohteet, varmisti edellisten sulkemisen ja jatkoi edellisiä kehityskohteita. Käytännössä suunnittelutapahtuman miksi ja miten toteutetaan tipahtivat pois ja jäljelle jäi kysymys, mitä toteutetaan.

Jatkuvan kehityksen elementtejä sisältävän sovelletun Scrumin lisäksi käytännönläheistä työskentelytapaa selittivät myös mahdollisesti projektitiimin pitkäaikainen työskentely yhdessä. Tuote, sen osakokonaisuudet ja yhteiset pelisäännöt olivat kirkastuneet kaikille tiimin jäsenille yli vuoden työskentelyn jälkeen, minkä seurauksena tuoteomistajan ei tarvinnut jokaisessa tilanteessa perustella kehitystyön tuomaa lisäarvoa. Pitkän aikajänteen seurauksena myös

toimintatavat olivat selkeästi projektissa vakiintuneet, jolloin kehittäjät tiesivät itse parhaiten, miten heidän osuutensa toteutetaan.

Ilmiössä on varmasti pitkälti kyse samanlaisesta Scrum-tiimin rutinoitumisesta, jota yksi haastateltavista toi esille Scrum masterin kohdalla: Projekti aloitetaan juhlallisesti seremoniain, mutta ajan myötä seremoniat jäävät enemmän taustalle. Kuten haastateltava totesi, teorian täsmällisempi soveltaminen voisi olla hyödyksi projektitiimille, mutta samalla ylimääräisen muodollisuuden tuoma vaiva nähdään tarpeettomana. Yleisestikin Scrumin sovelluksien poikkeamat saattanevat pitkälti selittyä sillä, ettei rutinoituneessa tiimissä jakseta nähdä enää vaivaa ”turhan” virallisille toimintatavoille.

Opinnäytetyötä varten haastateltujen henkilöiden vastaukset heijastelivat myös hyvin käytännönläheistä lähestymistapaa sprintin suunnitteluun. Lähinnä sprintin suunnittelutilaisuudessa jaetaan tuotejonon kohtia pienempiin osiin, jotka valitaan työn alle kullekin vastuuhenkilölle. Osa haastateltavista mainitsi seuratussa projektissakin havaitun käytännön yhdistää sprintin suunnittelu- ja katselmointitapahtumat yhdeksi olevan yleinen toimintatapa.

Keskeinen osa sprintin suunnittelun kehittäjävetoista toista osaa on, että kehittäjät itse valitsevat sprintin aikana tekemänsä työn määrän sen sijaan, että tuoteomistaja ohjaisi kehitystiimin työtaakkaa (Deemer ym. 2012, 9). Näin ei kuitenkaan aina ole, kuten sprintin kehitysjonoa sekä inkrementtejä ja yksittäisiä tehtäviä käsittelevissä kappaleissa tullaan myöhemmin huomaamaan.

5.2.3 Päivittäispalaveri

Päivittäispalaverit eli tutummin ”dailyt” ovat nimensä mukaisesti päivittäin pidettäviä 15 minuutin mittaisia kokouksia, joissa kehittäjät voivat yhdessä tarkastella etenemistä kohti sprintin tavoitetta ja tarvittaessa mukauttaa sprintin kehitysjonoa. Prosessin sujuvoittamiseksi päivittäispalaverit tulisi järjestää aina samaan aikaan ja samassa paikassa. Päivittäispalaverin perimmäisenä tarkoituksena on tuottaa kehittäjille suunnitelma tulevalle päivälle ja auttaa kehitystiimiä tekemään nopeita ratkaisuja. (Sutherland & Schwaber 2020, 9.) Scrum Primer täsmentää päivittäispalaverin aikana käytävät kysymykset

seuraavasti: mitä on toteutettu sitten viime tapaamisen, mitä tullaan saavuttamaan seuraavaan tapaamiseen mennessä ja millaisia etenemisen esteitä tiimillä on edessään (Deemer ym. 2012, 11).

Dieboldin ym. haastatteluissa selvisi, että suurimman osan vastaajista päivittäispalaverit kestivät tasan 15 minuuttia. Osalla haastateltavista päivittäispalaverit olivat myös 30 minuutin pituisia ja mahdollisesti vain tarpeen mukaan esimerkiksi joka toinen päivä tai kerran viikossa. (Diebold ym. 2015, 5, 7.) Strayn, Sjøbergin ja Dybån tutkimuksessa sen sijaan havaittiin, että yli 15 minuutin Scrum tapaamiset olivat melko yleisiä (Stray, Sjøberg & Dybå 2016, 116). Myös Mortadan ym. case- ja kyselytutkimuksessa havaittiin päivittäispalavereiden venymistä yli 15 minuutin, sillä tätä tapahtui sekä toisessa seurantaryhmistä että yli puolessa kyselyyn vastanneiden tiimeistä. Merkittävimpinä syinä tähän mainittiin kehittäjien ja läpikäytävien asioiden suuri määrä. (Mortada ym. 2020, 75–76.) Wróbelin ym. kyselytutkimuksen vastauksissa toistui havainto ylipitkistä päivittäispalavereista, sillä yli puolet vastaajista totesi päivittäispalavereiden usein tai aina menevän yli 15 minuutin (Wróbel ym. 2023, 817).

Scrum Primer suosittaleekin järjestämään päivittäispalaverit seisten tapaamisten nopeuttamiseksi (Deemer ym. 2012, 11). Stray ym. tekivätkin tutkimuksessaan mielenkiintoisen havainnon, että istualtaan järjestetyt päivittäispalaverit kestivät keskimäärin 12 minuuttia, kun taas seisten palaverit kestivät keskimäärin 19 minuuttia eli noin 60 % kauemmin (Stray ym. 2016, 113).

Päivittäispalaverien kohdalla opinnäytetyötä varten tehdyissä haastatteluissa havaittiin selkeää poikkeamaa Scrumin teoriasta. Ylivoimaisesti suurin osa haastateltavista mainitsi Scrum-projektien päivittäispalaverien olevan noin 30 minuutin mittaisia kahdesta kolmeen kertaa viikossa. Yksi haastateltavista kertoi myös olleensa mukana projektissa, jossa on pidetty kerran viikossa tunnin mittainen päivittäispalaveri. Ainoastaan yksi haastateltavista oli ollut mukana Scrum-projektissa, jossa oli pidetty 15 minuutin mittaiset päivittäispalaverit, mutta tällöinkin palaverit oli pidetty vain kolmena päivänä viikossa.

Haastatteluista saadut vastaukset oli mahdollista todentaa opinnäytetyötä varten seuratussa asiakasprojektissa. Seuratussa projektissa sprinttien päivittäis-palaverit järjestettiin, muutamia arkivapaita lukuun ottamatta, kahtena kertana viikossa 30 minuuttia kerrallaan. Päivittäispalaverit olivat tyypillisesti aina tiistaisin ja torstaisin samaan aikaan aamulla, mutta joka toinen viikko jompikumpi päivittäispalavereista korvautui kahden tunnin sprintin katselmointi- ja suunnittelusessioilla. Tällaisina viikkoina Scrum-tiimillä oli siis käytettävissään vain yksi 30 minuutin päivittäispalaveri koko viikon aikana. Päivittäispalavereita järjestettiin siis kaksinkertaisella pituudella, mutta huomattavasti harvemmin Scrumin teoriaan verrattuna.

Toisaalta opinnäytetyötä varten seuratussa projektissa toteutui Scrum-oppaan suositus päivittäispalavereiden järjestämisestä aina samaan aikaan ja samassa paikassa. Näin ei nimittäin aina ole, sillä esimerkiksi Mortada ym. havaitsivat tutkimuksensa toisessa seurantaryhmässä sekä kyselyvastauksissa päivittäis-palavereiden rytmityksen epäsäännöllisyyttä ja suurimmiksi syiksi mainittiin kehittäjien päällekkäin menevät muut palaverit ja myöhästely päivittäis-palavereista (Mortada ym. 2020, 75–76).

Pituuden ja jaksollisuuden puolesta sovelletut päivittäispalaverit voidaan kenties nähdä kuitenkin pienempänä poikkeamana kuin päivittäispalaverien puuttuminen kokonaan. Ainakin Hassani-Alaouin ym. tutkimuksessa selvisi, että yli puolessa seuratuissa Scrum-tiimeissä ei järjestetty päivittäispalavereita lainkaan, ja lisäksi tiimeissä, joissa päivittäispalavereja järjestettiin, saattoi olla myös ulkopuolisia henkilöitä mukana (Hassani-Alaoui 2020, 6261).

Opinnäytetyötä varten havainnoidussa projektissa tuoteomistaja toimi samanaikaisesti kehittäjänä, joten tuoteomistaja oli aina päivittäispalavereissa mukana. Tuoteomistaja ja Scrum master voivat osallistua päivittäispalavereihin kehittäjän roolissa osallistuessaan kehitystyöhön (Sutherland & Schwaber 2020, 9). Loppukäyttäjää edustavan ja tuoteomistajan mukana olo kuitenkin muunsi pidennettyjen päivittäispalaverien luonnetta. Scrumin esittämien statuspäivityksien lisäksi seuratuissa päivittäispalavereissa lähes poikkeuksetta käsiteltiin myös projektin yleistä etenemistä ja jalostettiin sekä tuotteen että sprintin kehitysjojoja.

Myös Dieboldin ym. teettämässä haastatteluissa ilmeni, että joissakin Scrum-tiimeissä käydään statuksien synkronoinnin lisäksi myös yleistä keskustelua koko tiimiä koskevista asioista (Diebold ym. 2015, 7). Mortadan ym. kyselytutkimuksessa selvisi, että jopa yli kaksi kolmasosaa vastaajista osallistuu päivittäispalavereihin, joissa ei ole strukturoitua keskustelua, vaan tilaisuuden keskustelu on vapaamuotoisempaa (Mortada ym. 2020, 75). Keskustelun rakenteellisuuden puute saattanee selittää aiempaa havaintoa päivittäispalavereiden venymisestä keskustelun rönnyillessä. Toisaalta voidaan pohtia, onko liian sääntillinen ja asiapitoinen päivittäispalaveri omiaan edistämään tiimin ketteryyttä ja turvalliseksi koettua ilmapiiriä

Päinvastaisesti on myös mahdollista, että kaikissa Scrum-tiimeissä ei vastata edes kaikkiin päivittäispalavereiden peruskysymyksiin edistymisestä ja esteistä. Ainakin Paasivaaran ym. tekemässä case-tutkimuksessa ilmeni, että varsinkin projektin alkupuolella Scrumin päivittäispalaverit jäivät lyhyiksi puutteellisen raportoinnin takia ja varsinkin etenemisen esteiden mainitsematta jättämisen vuoksi (Paasivaara ym. 2008, 533; 534). Lisäksi esimerkiksi Mortadan ym. case- ja kyselytutkimuksessa havaittiin sekä seuratuissa tiimeissä että kyselyiden vastauksissa, että kaikkiin päivittäispalaverin kysymyksiin ei aina vastattu ja etenkin, jälleen kerran, etenemisen esteet jäivät useimmiten raportoimatta. Lisäksi tutkimuksessa selvisi, että toisessa seurantaryhmissä kaikki päivittäispalaverin osallistujat eivät aktiivisesti osallistuneet keskusteluun. Sama toistui myös kyselyiden vastauksissa, joissa reilu kaksi viidesosaa vastaajista totesi, ettei kaikkien panosta odoteta päivittäispalavereissa. Suurimpana syynä mainittiin, että kehittäjät eivät halua osallistua keskusteluun, johon heillä ei omasta mielestään ole arvokasta lisättävää. (Mortada ym. 2020, 75–76.) Myös Wróbelin ym. kyselytutkimuksessa huomattava osa vastaajista totesi, ettei päivittäispalavereissa aina käydä läpi tärkeimpiä, teorian mukaisia aiheita (Wróbel ym. 2023, 817).

Merkittävimpänä syynä päivittäispalaverien järjestämiseen pidempinä ja harvemmin viikossa mainittiin opinnäytetyötä varten tehdyissä haastatteluissa useamman päällekkäisen projektin tekeminen. Taustayrityksessä lähes poikkeuksetta suurin osa työntekijöistä työskentelee samanaikaisesti useamman eri asiakkuuden parissa, minkä vuoksi 15 minuutin yhteisen ajankohdan

löytäminen jokaiselle Scrum-projektin kehittäjälle olisi todella vaikea tehtävä. Toisaalta päivittäispalavereiden yhteensovittamisen vaikeus kalentereihin tällä tavoin toimittaessa on varmasti yksi syy, miksi Scrumin teoria suosittelee Scrum-tiimiläisten keskittyvän yhteen projektiin kerralla.

5.2.4 Sprintin katselmointi

Sprintin katselmoinnissa Scrum-tiimi esittelee sprintin aikana valmistuneet tuotokset projektin keskeisimmille sidosryhmille. Samalla keskustellaan seuraavista kehitystarpeista ja myös tuotteen kehitysjonoa voidaan tarvittaessa uudelleen järjestää ja priorisoida. Katselmoinnin tulisi olla työpajamainen, eikä pelkästään demo, jota yleisö passiivisesti seuraa. Enimmäispituus sprintin katselmoinnille on neljä tuntia sprintin ollessa neljän viikon mittainen, mutta sprintin suunnittelun tapaan tämäkin tapahtuma on vastaavasti lyhyempi, jos sprintti on alle neljä viikkoa. (Sutherland & Schwaber 2020, 10.) Scrum Primer suosittaa edellisten lisäksi myös järjestämään tuoteomistajalle ja sidosryhmille mahdollisuuden kokeilla muuttunutta tuotetta omin käsin testausympäristössä, jotta katselmointi ei latistuisi demoksi (Deemer ym. 2012, 14).

Dieboldin ym. tuottaman haastattelututkimuksessa jokainen haastateltava ilmaisi, että yrityksessä järjestettiin vähintäänkin jonkinlainen katselmointitilaisuus. Yhdessä yrityksessä nämä katselmointitilaisuudet olivat ensisijaisesti keino saada loppukäyttäjiltä palautetta puuttuvista ominaisuuksista. Kahdessa yrityksessä katselmoinnit järjestettiin kahdessa osassa, joista ensimmäinen oli sisäinen katselmointi ja seuraava sidosryhmien kanssa järjestettävä katselmointi. (Diebold ym. 2015, 8.)

Opinnäytetyötä varten haastatelluista jokainen mainitsi sprintin katselmoinnin lähtevän liikkeelle demosta, jossa projektin sidosryhmille on esitelty sprintin aikana valmiiksi saadut toiminnallisuudet. Demon aikana sidosryhmiltä on kerätty ajatuksia ja mahdollisia muutostarpeita, jotka on kirjattu ylös. Osa haastateltavista erikseen mainitsi katselmointisessioiden olevan liiketoiminta- ja muutoslähtöisiä loppukäyttäjiä silmällä pitäen. Haastateltavien vastauksissa korostui kuitenkin katselmointitilaisuuksien demomaisuus, jolloin teorian suosittama työpajamainen lähestymistapa ei välttämättä haastateltujen

projekteissa toteutuneet. Useimmat haastateltavista mainitsivat myös seuraavan sprintin suunnittelun olevan seuraavaksi osana ”katselmointia”, jolloin katselmointitilaisuus oli samalla seuraavan sprintin aloittava sprintin suunnittelu-tilaisuus.

Myös seuratussa asiakasprojektissa sprintin katselmointi oli yhdistetty osaksi seuraavan sprintin suunnittelua kahden tunnin mittaiseksi ”demo ja sprintin suunnittelu” -nimiseksi tapahtumaksi, kuten aiemmassa sprintin suunnittelua käsittelevässä kappaleessa todettiin. Katselmointi järjestettiin aina jokaisen kahden viikon sprintin lopuksi ja mukana oli tuoteomistajan lisäksi useimmiten liiketoiminnan johtoa ja asiakkaan loppukäyttäjiä edustavia työntekijöitä. Noin tunnista puoleentoista tuntiin kestäneen katselmointitilaisuuden jälkeen koitti suoraan puolesta tunnista tuntiin kestävä seuraavan sprintin suunnittelusessio, jota varten ylimääräiset sidosryhmät päästettiin pois tapahtumasta. Tällöin paikalle jäivät ainoastaan tuoteomistaja, kehittäjät ja Scrum master.

Kuten mainittu, sekä Scrum-oppaassa että Scrum primerissa suositetaan työpajamaista ja interaktiivista lähestymistapaa sprintin katselmoinnin toteuttamisessa. Seuratussa projektissa demot kuitenkin olivat enemmän perinteisiä demoja, joissa taustayrityksen kehittäjät esittelivät vuorollaan uusia toiminnallisuuksia asiakkaalle. Osittain tätä selittää tapahtumien järjestäminen etänä. Toisaalta myös asiakkaan tuoteomistaja järjesti erikseen asiakasyrityksen sisäisiä demoja, joissa loppukäyttäjien oli mahdollista testata toiminnallisuuksia tuoteomistajan ohjauksessa omatoimisesti. Toisaalta taustayrityksessä katselmointi on virallisestikin yhdistetty demoon, kuten kuvasta 6 selviää.

Katselmointitilaisuuden demomainen luonne ei tosin ole ominaista ainoastaan seuratussa taustayritykselle, vaan myös esimerkiksi Mortada ym. havaitsivat case- ja kyselytutkimuksessaan, että toisessa seurantarhmissä sprintin katselmointitilaisuus koostui lähinnä sprintin kehitysjonon läpikäynnistä. Tätä selitettiin tuotteen kompleksisuudella. Lisäksi toisessa seuratussa tiimissä, jossa demot järjestettiin, pidettiin demot ryhmälle, joka ei ollut millään tavalla sidottu tuotteeseen – toisin sanoen demoa ei järjestetty sidosryhmille. Myös kyselyiden vastauksissa lähes puolet vastanneista totesi, ettei heidän projekteissaan sprintit pääty aikaansaannoksien demoon. (Mortada ym. 2020, 75–76.) Sprintin

katselmoinnin järjestämättä jättäminen, edes demon muodossa, tuskin on järkevää, sillä tällöin menetetään yksi ketterien menetelmien isoimmista valteista – loppukäyttäjiltä saatava, korjaava palaute kehitysprosessin aikana.

5.2.5 Sprintin retrospektiivi

Sprintin päättävän retrospektiivissä Scrum-tiimi perehtyy prosesseihinsa ja ympäristöönsä selvittäen, mikä sprintin aikana onnistui, mikä ei onnistunut ja miten mahdollisista haasteista selvittiin. Retrospektiivin enimmäispituus on kolme tuntia neljän viikon sprinteille, mutta lyhyemmillä sprinteillä myös retrospektiivit ovat vastaavasti lyhyempiä. (Sutherland & Schwaber 2020, 10.) Retrospektiivien yksityiskohtaisesta järjestämisestä Scrum on varsin vaitonainen. Scrum Primerissa annetaan kuitenkin taustakirjallisuutta retrospektiiveihin soveltuvista tekniikoista (Deemer ym. 2012, 14).

Dieboldin ym. toteuttaman haastattelututkimukseen osallistuneista yhdeksän kymmenestä vastaajasta totesi, että heidän yrityksessään järjestetään retrospektiivit. Yhdessä yrityksessä niitä ei siis järjestetty lainkaan ja toisaalta eräässä toisessa yrityksessä retrospektiivejä järjestettiin harvoin. Sprintin katselmointi järjestettiin lähes jokaisessa yrityksessä vähintäänkin samana päivänä kuin sprintin katselmointi ja näiden tapahtumien yhteiskesto oli likipitään aina yhdestä tunnista kolmeen ja puoleen tuntiin. (Diebold ym. 2015, 8.)

Hassani-Alaouin ym. case-tutkimuksessa ilmeni, että vain hieman yli puolessa seuratuissa tiimeissä sprintin retrospektiivit järjestettiin asianmukaisesti tiimin prosessien kehittämismielessä. Kahdessa tiimissä retrospektiivit järjestettiin lähinnä harmituksen purkamista varten. Toisaalta yhdessä tiimissä retrospektiivit jätettiin kokonaan välistä ylemmän johdon pitäessä niitä ajan haaskauksena (Hassani-Alaoui ym. 2020, 6264.)

Retrospektiivien kohdalla suurin osa opinnäytetyötä varten haastatelluista totesi käytäntöjen olevan vaihtelevia: osassa projekteissa retrospektiivit järjestettiin sääntillisesti, kun taas toisissa retrospektiivejä ei pidetty lainkaan. Opinnäytetyötä varten havainnoidussa asiakasprojektissa retrospektiivejä ei pidetty lainkaan kevään ja alkukesän 2024 aikana. Haastateltavat kertoivat, että tyypillisesti

pidetyissä retrospektiiveissä keskityttiin käytännötasolla Scrum-projektin toteutuksen vahvuuksiin ja ongelmakohtiin teorian mukaisesti.

Opinnäytetyötä varten haastatellut henkilöt kuvailivat pidettyjen retrospektiivien tarkoituksena olleen selvittää, että saiko asiakas haluamansa tuotokset asetettujen tavoitteiden mukaisesti. Keskustelut on käyty toisinaan sisäisesti ja toisinaan myös asiakkaan kanssa. Eräät haastateltavista mainitsivat myös eräänlaiset ”asiakashoidolliset” retrospektiivit, jotka eivät virallisesti olleet projektin retrospektiivejä, vaan ylimääräisiä tapahtumia, joissa asiakkaalta kerättiin palautetta projektin toimintamalleista ja mahdollisista ongelmakohdista. Usein tällaiset retrospektiivimäiset keskustelut on järjestetty silloin, kun projektissa on ylitetty budjetti tai jokin tavoite on jäänyt saavuttamatta.

Monet Scrum-tiimit järjestävät retrospektiivejä keskittyen ainoastaan projektin ongelmiin, minkä seurauksena monet voivat kokea retrospektiivit kielteisesti (Deemer ym 2012, 14). Yksi haastateltavista mainitsi olleensa retrospektiivissä, jossa tapahtuman painopiste oli ollut löytää syylliset projektin ongelmiin. Myös eräs toinen haastateltavista nosti esille, että erään projektin retrospektiiveissä usein keskityttiin lähinnä projektin ongelmiin, jolloin retrospektiivien täyttää potentiaalia ei välttämättä saavutettu. Retrospektiivien avulla voisi myös löytää menestystekijöitä, joita edistyksellinen Scrum master voisi välittää muiden Scrum-tiimien käyttöön organisaation sisällä.

5.3 Erot tuotoksissa

Tuotoksien kohdalla opinnäytetyötä varten haastatelluilta henkilöiltä kysyttiin alla olevat kysymykset. Kysymyksissä keskityttiin selvittämään, kenen vastuulla tuotokset ovat ja kuinka tarkasti tuotoksien laatua ylläpidetään.

- Miten tuotekehitysajono on tyypillisesti muodostettu (Mistä kehitettävät ideat syntyvät kehitysajonoon, kuinka usein kehitysajonoa päivitetään, jne.)?
- Kuka projekteissa on tyypillisesti vastannut työmääräarvioista? Miten työmääräarviot on ilmoitettu ja muodostettu? Milloin työmääräarviot yksittäisille tehtäville on annettu?

- Onko valmiin määritelmä tyypillisesti selkeästi kuvattuna kullekin tehtävälle? (Aikamääre, valmiin määritelmä, jne.)
- Onko sprintin kehitysjohto tyypillisessä projektissa tarkasti priorisoitu? Miten priorisointi on tehty ja kuka vastaa priorisoinnista projektissa tyypillisesti?
- Kuka tyypillisesti on vastannut projektissa vastuiden ja tehtävien jakamisesta kehittäjille?
- Dokumentoidaanko ratkaisut tyypillisesti? Miten?

5.3.1 Tuotteen kehitysjohto

Tuotteen kehitysjohto on vaiheittain projektin edetessä kehittyvä, järjestetty listaus tarpeista, jotka vaaditaan tuotteen kehittämiseksi. Samalla se on myös ainoa lähde Scrum-tiimin kehitystyölle. Jos tuotteen kehitysjonon kohta on sidottu tarkkaan valmiin määritelmään, voidaan kyseinen kohta katsoa valmistelluksi, jolloin se voidaan ottaa sprintin suunnittelusessiossa sprintin kehitysjohtoon. Tuotteen kehitysjonon kohtia tulisi säännöllisesti pilkkoa, tarkentaa ja järjestää. Tätä työtä kutsutaan kehitysjonon jalostamiseksi. Päävastuu tuotteen kehitysjonon ja tätä kautta tuotteen tavoitteiden saavuttamisesta on tuoteomistajalla. (Sutherland & Schwaber 2020, 11; 6.) Tuotteen kehitysjohto on siis tuoteomistajalle konkreettinen työkalu, jonka avulla hän viestii loppukäyttäjien tarpeet Scrum-tiimin kehittäjille sekä kehittäjien esittämät aika-arviot loppukäyttäjille.

Sverrisdottirin ym. pienen otannan haastattelututkimuksessa on kuitenkin ilmennyt, että tuoteomistaja ei aina ole tuotteen kehitysjonosta päävastuussa. Esimerkiksi kaksi haastelluista tuoteomistajista kertoi käyttävänsä vain noin 2–5 % ajastaan tuotteen kehitysjonon hallintaan. (Sverrisdottir ym. 2014, 262.) Tämän haastattelututkimuksen perusteella vaikuttaisi siis, että tuotteen kehitysjonon omistajuus ei aina välttämättä ole tuoteomistajalla Scrumin oppien mukaisesti. Lisäksi tuotteen kehitysjonon jalostaminen saattaa siirtyä kokonaan kehitystiimin ulkopuolelle, esimerkiksi tiimin ulkopuoliselle yrityksen keski-johdolle, kuten Hassani-Alaouin ym. tutkimuksessa selvisi (Hassani-Alaoui ym. 2020, 6262).

Wróbelin ym. kyselytutkimus antaisi ymmärtää, että tuotteen kehitysjonoon tulee toisinaan kehityskohteita muualtakin kuin tuoteomistajan toimesta (Wróbel ym. 2023, 818). Myös Masood ym. havaitsivat tutkimuksessaan, että tuotteen kehitysjonoon saattoi ilmestyä kehityskohtia useammista kanavista tuoteomistajan ohitse, esimerkiksi tukipyynnötkettien muodossa (Masood ym. 2022, 1588). Toisaalta tällaiset tukipyynnöt ja kehityskohdat saattavat ohittaa kokonaan tuotteen kehitysjonon, kuten Hassani-Alaouin ym. case-tutkimuksessa ilmeni. Lähes puolessa Hassani-Alaouin ym. seuraamissa Scrum-tiimeissä tuotteen kehitysjono ei ollut ainoa kehitystyön lähde, vaan sprintin aikana toteutettiin muitakin asiakkailta suoraan tulevia tehtäviä reaktiivisesti. Lisäksi ainakin yhdessä tiimissä tuotteen kehitysjono jäi toissijaiseksi lähteeksi tarkasti dokumentoitujen vaatimusmäärittelyjen ottaessa isomman roolin (Hassani-Alaoui ym. 2020, 6261.)

Saattaa siis hyvin olla, että vaikka tuotteen kehitysjono olisi muodostettu oikeaoppisesti Scrumin teorian mukaisesti, niin tästä huolimatta yrityksen vanhat toimintatavat ja aiemmin suurin kustannuksin tuotettu dokumentaatio ajavat Scrum-tiimin tuotteen kehitysjonon ohi. Tosin jos tuotteen kehittäminen tapahtuu jo olemassa olevaan ympäristöön tai testauskäytänteissä on puutteita, kuten sprinttejä käsittelevässä kappaleessa huomattiin, voi asiakassuhteen ylläpidon kannalta olla kriittistä, että akuutteihin ongelmiin vastataan ajallaan.

Elorannan ym. haastatteluissa selvisi, että toisinaan tuoteomistajalla ei ole aikaa tai vaadittua osaamista tuotteen kehitysjonon hallinnoimiseen. Aina tuotteen kehitysjono ei myöskään välttämättä ole järjestetty. Peräti puolet yhtä tuotteen kehitysjonoa käyttävistä tiimeistä ei käyttänyt järjestettyä tuotteen kehitysjonoa. Lisäksi haastatteluissa ilmeni, että joissakin tiimeissä tuotteen kehitysjonoa ei käytetty lainkaan, mutta yhdessä tiimissä käytettiin jopa useampaa kuin yhtä tuotteen kehitysjonoa (Eloranta ym. 2016, 197.)

Myös Mortadan ym. case- ja kyselytutkimuksen toisessa seuratuissa tiimeissä ei käytetty lainkaan tuotteen kehitysjonoa, vaan sprinteille muodostettiin aina sprintin suunnittelutilaisuudessa tyhjästä oma sprintin kehitysjono. Kyselyiden tuloksissa lähes puolet vastaajista todensivat tätä toimintatapaa, jossa sprinteillä käytetään ainoastaan sprintin kehitysjonoa, eikä tuotteen kehitysjonoa ole

lainkaan. Syynä tuotteen kehitysjonon puuttumiselle mainittiin tuoteomistajan puuttuminen ja yleinen epämääräisyys projektissa (Mortada ym. 2020, 75–76.) Tuotteen kehitysjonon puuttuminen on melko suuri puute Scrumin sovelluksessa, sillä tuotteen kehitysjonohan on konkreettinen kuvaus siitä, mitä tuote voisi lopulta mahdollisesti olla. Tuotteen kehitykseltä puuttuu toisin sanoen siis selkeä, kaikille näkyvä tavoite.

Opinnäytetyötä varten seuratussa asiakasprojektissa tuotteen kehitysjonoa pidettiin aktiivisesti mukana kehitystyön aikana. Tuotteen kehitysjojo oli sprinteille valittavan työn ensisijainen lähde, jota ylläpidettiin lähtökohtaisesti kvartaalisuunnittelutilaisuuksissa, sprintin suunnittelusessioissa ja projektissa käytetyissä pidemmissä päivittäispalaverissa tuoteomistajan ja muun kehitystiimin voimin. Vaikka tuoteomistaja hallinnoi palaverien kautta tuotteen kehitysjojoa, käytännössä tuoteomistaja ei ohjannut tuotteen kehitysjojoa projektin Jira-alustalla, vaan oli delegoinut tämän tehtävän muulle kehitystiimille taustayrityksessä osana ostettua konsultointipalvelua. Kuten Scrumin teoriassa, myös seuratussa projektissa korkeamman prioriteetin tehtävät olivat järjestetty tehtävälisauksessa ylemmäksi ja sisälsivät tarkemmat kuvaukset.

Toisinaan havainnoidun projektin tuoteomistaja huomasi tilanteita, joissa hän kaipasi lisää kehitysehdotuksia tuotteen kehitysjojoon, jolloin hän esitti taustayritykselle toiveen erillisestä työpajasta. Tällöin tuoteomistaja otti osittain myös Scrum masterin tehtäviä hoitakseen sidosryhmäyhteistyön fasilitoinnin muodossa. Workshoppien lisäksi tuoteomistaja jakoi tuotteen kehitysjojoon muodostettavia ajatuksia erillisissä sessioissa, joiden aikana hän esitteli jonkin ongelman perin pohjin kehitystiimille, joka puolestaan muodosti aiheesta kehityskohteet tuotteen kehitysjojoon. Seuratussa projektissa tuoteomistajalla oli siis selkeä omistajuus tuotteen kehitysjojosta, jota hän hallinnoi yhdessä kehitystiimin kanssa.

Opinnäytetyötä varten tehdyistä haastatteluista oli mahdollista päätellä, että osassa projekteja tuotteen kehitysjojo pidettiin aktiivisesti mukana, kun taas muissa projekteissa tuotteen kehitysjojo luotiin aluksi siihen kuitenkaan myöhemmin palaamatta. Eräs haastateltavista mainitsi, että tuotteen kehitysjojo muodostetaan projektin alussa yhdessä asiakkaan kanssa, mutta kehitysjojoon

oli tapana palata vasta siinä vaiheessa, kun työt alkoivat todennäköisesti loppua. Toinen haastateltavista kertoi lisäksi, että usein projektista on saattanut puuttua tuoteomistaja, jolloin tuotteen kehitysjonon hallinta on jäänyt löyhästi kehittäjien vastuulle. Toisaalta taas kolmas haastateltava mainitsi, että usein projektissa mukana oleva arkkitehdin nimikkeellä toimiva henkilö kampa tuotteen kehitysjonoa kerran sprinttiin läpi.

Tuotteen kehitysjono ei siis haastattelujenkaan perusteella vaikuttaisi aina olevan tuoteomistajan vastuulla ja hallinnassa, vaikka tämä olisikin tuoteomistajan yksi tärkeimmistä työkaluista tuotteen tavoitteiden viestinnässä. Kuvassa 7 on vielä taustayrityksen koulutusmateriaalien tarjoama tuotteen kehitysjonon kuvaus. Esitetty kuvaus ei poikkea virallisesta Scrumin teoriasta, mutta huomionarvoista on, ettei tuotteen kehitysjonolle ole selkeää, erikseen määriteltyä vastuuhenkilöä.

Project Backlog

In the simplest definition the Project Backlog is simply a list of all things that need to be done within the project

- Project backlog in essence is a communication tool to the team and other stakeholders.
- Every item in the Project Backlog always to add value
- The level of detail depends on the position of the entry within the Project Backlog
 - High priority items are more detailed
- Most typical backlog item is formulated as a user story
- Also bug fixes, technical improvements, user support tasks etc. are backlog items

Kuva 7. Taustayrityksen määritelmä tuotteen kehitysjonosta.

5.3.2 Sprintin kehitysjono

Sprintin kehitysjono koostaa sprintin suunnittelussa esitetyt kysymykset: Miksi sprintti tehdään, mitä sprintin aikana tehdään ja miten sprintin aikana tehtävät työt toteutetaan. Sprintin kehitysjono on kehittäjien itselleen tekemä listaus sprintin aikana tehtävistä töistä, ja sprintin kehitysjonon tarkoituksena on auttaa kehittäjiä seuraamaan edistymistään kohti sprintin tavoitetta. Täten kehitysjonon on oltava riittävän tarkka, jotta kehittäjät voivat hyödyntää sitä päivittäispalaverien statustarkastuksissa. Sprintin kehitysjono on kiinteästi sidottu sprintin tavoitteeseen, joka on saatu tuoteomistajalta sprintin suunnittelusessiossa.

Kehittäjät voivat tarvittaessa sopia tuoteomistajan kanssa sprintin kehitysjonon laajuudesta uudelleen oletuksella, ettei sprintin tavoite vaarannu. (Sutherland & Schwaber 2020, 8–9, 11.)

Siinä missä tuotteen kehitysjohto on tuoteomistajan työkalu sidosryhmien ja kehittäjien välillä viestimiseen, on sprintin kehitysjohto keino kehittäjille hallita omaa työtaakkaansa ja suoriutumistaan sprintin tavoitteiden saavuttamiseksi. Aina sprintin kehitysjohto ei kuitenkaan ole täysin kehittäjien omistuksessa.

Esimerkiksi Dieboldin ym. haastattelututkimuksessa ilmeni, että yhdessä Scrum-tiimissä sprintin kehitysjonon muodostivat ja priorisoivat tuoteomistaja, arkkitehti ja projektin johtaja ilman kehittäjien osallistumista. Kaikissa muissa tiimeissä sprintin kehitysjonon muodostaminen tehtiin kuitenkin oppikirjan mukaisesti kollektiivisesti. (Diebold ym. 2015, 8.) Myös Hassani-Alaouin ym. case-tutkimuksessa ilmeni, että selvästi yli puolissa tiimejä sprintin kehityskohteita ei valittu pelkästään kehittäjien toimesta, vaan mukana saattoi olla myös ylemmän johdon edustaja tai autoritäärinen tuoteomistaja asettamassa kohteita sprintin kehitysjohtoon. Kehitystiimin kontrollin puute voi puolestaan johtaa alentuneeseen sitoutuneisuuteen. (Hassani-Alaoui ym. 2020, 6261; 6263.)

Masoodin ym. toteuttamassa kyselytutkimuksessa selvisi vastaavasti, että toisinaan yksi henkilö – tuoteomistaja, liiketoiminnan konsultti, projektipäällikkö tai asiakas itse – saattaa valita sprintin kehitysjonon kohteet ”paremman näkyvyyden” oikeuttamana. Myös tapauksissa, joissa kehitystiimi oli osallisena sprintin kehitysjohto luomiseen, saattoi viimeinen päätäntävalta olla yhdellä henkilöllä. (Masood ym. 2022, 1578.) Näin toimiessa mahdollisesti suositaan vakautta ja varmuutta keskinäisten riippuvuuksien havaitsemisesta osan tai kaikkien muiden kehitystiimin jäsenien autonomian hinnalla. Toisaalta etenkin nuorempien kehittäjien kohdalla delegointi saattaa olla jopa välttämätöntä, jotta työpanokset kohdistetaan oikeisiin tehtäviin.

Opinnäytetyötä varten tehdyissä haastatteluissa ilmeni edellä mainitun kaltaisia tuloksia, ja yksi haastateltavista kertoi projektin arkkitehdin ja Scrum masterin olevan ensisijaisesti sprintin kehitysjonon priorisoinnista. Toisaalta yksi haastateltava esitti, että lähtökohtaisesti sprintin kehitysjonon priorisointi on

asiakkaan vastuulla ja kehittäjille varattu priorisointivastuu lähinnä häiritsee kehittäjien keskittymistä itse ongelmien ratkomiseen. Yksi haastateltavista sanoi kehittäjien olevan pääasiallisesti vastuussa sprintin kehitysjonon priorisoimisesta kokempohjaisesti, mutta toisinaan priorisointi on tehty aluksi itse, minkä jälkeen sprintin kehitysjonoon on palattu yhteisesti mahdollisten riippuvuuksien havaitsemiseksi.

Opinnäytetyötä varten seuratussa asiakasprojektissa sprintin kehitysjono muodostettiin varsin oikeaoppisesti tuoteomistajan ja kehittäjien toimesta itse itselleen. Toisaalta usein projektissa oli havaittavissa tilanteita, joissa sprintin kehitysjonon kohtia otettiin suoraan karkeina otsikkotason kuvauksina tuotteen kehitysjonosta. Toisinaan karkeat kuvaukset ja aika-arviot säilyivät sellaisinaan sprintin alusta loppuun, ja kehityskohdan tavoitteet ja edistymisen taso olivat lähinnä kehittäjien kollektiivisessa ymmärryksessä. Tämä ei tosin ole aivan epätavanomaista. Esimerkiksi Masoodin ym. haastattelututkimuksessa selvisi, että joissakin Scrum-tiimeissä aika-arviot otetaan sprintin kehitysjonon kohteisiin suoraan tuotteen kehitysjonosta ilman jatkojalostusta (Masood ym. 2022, 1584).

Opinnäytetyötä varten seuratussa asiakasprojektissa oli lisäksi mahdollista havaita, että joitakin sprintin kehitysjonoon poimittuja kehityskohteita pidettiin sprintiltä toiselle esimerkiksi tilassa ”in progress” eli työn alla. Tällä tavoin sprintin suunnittelusessiossa uusien kehitettävien kohteiden lisäksi sprintiltä toiselle siirtyi automaattisesti edellisen sprintin vanhoja tehtäviä. Tällä tavoin työskennellessä sprintin kehitysjono saattaa tosin menettää merkitystään ajantasaisena ja rajattuna kuvauksena sprintin aikana valmiiksi saatettavasta kehitystyöstä. Sprintin kehitysjononhan tulisi olla kehittäjien itselleen laatima, *reaaliaikainen* kuva sprintin aikana tehtävästä työstä (Sutherland & Schwaber 2020, 11).

Seuratussa projektissa asiakkaalle toteutettiin myös usein samalla sprintillä montaa toisiinsa liittymätöntä asiaa samanaikaisesti. Esimerkiksi projektissa saatettiin toteuttaa samalla sprintillä sekä työsuunnitteluun että tarjouksien hallintaan liittyviä komponentteja, jolloin sprintin tavoite oli sidottu kahteen toisistaan riippumattomaan tuotteeseen. Toisaalta myös Masoodin ym. haastatteluissa selvisi, että toisinaan Scrum-tiimien sprintin suunnittelussa

sprintin tavoite saattaa puuttua tai olla monitahoinen (Masood ym. 2022, 1585, 1587).

Pahimmillaan sprintin tavoite saattaa kuitenkin jäädä kokonaan määrittelemättä. Esimerkiksi Mortadan ym. yhdistetyssä case- ja kyselytutkimuksessa selvisi, että peräti yli kolmasosalla kyselyyn vastanneista ei ollut ollut tapana määritellä sprinteilleen tavoitetta (Mortada ym. 2020, 75). Myös Wróbelin ym. kyselytutkimuksessa selvisi, että sprintin tavoite saatetaan usein määritellä vasta suunnittelutilaisuuden lopussa, jos tavoitetta ylipäätään määritellään (Wróbel ym. 2023, 818). Etenkin loppukäyttäjien ja tuoteomistajan kannalta selkeän tavoitteen puuttuminen on epäedullista, koska tällöin kehitystiimin vastuuttaminen puuttuvista toteutuksista on huomattavasti vaikeampaa. Vastavuoroisesti kehitystiimi saattaa kokea epätietoisuutta – tai suunnatonta vapautta – toteuttaessaan yksittäisiä kohtia ilman ohjaavaa ydintavoitetta.

5.3.3 Inkrementit ja yksittäiset itemit

Inkrementit ovat sprinttien aikana luotavia käytännön arvoa sisältäviä osatoteutuksia, jotka rakentuvat toinen toisensa päälle, ja ollakseen hyväksyttäviä inkremeneittain tuotettujen ratkaisujen tulee olla toimivia niin yhdessä kuin erikseen. Scrum-tiimi noudattaa joko organisaation sisäistä valmiin määritelmää tai vähintäänkin omaa luomaansa valmiin määritelmää, jos organisaatiolta puuttuu määritelmä (Sutherland & Schwaber 2020, 12.) Scrum Primer tarkentaa vielä, että vastuu valmiin määritelmän noudattamisesta on tuoteomistajalla ja kehittäjillä, joista erityisesti tuoteomistajan tulisi aina vaatia mahdollisimman hyvin kuvattua valmiin määritelmää (Deemer ym. 2012, 8).

Aina valmiin määritelmä ei ole tarkoin kuvattuna yksittäisellä inkrementille varatulla tehtävällä, kuten opinnäytetyötä varten seuratussa projektissa ilmeni. Opinnäytetyötä varten havainnoidussa projektissa sprintin suunnittelussa sprintille varatut tehtävät saattoivat nojata pitkälti tuoteomistajan ja kehittäjien yhteiseen näkemykseen valmiista. Tällöin sprintin kehitysjonoon nostettavat tehtävät saattoivat olla sprintin loppuessa yhä otsikkotasolla kuvattuina, kuten ne olivat olleet tuotteen kehitysjonossa. Valmiin määritelmä oli pitkälti tuoteomistajan hyväksyntä tehdystä ja tuoteomistajalle esitellystä toteutuksesta.

Kuvassa 8 on esiteltyä taustayrityksen kuvaus valmiin määritelmästä, joka vaikuttaisikin pitkälti korostavan yhteiseen valmiin käsityksen ja testauksen merkityksiä. Tämä saattanee osaltaan selittää taustayrityksen projekteissa olevaa mallia, jossa projekteissa ei aina käytetä tarkoin dokumentoitua valmiin määritelmää. Toisaalta näin toimiessa ei sidota liikaa resursseja itse suunnittelu-prosessiin ja lukita kustannuksia mahdollisesti toteutuskelvottomaan ratkaisuun. Toisin sanoen kehitystiimi saa laajemman vapauden käyttää luovuuttaan.

Operating model "Definition of Done"

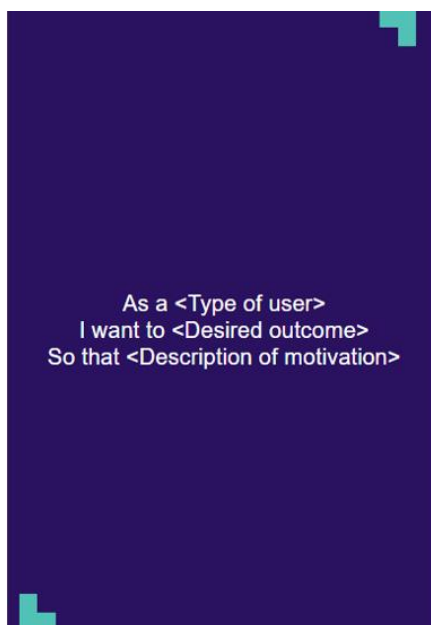
When a Project Backlog item or an Increment is described as "Done", one must understand what 'Done' means. Team members must have a shared understanding of what it means for work to be completed and to ensure transparency. This is the definition of 'Done' for the Team and it is used to assess when work is complete on the software Increment

Project XX Definition of Done:

- All acceptance criteria met
- The user story is tested
- Peer review passed
- The user story is demonstrated and approved by the customer
- Documentation completed

IMPORTANT! When a user story reaches the Done status, it CANNOT be moved back to any other status. If there is a change request or a bug we simply create a bug fix or a new user story.

Kuva 8. Taustayrityksen määritelmä valmiille eli "definition of done".



Functional Design - User stories

The functional design is usually written as user stories. They describe the deliverables from a user perspective in an understandable way to all stakeholders.

- Who the user is?
- What do they want?
- Why do they want it?

User stories often contain other types of requirements listed below:

- Acceptance criteria (what is required to be done, that the user story is considered done)
- Visual guidelines, layouts, wireframes etc.
- Technical details
- Non-functional requirements

Kuva 9. Taustayrityksen kuvaus käyttäjätarinasta. Yrityksen nimi häivytetty.

Yksi opinnäytetyötä varten haastatelluista henkilöistä kertoi, että tällainen "kollektiivinen" käsitys valmiin määritelmästä on varsin yleinen projekteissa. Haastatellun kertoman mukaan tehtävä on määritelty valmiiksi, kun toteutus on

läpäissyt sekä sisäiset että asiakkaan toteuttamat testaukset. Toinen haastateltavista kertoi samoin, että usein testauksien läpäisyä pidettiin ensisijaisena merkinä valmiista. Toisaalta kolmas haastateltava kertoi, että testauksien pohjana käytetään useimmiten käyttäjätarinoita, joiden läpäistessä testit toteutus on valmis. Kuvassa 9 on esiteltynä taustayrityksen koulutusmateriaalien pohja käyttäjätarinoille.

Käyttäjätarinat ovatkin usein vaatimusmäärittelyjen pohjana, kuten Dieboldin ym. tutkimuksessa kävi ilmi. Ainoastaan yksi tutkimuksessa seuratuista Scrum-tiimeistä noudatti Scrum-viitekehyksen mukaista valmiin määritelmää sanan varsinaisessa merkityksessä. Jos käyttäjätarinamuotoisia tehtäviä ei saatu valmiiksi sprintin aikana, Dieboldin ym. tarkkailemat tiimit noudattivat kahta erilaista toimintamallia: joko kehityskohta palautettiin kokonaisuudessaan tuotteen kehitysjonoon tai sitten kehityskohta pilkottiin sprintin aikana valmistuvaan ja seuraavalle sprintille siirtyvään osaan. (Diebold ym. 2015, 9.)

Tavoista ensimmäinen noudattaa oikeaoppista Scrumia, jälkimmäinen ei. Jos kehityskohta ei täyty valmiin määritelmää, kohtaa ei julkaista tai edes esitellä sprintin katselmoinnissa, vaan sen sijaan kohta palaa kokonaisuudessaan tuotteen kehitysjonoon (Sutherland & Schwaber 2020, 12). Opinnäytetyötä varten seuratussa projektissa oli kuitenkin havaittavissa yllä mainittua jälkimmäistä toimintaa. Joissakin sprintin katselmoinneissa saatettiin esitellä keskeneräisiä toiminnallisuuksia, joiden jatkokehitys siirrettiin suoraan seuraavalle sprintille. Projektissa ei kuitenkaan tehty Dieboldin ym. kaltaista "pilkkomista", vaan tehtävät siirtyivät joko sellaisenaan tai jatkojalostettuna seuraavalle sprintille.

Yksi selittävä tekijä tehtävien jäämiselle yli sprinteiltä voi olla tehtävien puutteelliset työmääräarviot, jotka Scrumin teorian mukaan tulisi kuitenkin olla aina olemassa. Julkaistavan tuotoksen jokaisen kehityskohdan tulisi sisältää työmääräarvio (Deemer ym 2012, 7). Nämä arviot tulevat kehittäjien toimesta (Sutherland & Schwaber 2020, 11). Opinnäytetyötä varten seuratussa projektissa oli toisinaan tilanteita, joissa tuotteen kehitysjonon kohdilta puuttuivat kokonaan työmääräarviot, jotka annettiin vasta sprintin kehitysjonon tehtäville. Näin toimiessa toiminnan budjetointi ja ennustaminen on oletettavasti huomattavasti vaikeampaa, mutta samalla kehitystiimi ei välttämättä lukkiudu liikaa aika-

arvioihin, jotka saattavat olla pahimmillaan vain parhaita arvauksia. Eräs haastateltavista mainitsikin, että toisinaan aika-arviot tehtäville saattavat olla hyvin lähellä toteumaa, kun taas toisinaan aika-arviot voivat olla todella pielessä.

Työmääräarvioiden puutteellisuus ei tosin vaikuttaisi olevan ominaista vain taustayritykselle. Esimerkiksi Mortadan ym. case- ja haastattelututkimuksessa ilmeni, että toisessa tiimeistä sprintin kehitysjonoon poimituilta kohteilta puuttui kokonaan aika-arviot ja lisäksi kummassakin tiimissä sprintin kehitysjonon kohteita hyödynnettiin usein pelkällä otsikkotasolla lyhyen kuvauksen kera. Kyselytutkimuksen vastaukset kumpikin toistelivat seuratuissa ryhmissä tehtyjä havaintoja (Mortada ym. 2020, 75.) Myös Wróbelin ym. haastatteluista ilmeni, että toisinaan tehtävien tarkemmat aika-arviot saattavat jäädä tekemättä (Wróbel ym. 2023, 818). Elorannan ym. haastatteluissa selvisi, että parissa Scrum-tiimissä ei käytetty tuotteen kehitysjonon tehtäville lainkaan työmääräarvioita, mutta lisäksi kuudessa tiimissä työmääräarviot tulivat pelkästään tuoteomistajan toimesta (Eloranta ym. 2022, 197). Myös Masoodin ym. haastattelututkimuksessa paljastui, että toisinaan työmääräarvioita ei toteuteta kollektiivisesti kehittäjien toimesta (Masood ym. 2022, 1584).

Kehitettäviltä kohteilta saattaa siis usein puuttua tarkat kuvaukset ja aikamääreet, mutta lisäksi on mahdollista, että aika-arviot tulevat kokonaan muiden kuin kehittäjien toimesta. Kumpikaan tilanne ei ole Scrumin kannalta ihanteellinen, sillä ensimmäisessä tilanteessa kärsii kehittäjien ja sidosryhmien tilanne-tietoisuus ja jälkimmäisessä kärsii kehittäjien autonomia. Kuitenkin tällainen toimintatapa voi olla kenties ainoa mahdollinen, jos toteutetaan jotakin aivan uudenlaista tuotetta, jolle ei ole vertailukohdetta ja joka toisaalta vaatii korkeampaa ylätasoa näkemystä tuotteen lopullisesta muodosta ja käytettävissä olevista resursseista.

6 TULOKSET JA ANALYSOINTI

Taulukkoihin 1, 2 ja 3 on koottu poikkeamat, joita Scrumin teorian ja käytännön sovelluksien välillä on voitu havaita. Seuraavana on vielä avattu taulukoissa havaittuja eroja sekä alussa esitettyjen hypoteesien ja tuloksien vertautumista. Hypoteesien kohdalla esitin, että Scrumin teoria todennäköisesti eroaa käytänteistä. Kootuista tuloksista voidaan havaita, että näin todella on.

Roolituksien osalta esitin hypoteeseissa, että erikseen nimetty Scrum master saattaa projekteista puuttua. Näin tosiaan vaikuttaisi olevan, sillä usein Scrum masterin rooli saatetaan yhdistää projektipäällikön rooliin tai jättää kokonaan täyttämättä. On myös mahdollista, että Scrum masterin roolin ottaa hoitaakseen kehittäjä tai kehittäjien ryhmä kiertävästi. Samalla paljastui, että kokonaan puuttuvan Scrum masterin lisäksi projekteissa voi olla useampi kuin yksi Scrum master. Toisaalta myös ilmeni, että joissakin tapauksissa saattaa olla erikseen nimetty Scrum master, joka on kuitenkin samanaikaisesti useammassa projektissa Scrum masterina.

TAULUKKO 1. Havaittuja eroavaisuuksia roolituksissa.

	Kehittäjät ja tiimin koko	Tuoteomistaja	Scrum master
Vastuiden sekoittuminen	- Usein kehittäjien joukosta lisäksi Scrum master ja toisinaan tuoteomistaja	- Toisinaan delegoi ”liikaa” tehtäviään (esimerkiksi tuotteen kehitysjonon pois)	- Joskus yhdistetty projektipäällikön rooliin - Toisinaan kehittäjä (myös kehittäjien kesken kiertävä järjestelmä)
Määrä	- Määrä usein viitekehyksen mukainen, joskus kuitenkin liian pieniä tai suuria - ”Yhden miehen Scrum”	- Voi olla useampi kuin yksi - Joskus voi puuttua kokonaan	- Joskus useampi kuin yksi - Joskus voi puuttua kokonaan
Muuta	- Monialaisuuden sijasta asiantuntijatiimejä - Kehittäjillä saattaa olla Scrumin ulkopuoliset roolit - Kehittäjillä saattaa olla työrauhan rikkovia velvoitteita päällekkäin - Tiimit eivät aina ole vakaita	- Aina tuoteomistaja ei nauti taustaorganisaatioiden arvostusta - Toisinaan saattaa toimia myös Scrum masterina	- Joskus yksi Scrum master on jaettu useamman tiimin kesken, suositus yhdestä

Scrum master ei tosin ollut ainoa rooli, jonka kohdalla teoriasta nähtävästi poiketaan. Myös Scrumin kannalta kriittisen tuoteomistajan roolissa oli havaittavissa poikkeamia. Esimerkiksi tuoteomistaja saattaa delegoida

avaintehtäviään, kuten tuotteen kehitysjonon hallinnan, toisille tiimin jäsenille joko aika- tai kyvykkyyksirajoitteiden takia. Vastaavasti tuoteomistaja saattaa myös välillä hoitaa Scrum masterin tehtäviä, mikä on täysin vastoin Scrumin teoriaa. Toisaalta myös saattaa olla tilanteita, joissa tuoteomistajan rooli on jaettu useammalle kuin yhdelle henkilölle vastoin teorian esittämää vaatimusta yhdestä tuoteomistajasta. Vaikuttaisi myös siltä, että usein tuoteomistaja rooli jää kokonaan täyttämättä, mikä voi yhtä lailla saattaa tuotteen laadunvalvonnan ja tavoitteiden saavuttamisen vaaraan. Toisaalta saattaa olla myös tilanteita, joissa tuoteomistaja ei nauti taustaorganisaationsa luottamusta esimerkiksi hierarkioiden ja byrokratian seurauksena.

Myös kehittäjiä ja tiimin kokoon liittyen oli mahdollista havaita poikkeamia Scrumin teoriasta. On hyvin mahdollista, että kehittäjät saattavat omien tehtäviensä lisäksi joutua ottamaan työpöydälleen myös Scrum masterin tai tuoteomistajan tehtäviä, mikä ei sinänsä ole teorian kanssa ristiriidassa, vaikkakaan ei täysin teorian mukaan suositeltavaa toimintaa. Lisäksi kehittäjiillä saattaa olla usein päällekkäisiä projekteja ja muita velvoitteita, jotka haittaavat kehittäjiä keskittymistä aiheuttaen viivästyksiä sprinttien kehitystyöhön. Scrumin sisäisten roolien lisäksi kehitystiimin jäsenillä saattaa olla myös tarkemmat roolitukset, kuten "lead developer", "backend developer" tai "lead consultant", mikä ei vastaa täysin teoriaa, joka viittaa kehittäjiin puhtaasti Scrum-tiimin jäseninä. Scrum-tiimin pitäisi olla monialainen, toisten vahvuuksia täydentävä kokonaisuus osajia, jolloin erillisille nimikkeille ei olisi tarvetta. Teoriasta poiketen tiimit saattavatkin tosin olla usein erityisen osaamisen hallitsevia asiantuntijatiimejä. Tällöin tiimeihin saatetaan toisinaan tarvita ulkopuolista apua, joka rikkoo puolestaan Scrumin suositusta vakaasta tiimistä. Scrum-tiimit saattavat myös toisinaan olla vain "yhden miehen tiimejä" tai sitten yli kokosuosituksien eli yli kymmenen hengen kokonaisuuksia.

Tapahtumien kohdalla esitin hypoteeseina, että eroja löytyy ja esimerkiksi päivittäispalavereiden kohdalla saatetaan lipsua. Näin tosiaan vaikuttaisi olevan ja monesti päivittäispalaverit saatetaan järjestää pidempinä ja harvemmin kuin kerran päivässä 15 minuuttia kerrallaan. Toisaalta päivittäispalaverit eivät aina ole välttämättä pelkästään statuspäivityksiä varten, vaan päivittäispalavereissa saatetaan käydä lisäksi tarkemminkin tuotteen kehittämiseen liittyviä aiheita. On

myös mahdollista, ettei statuksia käydä läpi lainkaan, vaan päivittäispalaveri on varattu lähinnä yhteisten asioiden läpikäyntiä varten.

TAULUKKO 2. Havaittuja eroavaisuuksia tapahtumissa.

	Sprintti	Sprintin suunnittelu	Päivittäispalaveri	Sprintin katselmointi	Sprintin retrospektiivi
Erot pituuksissa	- Ei havaittuja poikkeamia pituudessa - Joskus käynnissä olevaa sprinttiä jatketaan	- Joskus lyhyempiä tai pidempiä kuin ohjeistettu	- Joskus pidempiä kuin ohjeistettu	- Usein aikarajoissa pysytty	- Ei havaittuja poikkeamia pituudessa
Erot intervalleissa	- Ei aina vakioittainen sprinttiväli	- Lähtökohtaisesti teorian mukaisia	- Saattavat olla teoriaa harvemmin	- Joskus jätetään täysin pitämättä	- Ei aina pidetä
Muuta	- Toisinaan dokumentointi ja "häntiä" saattaa jäädä sprintin lopun jälkeen - Joskus sprinttejä toteutetaan ilman valmista tuotetta lopussa - Joskus sprintit jäävät kokonaan toteuttamatta	- Aina ei vastata teorian esittämiin kysymyksiin, vaan edetään suoraan toteutuksen tehtäviin - Joskus kehittäjät eivät itse valitse tehtäviään - Toisinaan ei määritetä käytössä olevia resursseja	- Joskus päivittäis-palavereita ei järjestetä lainkaan - Statuksen lisäksi muutakin keskustelua - Aina status-päivitystä ei ole, vaan tilaisuudessa käydään läpi yleisiä asioita - Joihinkin kysymyksiin ei vastata	- Työpajamaisen tapahtuman sijasta demo - Suunnittelu-tapahtumaan yhdistäminen - Toisinaan erikseen sisäinen ja ulkoinen	- Usein keskittyminen negatiiviseen

Lisäksi myös muissa tapahtumissa oli havaittavissa eroja. Esimerkiksi sprintin suunnittelusessiot saattavat olla toisinaan teorian esittämien aikarajoitteiden ylittäviä tai alittavia. Lisäksi sprintin suunnittelu saattaa toisinaan olla vain tehtävien delegoimista varten ilman sen tarkempaa sitoutumista sprintin tavoitteeseen. Aina tehtävien valitseminen ei mene Scrumin ohjeiden mukaisesti kehittäjävetoisesti, vaan tehtävät delegoidaan jonkin ylemmän tahon toimesta.

Sprintin katselmointitilaisuuksien osalta ei ollut havaittavissa merkittäviä eroja aikarajoitteissa pysymisen sijaan. Sen sijaan sprintin katselmointien luonne ei aina vaikuta vastaavan Scrumin teorian pyrkimystä työpajasta. Taustayrityksessä oli havaittavissa katselmoinnin pelkistymistä demoksi, jossa sidosryhmien rooli oli passiivinen suhteessa kehittäjiin. Toisaalta katselmointi saatettiin myös yhdistää sprintin suunnitteluun tai järjestää kahdessa osassa, erikseen sisäisesti ja ulkoisesti. Kaksi edellä mainittua eivät sinänsä ole teorian kanssa ristiriidassa, mutta ovat eräänlaisia käytännön sovelluksia teoriasta.

Sprintin retrospektiivit saattavat toisinaan jäädä järjestämättä tai niitä järjestetään vähintäänkin epäsäännöllisesti. Lisäksi vaikuttaisi siltä, että retrospektiiveissä keskitytään pelkästään Scrum-tiimin kohtaamiin ongelmiin ja haasteisiin. Tällöin retrospektiivit jäävät helposti kielteisenä koetuksi tapahtumaksi, eivätkä mahdollisuuksiksi kehittyä ja tehostaa Scrum-tiimin toimintaa.

Itse muiden tapahtuvien ylle rakentuvien sprinttien käytänteissä oli myös havaittavissa jonkinasteista eroa teorian ja käytännön välillä. Sprintin pituudet olivat poikkeuksetta alle kuukauden mittaisia ja tyypillisesti kahden viikon mittaisia, joten aikarajoitteissa pysyttiin. Toisinaan kuitenkin teoriasta saatetaan poiketa venyttämällä käynnissä olevia sprinttejä tai vuorottelemalla erimittaisia sprinttejä. Toisaalta sprintin aikana kaikkia tehtäviä ei välttämättä saada valmiiksi, vaan tiimit saattavat ottaa erillisiä dokumentaationsprinttejä tai järjestää dokumentaation toisen sprintin aikana. Aina sprintin lopussa ei synny valmista inkrementtiä, mutta tästä huolimatta keskeneräinen tuote esitellään sprintin katselmoinnissa. Harvoissa tapauksissa saattaa myös olla, ettei Scrumia soveltava tiimi käytä lainkaan sprinttejä.

Tuotoksien kohdalla esitin hypoteeseissa, että valmiin määritelmän noudattamisessa saattaa olla puutteita, mikä vaikuttaisi pitävän paikkansa. Lisäksi esitin, ettei artefakteissa olisi välttämättä suurta eroa teorian ja käytännön välillä, mutta tämä hypoteesini vaikuttaisi osoittautuneen vääräksi.

TAULUKKO 3. Havaittuja eroavaisuuksia tuotoksissa.

	Tuotteen kehitysjono	Sprintin kehitysjono	Inkrementit ja yksittäiset itemit
Eroja	<ul style="list-style-type: none"> - Ei aina pelkästään tuoteomistajan hallussa - Toisinaan tuotteen kehitysjono "unohdetaan" - Joskus useampi kuin yksi tuotteen kehitysjono - Joskus työmääräarviot puuttuvat - Joskus ei ole tuotteen kehitysjonoa alkuunkaan - Joskus tuotteen kehitysjonon tavoite puuttuu 	<ul style="list-style-type: none"> - Ei aina kehittäjien toimesta muodostunut - Joskus hienojakoistuminen jää puolitiehen - Toisinaan sprintin kehitysjonoon ilmestyy sprintin ulkopuolisia asioita - Joskus otettu "stretch taskeja", jotka voidaan tehdä, jos ehditään - Toisinaan sprintin kehitysjonossa saattaa olla tehtäviä, joiden tavoitteet eivät kytkeydy toisiinsa 	<ul style="list-style-type: none"> - Aina valmiin määritelmää ei noudateta -> Kollektiivinen käsitys valmiista, testaukset - Tehtäviä saatetaan jakaa seuraavalle sprintille - Joskus kehittäjät eivät itse valitse tehtäviään ja anna näille aika-arviota

Vaikuttaisi siltä, että usein valmiin määritelmä saattaa rakentua Scrum-tiimin kollektiivisesta käsityksestä siitä, mitä valmis voisi olla. Tämä valmiin määritelmä nojaa pitkälti oletuksiin samanlaisesta, jaetusta käsityksestä valmiista sekä

käyttäjien tekemiin testauksiin. Tässä saattaa piillä vaaranpaikka virheellisille oletuksille ja kommunikaatiokatkoksille. Toisaalta yksittäiset tehtävät eivät aina ole Scrumin teorian mukaisesti kehittäjien itse itselleen valitsemiaan, vaan jonkin toisen delegoimia. Vastaavasti myös aika-arviot saattavat tulla muiden kuin kehittäjien toimesta.

Toisaalta tuotteen kehitysjonon kohdista saattaa puuttua kokonaan aika-arviot. Tällaiset tehtävät saattavat siirtyä tuotteen kehitysjonosta sellaisenaan sprintin kehitysjonoon ilman jatkojalostusta, jolloin sprintin kehitysjono ei enää olekaan Scrumin teorian esittämä reaaliaikainen kuvaus sprintin etenemisestä. Samalla jotkut tehtävät saattavat jäädä sprintin kehitysjonoon yli sprinttien, mikä entisestään horjuttaa sprintin kehitysjonon asemaa todenaikaisena kuvana sprintin aikana tehtävistä ja valmistuvista tehtävistä. Pahimmillaan sprintin kehitysjonon kohdat eivät ole edes kehittäjien itse itselleen valitsemia, vaan ovat jälleen jonkin toisen tahon delegoimia sprintille. Aina sprintin tavoitteet eivät kehitysjonon sisällön osalta kytkeydy toisiinsa ja saattavat myös olla luonteeltaan sellaisia, että tehtävät tehdään, jos niihin jää aikaa.

Tuotteen kehitysjonon kohdalla saattaa toisinaan käydä niin, että kehitysjono muodostetaan seremoniaalisesti työpajojen avulla, mutta unohdetaan kuitenkin taustalle. Toisaalta saattaa myös olla, että kehitysjono ei unohdu taustalle, mutta ei kuitenkaan ole tuoteomistajan hallussa, kuten teoriassa esitetään. Lisäksi saattaa olla tilanteita, ettei kehitysjonoa ole lainkaan tai sitten kehitysjono on heikosti priorisoitu. Erittäin poikkeuksellisesti on myös havaittu Scrum-tiimi, jossa on ollut käytössä useampi kuin yksi tuotteen kehitysjono, vaikka Scrumin teoria nimenomaan esittää tuotteen kehitysjonon yhtenä ainoana lähteenä kaikelle tuotteen eteen tehtävälle työlle. On kuitenkin myös havaittu, että aina tuotteen kehitysjono ei ole ainoa sprintin aikana tehtävän työn lähde, vaan toisinaan aika-kriittiset, asiakkailta tulevat tehtävät ja raskaat vaatimusmäärittelydokumentit saattavat ajaa tuotteen kehitysjonon ohi tehtävän työn lähteenä.

Voidaan siis todeta, että Scrumin sovelluksissa on merkittäviä eroja teorian ja käytännön välillä. Osa eroavaisuuksista on pienempiä poikkeamia suosituksista, kun taas toiset ovat selkeitä teorian rikkomuksia. Havaittujen poikkeamien tuomien mahdollisten hyötyjen ja haittojen vertailu vaatisi tosin jatkotutkimusta.

7 POHDINTA

7.1 Tutkimuksen rajoitteet ja heikkoudet

Tutkielman suurimpana rajoitteena voidaan pitää pientä otantaa aiheesta kaikilla menetelmillä. Kirjallisuuskatsauksen kohdalla yhtenä heikkoutena voidaan lisäksi mainita kirjallisuuskatsauksen epäsystemaattisuus. Kirjallisuuskatsaus on toteutettu kattavamman systemaattisen katsauksen sijasta kuvailevana yleiskatsauksena. Tämän seurauksena kaikkea mahdollista tutkimustietoa ei välttämättä ole tavoitettu tutkielmaa varten.

Opinnäytetyössä käytettyjen työelämälähteiden kohdalla selkeä rajoite on vain yhden yrityksen ottaminen mukaan tutkielmaan. On ilmiselvää, ettei yhden yrityksen poikkeamia Scrumin teoriasta voida induktiivisesti yleistää koskemaan kaikkia Scrumia soveltavia yrityksiä. Lisäksi projektiseurannan rajoitteena voidaan mainita, että taustayrityksen sisällä seurattiin ainoastaan yhtä projektia, joten tuloksia ei välttämättä voida kovin suurella luottamuksella yleistää koskemaan yrityksen toimintaa sisäisestikään. Toisaalta taustayrityksen työntekijöiden haastatteluista on ollut mahdollista saada jonkinlaista tukea seuratussa projektissa tehdyille havainnoille ja päinvastoin.

Haastattelujen osalta täytyy myös huomioida, että haastattelijan laatijana ja suorittajana toimi yksittäinen opiskelija, jolta puuttuu kattava kokemus tutkimuksellisten haastattelujen tekemisestä. Näin ollen haastattelut, vaikkakin suuntaa antavia ja mahdollisesti nauhoitettuna ja litteroituina hyödynnettävissä olevia, eivät välttämättä edusta laadukkainta otosta työelämän edustajien kokemuksista Scrumin poikkeamista. Lisäksi haastatteluihin oli taustayrityksestä saatu vain viisi henkilöä, mikä tarkoittaa jälleen äärimmäisen pientä otantaa kaikista Scrum-projekteissa olleista sekä yrityksen sisällä että kaikista maailman Scrum-projekteihin osallistuneista henkilöistä.

Vaikka haastattelut toisaalta olisikin tehty ammattimaisemmin, on laadullisissa haastatteluissa aina haasteena ihmisten subjektiivisten kokemusten ja muistin rajallisuuden heijastuminen haastattelujen vastauksiin. Samalla haastateltavien

antamiin vastauksiin voi vaikuttaa haastateltavien ja haastattelijan väliset sosiaaliset suhteet haastattelijan ollessa haastateltavien kollega samassa työyhteisössä.

Tutkimusmenetelmien kohdalla opinnäytetyön yhtenä haasteena voidaan yleisestikin mainita opinnäytetyön nojautuminen puhtaasti laadullisiin menetelmiin, eikä tehtyjä havaintoja ole kvantifioitu millään tavalla. Määrällisen tiedon esittäminen olisi helpottanut Scrum-projekteissa tapahtuvien poikkeamien arvioinnissa. Opinnäytetyö antaa sinänsä mielenkiintoisia havaintoja Scrumin soveltamisesta työelämässä kvalitatiivisena tutkielmana, mutta jälkikäteen tiedon esittäminen kvantitatiivisesti olisi antanut kenties selkeämmän läpileikkauksen havaittujen poikkeamien yleisyydestä.

Viimeisenä rajoitteena voidaan vielä mainita opinnäytetyön taustayrityksen Scrumin soveltaminen, sillä taustayrityksessä joistakin Scrumin teorian vaatimuksista poiketaan systemaattisesti ja tietoisesti – esimerkkeinä Scrum masterin puuttuminen, kuten kuvasta 5 nähdään, tai sprintin katselmoinnin demomaisuus, kuten kuvasta 6 huomataan. Toisaalta opinnäytetyössä tarkoituksena olikin selvittää juuri, miten teoriasta mahdollisesti poiketaan ja samalla taustayrityksen työvoima on Scrumin osalta suurelta osin virallisesti sertifioitua. Toisin sanoen teorian soveltaminen ei ole täysin sattumanvaraista.

7.2 Mahdollinen jatkotutkimus

Tässä opinnäytetyössä keskityttiin pelkästään havaittuihin eroihin Scrumin teorian ja käytännön välillä menemättä syvemmin juurisyihin eroissa. Mahdollisten eroavaisuuksien kohdalla tutkielmassa vastattiin lähinnä kysymykseen ”miten”, eikä niinkään ”miksi”. Voisikin olla hyvä selvittää syvemmin, miksi Scrumin teoriaa sovelletaan poikkeavasti. Tämän tiedon avulla voitaisiin puolestaan paremmin arvioida, ovatko poikkeamat perusteltavissa ja millaisia hyötyjä ja haittoja näillä voitaisiin saavuttaa. Tämä taas voisi edistää Scrumin teorian kehittymistä ja soveltamista työelämässä tehokkaammin.

LÄHTEET

Deemer, P., Benefield, G., Larman, C. & Vodde, B. 2012. The Scrum Primer – A Lightweight Guide to the Theory and Practice of Scrum (Version 2.0). Viitattu 16.05.2024. https://Scrumprimer.org/Scrumprimer20_small.pdf

Diebold, P., Ostberg, J.P., Wagner, S. & Zender, U. 2015. What Do Practitioners Vary in Using Scrum? Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds) Agile Processes in Software Engineering and Extreme Programming. XP 2015. Lecture Notes in Business Information Processing, vol 212. Springer, Cham. Viitattu 16.05.2024. https://doi.org/10.1007/978-3-319-18612-2_4

Digital.ai 2023. 17th State of Agile Report. Digital.ai. Viitattu 07.06.2024. <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>

Eloranta, V.P., Koskimies, K. & Mikkonen, T. 2016. Exploring ScrumBut—An empirical study of Scrum anti-patterns. Information and Software Technology. Vol. 74, 2016, 194-203. Viitattu 14.06.2024: <https://doi.org/10.1016/j.infsof.2015.12.003>

Ereiz, Z. & Mušić, D. 2019. Scrum Without a Scrum Master. 2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI). Viitattu 16.05.2024. <https://ieeexplore.ieee.org/document/8938877>

Hassani-Alaoui, S., Cameron, A. & Giannelia, T. 2020. "We Use Scrum, but ...": Agile Modifications and Project Success. Proceedings of the 53rd Hawaii International Conference on System Sciences. Viitattu 23.08.2024. <http://hdl.handle.net/10125/64507>

Kadenic, M.D., Koumaditis, K. & Junker-Jensen L. 2023. Mastering scrum with a focus on team maturity and key components of scrum. Information and Software Technology, 2023, vol. 153, iss. 107079. Viitattu 23.08.2024. <https://doi.org/10.1016/j.infsof.2022.107079>

Lahti, J.D., Tuovinen, A., Mikkonen, T. & Capilla, R. 2022. ScrumBut as an Indicator of Process Debt. 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Gran Canaria, Espanja, 2022, pp. 318-321. Viitattu 23.08.2024. <https://doi.org/10.1109/SEAA56994.2022.00057>

Masood, Z., Hoda, R. & Blincoe, K. 2022. Real World Scrum – A Grounded Theory of Variations in Practice. IEEE Transactions on Software Engineering, 2022-05, vol. 48 (5). Viitattu 03.05.2024. <https://ieeexplore.ieee.org/document/9201058>

Mortada, M., Ayas, H.M. & Hebig, R. 2020. Why do Software Teams Deviate from Scrum?: Reasons and Implications. ICSSP '20: Proceedings of the International Conference on Software and System Processes, pages 71 – 80. Viitattu 26.08.2024. <https://doi.org/10.1145/3379177.3388899>

Noll, J., Razzak, M.A., Bass, J.M. & Beecham, S. 2017. A Study of the Scrum Master's Role. Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M.,

Sarro, F., Winkler, D. (eds) Product-Focused Software Process Improvement. PROFES 2017. Lecture Notes in Computer Science(), vol. 10611. Springer, Cham. Viitattu 24.06.2024. https://doi.org/10.1007/978-3-319-69926-4_22

Paasivaara, M., Durasiewicz, S. & Lassenius, C. 2008. Using scrum in a globally distributed project: a case study. Vol. 13, Iss. 6 Special Issue: Global Software Development: Where Are We Headed? Marraskuu/Joulukuu 2008: 527-544. Viitattu 26.08.2024. <https://doi.org/10.1002/spip.402>

Paulk, M. 2013. A Scrum Adoption Survey. Software Quality Professional; Milwaukee vol. 15, Iss. 2, (Maaliskuu 2013): 27-34. Viitattu 16.05.2024. <https://www.proquest.com/docview/1433123811?pq-origsite=primo&sourcetype=Scholarly%20Journals>

Salesforce 2024. Salesforce Ranked #1 CRM Provider for the 11th Consecutive Year. Salesforce. Viitattu 11.06.2024. <https://www.salesforce.com/news/stories/idc-crm-market-share-ranking-2024/#:~:text=Salesforce%20Ranked%20%231%20CRM%20Provider%20for%2011th%20Consecutive%20Year%20%2D%20Salesforce>

Salimi, S. n.d. The origins of Scrum. Agile Academy. Viitattu 03.05.2024. <https://www.agile-academy.com/en/foundations/the-origins-of-Scrum/#:~:text=Scrum%20is%20not%20an%20acronym,Harvard%20Business%20Review%20in%201986>.

Sutherland, J., & Schwaber, K. 2020. Scrum-opas – Scrumin määritelmä ja pelisäännöt. Viitattu 03.05.2024. <https://Scrumguides.org/docs/Scrumguide/v2020/2020-Scrum-Guide-Finnish.pdf>

Stray, V., Sjøberg, D.I.K. & Dybå, T. The daily stand-up meeting: A grounded theory study. Journal of Systems and Software vol. 114, huhtikuu 2016, sivut 101-124. Viitattu 06.06.2024. <https://www.sciencedirect.com/science/article/pii/S0164121216000066?via%3Dihub>

Takeuchi, H. & Nonaka, I. 1986. The New New Product Development Game. Harvard Business Review. Viitattu 03.05.2024. <https://hbr.org/1986/01/the-new-new-product-development-game>

Wróbel, M.R., Przała, D., & Weichbroth, P. 2023. Exploring the Prevalence of Anti-Patterns in the Application of Scrum in Software Development Organizations. 18th Conference on Computer Science and Intelligence Systems (FedCSIS), Varsova, Puola, 2023, pp. 813-822. Viitattu 26.08.2024. <http://dx.doi.org/10.15439/2023F9562>