



Kandidutkielma

Tietojenkäsittelytieteen kandiohjelma

Tuoteomistajan rooli Scrumissa

Riikka Salojoki

20.12.2022

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
HELSINGIN YLIOPISTO

Yhteystiedot

PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

Sähköpostiosoite: info@cs.helsinki.fi
URL: <http://www.cs.helsinki.fi/>

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen kandiohjelma	
Tekijä — Författare — Author			
Riikka Salojoki			
Työn nimi — Arbetets titel — Title			
Tuoteomistajan rooli Scrumissa			
Ohjaajat — Handledare — Supervisors			
Toht. Kai-Kristian Kemell			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidutkielma	20.12.2022	23 sivua	
Tiivistelmä — Referat — Abstract			
<p>Ketteristä ohjelmistokehitysmenetelmistä suosituimmaksi kasvanutta Scrumia sovelletaan eri tavoin eri organisaatioissa. Myös tuoteomistajan rooli Scrumissa vaihtelee sekä akateemisessa- että yritysmaailmassa eikä roolille ole muodostunut vakiintunutta kehystä. Samaan aikaan ollaan myös tehty havainto tuoteomistajan tärkeästä roolista Scrumin kontekstissa. Tässä kandidutkielmassa kartoitetaan mikä on tuoteomistajan rooli Scrumissa. Tämä selvitetään tutkimalla tuoteomistajan määritelmää ja tehtäviä Scrumin kontekstissa.</p> <p>Tutkimus on suoritettu kirjallisuuskatsauksena, jonka perusteella tuoteomistajan rooli on merkittävä Scrum-projektin onnistumisen kannalta. Tuoteomistaja edustaa asiakasta ja muita sidosryhmiä Scrum-tiimille sekä hoitaa vaativia tehtäviä, jotka parantavat Scrum-tiimin työn tuottavuutta ja tehokkuutta. Näiden avulla voidaan varmistaa, että asiakas saa maksimaalisen arvon tuotteesta, mikä on yksi Scrum-projektin menestystekijöistä. Onnistuakseen tehtävässään tuoteomistajan tulee ansaita luottamus projektin eri osapuolilta. Tämä voidaan saavuttaa tuntemalla oma toimialue, kommunikoimalla ja tekemällä yhteistyötä muiden osapuolien kanssa, luomalla empatiaa heidän välillä sekä tekemällä päätöksiä pohjautuen tuotteen visioon.</p>			
<p>ACM Computing Classification System (CCS) Software and its engineering Agile software development</p>			
Avainsanat — Nyckelord — Keywords			
ketterä ohjelmistokehitys, ketterät menetelmät, scrum, tuoteomistaja			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsingin yliopiston kirjasto			
Muita tietoja — övriga uppgifter — Additional information			

Sisällys

1	Johdanto	1
2	Scrum ketterässä ohjelmistokehityksessä	2
2.1	Ketterä ohjelmistokehitys	2
2.2	Scrum	4
2.2.1	Scrumin taustaa	4
2.2.2	Scrumin määritelmä	4
2.2.3	Scrumin tapahtumat ja tuotokset	6
3	Tuoteomistaja Scrumissa	8
3.1	Tuoteomistajan määritelmä	8
3.2	Tuoteomistajan ominaisuudet	9
3.3	Tuoteomistajan tehtävät	11
3.3.1	Vaatimusten määrittely ja tuotevision rakentaminen	12
3.3.2	Julkaisun suunnittelu	13
3.3.3	Vaatimusten kirjoittaminen	14
3.3.4	Tuotteen kehitysjonon hallinta	16
3.3.5	Osallistuminen ja yhteistyö Scrumin tapahtumissa	18
4	Yhteenveto	20
	Lähteet	21

1 Johdanto

Ketterät menetelmät ovat saaneet valtavirran hyväksynnän ohjelmistokehitysyhteisössä ja useiden tieteellisten lähteiden perusteella niistä on kehitetty erilaisia viitekehyksiä. Näitä ovat esimerkiksi Extreme Programming (XP), Dynamic Systems Development Method (DSDM), Crystal Methods ja Scrum. Näistä Scrum vaikuttaa olevan tähän asti kaikista suosituin ketterä menetelmä. Scrum on tarjonnut erityisesti yrityksille mahdollisuuden parantaa tuotteiden laatua, läpinäkyvyyttä, kommunikaatiota, asiakastyytyväisyyttä ja pienentää kustannuksia, mikä on vaikuttanut positiivisesti yrityksen liiketoiminnalliseen tulokseen ja edesauttanut arvon tuottamista asiakkaalle (Paasivaara et al., 2009). Tästä johtuen on hyvä kartoittaa vaatimuksia sille, miten yritys voi onnistua Scrum-projektissaan.

Pichler (2010), Sverrisdottir et al. (2014) ja Abrahamsson et al. (2017) väittävät tuoteomistajan roolin olevan kulmakivi ketterän tuotekehityksen menestyksekkäälle soveltamiselle, erityisesti Scrumissa, sillä roolin vastuulla on työn tehokkuuden ja tuottavuuden maksimoiminen. Tuoteomistajalle ei ole kuitenkaan muodostunut yhtenäistä kehystä akateemisessa kirjallisuudessa ja tulokset vaihtelevat siitä huolimatta, että varsinaisen Scrum-viitekehityksen kehittäjät Jeff Sutherland ja Ken Schwaber ovat määritelleet tuoteomistajan roolia heidän luomassa ja ylläpitämässä Scrum-oppaassa (engl. The Scrum Guide), jonka tarkoituksena on ollut asettaa ohjenuorat Scrumia harjoittaville taholle. Empiirisissä tutkimuksissa ollaan havaittu, että Scrumia sovelletaan todellisuudessa eri tavoin eivätkä organisaatiot noudata täysin Scrum-oppaan sääntöjä. Toisaalta Scrum-oppaassa onkin mainittu ettei kaikkia sen sääntöjä tarvitse noudattaa sanasta sanaan. Näin ollen tuoteomistajan rooli vaihtelee yrityksestä toiseen eikä yrityksillä ole yhtenäistä näkemystä roolin jalkauttamisessa organisaatioon.

Näiden sanojen saattelemana, tässä kandidutkielmassa tarkastelen Scrum-opasta ja eri empiirisiä tutkimuksia aiheesta. Päättökysymys on muotoa:

- Mikä on tuoteomistajan rooli Scrumissa?

Tähän tutkimuskysymyksen vastataan seuraavien alakysymyksien avulla:

- Mikä on tuoteomistajan määritelmä Scrumin kontekstissa?
- Mitkä ovat tuoteomistajan tehtävät Scrumin kontekstissa?

2 Scrum ketterässä ohjelmistokehityksessä

Tässä luvussa esitellään ketterä ohjelmistokehitys ja Scrum saatavilla olevan tieteellisen kirjallisuuden pohjalta. Ensimmäisessä alaluvussa kartoitetaan ketterän ohjelmistokehityksen lähtökohdat ja periaatteet, jonka pohjalta toisessa alaluvussa tarkastellaan seikka-peräisesti Scrumin taustaa, määritelmää ja periaatteita.

2.1 Ketterä ohjelmistokehitys

Ketterä ohjelmistokehitys (engl. agile software development) kehitettiin vastaamaan perinteisen ohjelmistokehityksen (engl. traditional software development) tavoille tuottaa ohjelmistoa (Dybå ja Dingsøyr, 2008). Perinteisille menetelmille on ollut tyypillistä raskas dokumentointi ja prosessimaisuus, mikä on koettu ongelmalliseksi ohjelmistokehityksessä. Esimerkiksi yhdessä tunnetussa perinteisessä ohjelmistokehitysmenetelmässä, vesiputousmallissa (engl. the waterfall model), ohjelmistokehityksen vaiheet oletetaan tapahtuvan strukturoidusti peräkkäin niin, että kunkin vaiheen tuotokset ovat erillisten tuotantotiimien tekemiä. Lisäksi malli on suunnitelmavetoinen eli ohjelmistoprosessi on tarkkaan suunniteltu, resursoitu ja aikataulutettu. Nämä piirteet on nähty ongelmallisina, sillä muutoksiin, kuten asiakkaiden uusiin vaatimuksiin tai ohjelmistossa todettuihin ongelmiin, ei olla pystytty vastaamaan onnistuneesti. (Stoica et al., 2013)

Perinteisiin menetelmiin liittyvien ongelmien pohjalta Beck et al. (2001) kehittivät vuonna 2001 ketterän ohjelmistokehityksen julistuksen (engl. agile manifesto). Julistuksen tarkoituksena on ollut luoda yhteistä pohjaa ketterille menetelmille, jonka myötä yritykset voivat lisätä asiakastyytyväisyyttä, lyhentää kehitystyön elinkaarta, vähentää virheiden määrää ja mukautua muuttuviin vaatimuksiin kehitysprosessin aikana (Cho, 2008). Julistuksen mukaisesti ketterissä menetelmissä tulisi arvostaa erityisesti yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja, toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota, asiakasyhteistyötä enemmän kuin sopimusneuvotteluja ja vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa (Beck et al., 2001). Abrahamsson et al. (2017) viittaavat akateemisessa paperissaan ”*Agile Software Development Methods:*

Review and Analysis” ketterän ohjelmistokehityksen julistukseen ja selventävät julistuksen periaatteita:

Yksilöiden ja kanssakäymisen arvostaminen ohjelmistokehityksessä viittaa ohjelmistokehittäjien suhteisiin ja yhteisöllisyyteen. Kun ohjelmistokehitysprojehtissa työskenteleville annetaan vapaus toimia omien näkemysten mukaisesti ja kehitystyö tapahtuu suotuisassa ympäristössä, tiimihengen ja luottamuksen uskotaan maksimoituvan. Voidaan nähdä, että asiakkaalle pystytään tällöin tuottamaan arvoa, koska työyhteisössä ei takerruta liikaa kankeiden menetelmien suorittamiseen ja työkalujen passiiviseen käyttöön. (Abrahamsson et al., 2017)

Toimivaa ohjelmistoa toimittamalla, vähintään parin viikon tai kuukauden välein ja tarpeeksi aikaisessa vaiheessa, voidaan tuottaa arvoa asiakkaalle. Tällöin kehittäjien tulee pitää koodi yksinkertaisena, suoraviivaisena ja teknisesti mahdollisimman hyvälaatuisena. Nämä kaikki ominaisuudet vähentävät taas dokumentaation tarvetta. Kun kehittäjien ei tarvitse jatkuvasti käydä läpi toisten kirjoittamaa dokumentaatiota ymmärtääkseen ohjelmistoa, on sitä epäilemättä helpompi kehittää eteenpäin. (Abrahamsson et al., 2017)

Asiakasyhteistyötä ylläpitämällä voidaan tuottaa arvoa jo kehitystyön aikana. Kun asiakkaan kanssa kommunikoidaan ja neuvotellaan läpi kehitystyön, voidaan ohjelmistosta saada juuri asiakkaan tarpeisiin sopiva tuote. Samoin asiakkaalle tulee selkeämpi näkemys siitä, mitä on mahdollista toteuttaa ja mitä ei, kun kommunikointi ja neuvottelu jatkuu läpi projektin. On kuitenkin tärkeää muistaa ettei sopimuksia saisi täysin laiminlyödä projektin aikana eli näistä tulisi pitää kiinni, mutta joitakin kohtia voi jättää aina neuvottelun varaan. (Abrahamsson et al., 2017)

Muutoksiin vastaaminen on yksi ketterien menetelmien tärkeimmistä lähtökohdista. Kaikkien projektin osapuolien tulisi olla perillä tärkeistä projektiin liittyvistä seikoista sekä olla valmiita ottamaan vastaan mahdolliset muutokset läpi ohjelmistokehitysprosessin, myös kehityksen myöhäisessä vaiheessa. Lisäksi heillä tulee olla aina valmius reagoida näihin muutoksiin. Kehitystiimin on myös tärkeää tarkastella säännöllisesti oman tehokkuuden parantamista ja työskentelyn mukauttamista toimintaansa. (Abrahamsson et al., 2017)

Kokonaisuudessaan voidaan nähdä, että ketteriä menetelmiä käyttävän tahon tulisi soveltaa ketteryyttä kokonaisvaltaisesti sekä ajattelu-, työ- ja johtamistavoissa. Toimimalla näin, ketteryys luo ennen kaikkea mahdollisuuden sopeutua nopeammin muuttuvaan maailmaan ja näin myös sujuvamman ohjelmistokehitysprojehtin hallinnan. (Abrahamsson et al., 2017)

2.2 Scrum

Tässä luvussa käsitellään tarkemmin Scrumin taustaa, määritelmää ja elinkaarta. Päämääränä on havainnoida syitä Scrumin käyttöönotolle ja muodostaa kokonaisvaltainen kuva Scrumista.

2.2.1 Scrumin taustaa

Scrum on kehitetty 1990-luvun alussa Jeff Sutherlandin ja Ken Schwaberin toimesta (Schwaber ja Sutherland, 2020). Scrumin kehittymiseen vaikutti voimakkaasti Takeuchi ja Nonaka (1986) julkaisema artikkeli ”*The New New Product Development Game*”. Artikkeleissa menestyneiden tuotekehitysryhmien käytäntöjä verrataan rugby pelissä tapahtuvaan ilmiöön, jossa tiimi etenee kentän poikki itseorganisoituvasti ja tiiviissä yhteistyössä niin, että jokaisella tiimin jäsenellä on hyvä yleiskuva pelin tämänhetkisestä tilasta. Schwaber (1995) mainitsee, että Scrum onkin saanut nimensä tämän artikkelin pohjalta termistä *scrumdown*.

Termiä *scrum* käytettiin ensimmäisen kerran kontekstissaan Peter DeGracen ja Leslie Stahlin vuonna 1990 kirjoittamassa kirjassa ”*Wicked Problems, Righteous Solutions*” (Schwaber, 2004). Kirjassa tarkastellaan perinteisiin menetelmiin kuuluvan vesiputousmallin ongelmakohtia ja ratkaisuna näihin ongelmiin tarjotaan niin sanottua *Kaikki kerralla* -mallia (engl. the all-at-once model). Malli tarjoaa ajatuksen, jonka mukaan ohjelmistokehityksen vaiheet eivät enään tapahtuisi peräkkäin, vaan samanaikaisesti. (DeGrace ja Stahl, 1990) Näiden pohdintojen saattelemana Easel Corporation kehitti yhdessä Sutherlandin kanssa ensimmäisen ohjelmiston Scrum-teoriaa hyödyntäen vuonna 1993. Yrityksen toimitusjohtaja huomasi positiivisen muutoksen ohjelmiston kehittämisessä jo ensimmäisen sprintin jälkeen, kun edistyminen tapahtui odotettua nopeammin. Myös itse lopullinen tuote saatiin valmiiksi aikataulun mukaisena päivänä, mikä vakuutti toimitusjohtajan Scrumin sopivuudesta ohjelmistokehitysmenetelmänä. (Sutherland ja Schwaber, 2011)

2.2.2 Scrumin määritelmä

Scrumin määritelmä on muovautunut aikojen saatossa sen kehittäjien puolesta. Alunperin Schwaber (1995) kuvaili Scrumia ohjelmistokehityksen hallinta-, parannus- ja ylläpitomenetelmäksi. Tällöin Scrum nähtiin prosessina. Myöhemmin Sutherland ja Schwaber ovat

Scrum-oppaassa muotoilleet Scrumin määritelmän uudelleen. Kirjallisuuskatsauksen pohjalta kyseistä teosta pidetään Scrumin määritelmän esikuvana ja näin ollen myös tässä kandidutkielmassa Scrumin määritelmä muodostetaan kyseisen teoksen pohjalta. Teoksessa Scrumia kuvaillaan arvoa tuottavaksi menetelmäkehyyksi, jossa tehdään mukautuvia ratkaisuja ohjelmistokehityksen ongelmien piirissä. Tarkoituksena on antaa sellaisia sääntöjä ihmisille, organisaatioille tai tiimeille, jotka ohjaavat heidän suhteitaan ja vuorovaikutusta pikemminkin kuin yksityiskohtaisia ja ehdottomia sääntöjä. (Schwaber ja Sutherland, 2020)

Itse Scrum-tiimi muodostuu maksimissaan kymmenestä henkilöstä. Sen sisällä ei saa olla hierarkiaa ja sen tulisi olla yhtenäinen yksikkö, joka keskittyy yhteiseen tavoitteeseen kerrallaan. Tiimi koostuu Scrum-oppaan mukaan *kehittäjistä* (engl. developers), *scrum-mestarista* (engl. scrum master) ja *tuoteomistajasta* (engl. product owner). Kehitystiimi vastaa tuotteen toiminnallisuuden teknillisestä toteutuksesta, kun taas scrum-mestari varmistaa, että tiimi noudattaa Scrumin periaatteita, arvoja ja sääntöjä mallikkaasti. Tuoteomistaja edustaa taas ohjelmistohankkeen asiakasta ja muita sidosryhmiä Scrum-tiimille varmistaen, että heidän vaatimukset on selkeästi kommunikoitu Scrum-tiimille. (Schwaber ja Sutherland, 2020)

Luonteeltaan Scrum on *iteratiivinen* (engl. iterative) eli toistuva ja *inkrementaalinen* (engl. incremental) eli vähittäin kasvava. Ohjelmistokehitys tapahtuu näin ollen *sprinteistä* (engl. sprint), jotka kestävät yleensä kahdesta viikosta neljään viikkoon. Jokaisessa sprintissä on olennaista noudattaa Scrumille ominaisia *läpinäkyvyyden* (engl. transparency), *tarkastelun* (engl. inspection) ja *mukauttamisen* (engl. adaption) periaatteita. Läpinäkyvyyden periaatteen mukaan työntekijöillä tulisi olla hyvä yleiskuva kehitysprojektista. Läpinäkyvyys luo taas pohjan tarkastelulle. Tiimin tulisi tarkastella säännöllisesti, kuinka hyvin tuotteen kehitys edistyy, jotta toimintaa voidaan mukauttaa tarpeiden mukaisesti. Toisin sanoen tarkastelu mahdollistaa mukauttamisen, jonka avulla minimoidaan ei-toivotut ongelmat. Muutoksia tulisi tehdä niin pian kuin mahdollista. (Schwaber ja Sutherland, 2020)

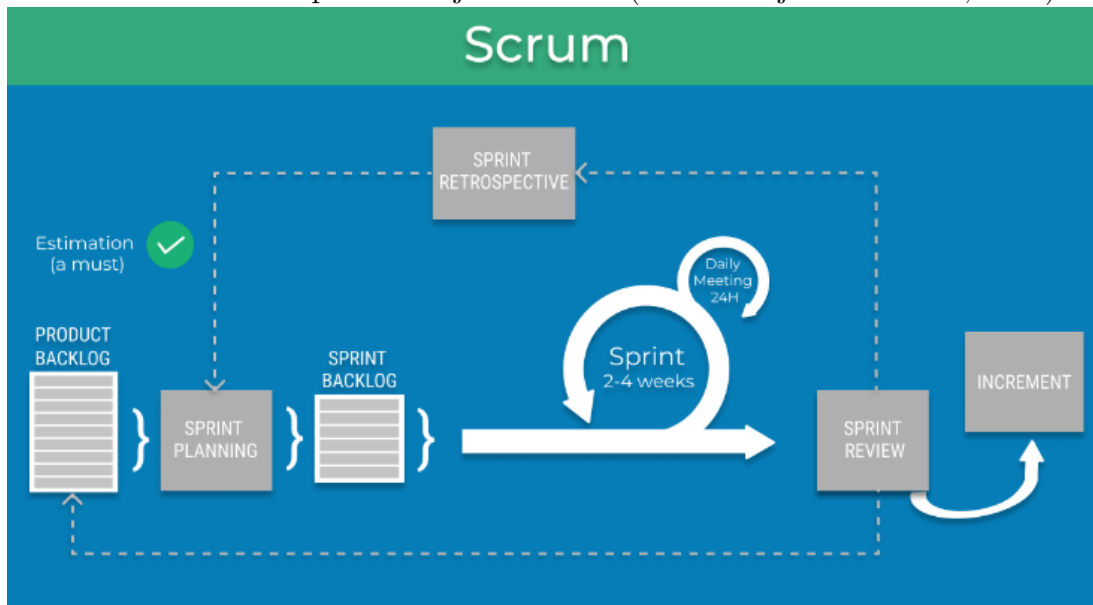
Periaatteiden lisäksi, Scrum-menetelmäkehys koostuu myös joukosta arvoja, jotka luovat pohjan tehokkaalle työlle, toiminnalle ja käyttäytymiselle. Näitä arvoja ovat Scrum-oppaan mukaan *sitoutuminen* (engl. commitment), *keskittyminen* (engl. focus), *avoimuus* (engl. openness), *kunnioitus* (engl. respect) ja *rohkeus* (engl. courage). Sitoutumien viittaa yhteisen tavoitteen saavuttamiseen sekä oikeiden asioiden tekemiseen. Tiimin tulisi keskittyä työhönsä, jotta tavoitteita voidaan edistää. Työn ja sen haasteiden tulisi olla avoimesti selvillä sekä Scrum-tiimille että sidosryhmille. Kaikkien osapuolien tulisi myös kunnioit-

taa toisiaan eli pitää toisia kyvykkäinä ja itsenäisinä ihmisinä. Lisäksi tiimillä tulee myös olla rohkeutta päätöksen teossa. (Schwaber ja Sutherland, 2020)

2.2.3 Scrumin tapahtumat ja tuotokset

Kuva 1 näyttää Scrumin tapahtumat joita ovat *sprintin suunnittelu* (engl. sprint planning), *päivittäispalaveri* (engl. daily scrum), *sprintin katselmointi* (engl. sprint review) ja *sprintin retrospektiivi* (engl. sprint retrospective). Lisäksi Kuvassa 1 voidaan nähdä Scrumin tuotokset eli *tuotteen kehitysjono* (engl. product backlog), *sprintin kehitysjono* (engl. sprint backlog) ja inkrementti (engl. increment). Nämä kaikki yhdessä mahdollistavat Scrumin periaatteet, ja joita noudattamalla voidaan toteuttaa Scrumille ominaisia arvoja. (Schwaber ja Sutherland, 2020)

Kuva 1: Scrumin tapahtumat ja tuotokset (Schwaber ja Sutherland, 2020).



Ennen ensimmäisen sprintin suunnittelupalaveria luodaan alustava tuotteen kehitysjono, mikä on priorisoitu lista asiakkaan vaatimuksista. Itse sprintti aloitetaan sprintin suunnittelulla, mikä on Scrum-tiimin ja asiakkaan välinen tapaaminen. Tässä tapaamisessa määritellään *sprintin tavoite* (engl. sprint goal) ja kerrotaan, miksi juuri kyseinen sprintti on arvokas asiakkaalle. Toisin sanoen Scrum-tiimi määrittelee miten sprintissä voidaan lisätä arvoa. Tämän lisäksi tuotteen kehitysjonosta valitaan sprintissä suoritettavat työtehtävät sprintin omaan kehitysjonoon. Sprintin kehitysjono on kehittäjien käyttämä lista sprintin aikana tehtävistä työtehtävistä, jonka avulla saadaan selkeä käsitys mitä tulevan

sprintin aikana tehdään. Lopuksi tapaamisessa on tarkoitus selvittää miten sprintin tavoitteet saavutetaan. Tämä tapahtuu selventämällä hyväksymiskriteerit sille, milloin tehtävät täyttävät *valmiin määritelmän* (engl. definition of done) eli sen milloin tehtävistä muodostuu uusia inkrementtejä tuotteelle. (Schwaber ja Sutherland, 2020) Tässä tutkielmassa valmiin määritelmää kutsutaan jatkossa englanninkielisellä nimityksellä.

Sprintin edetessä on olennaista pitää kehitysryhmän kesken päivittäispalavereita. Scrum-oppaan mukaan päivittäispalaverit parantavat kommunikaatiota, edistävät päätöksentekoa ja auttavat tunnistamaan mahdollisia esteitä tehtävien tekemiselle. Ideaalisti, päivittäispalaveri tapahtuu jokaisena työpäivänä ja sen kesto on 15 minuuttia. Tapaamisen aikana jokainen kehitysryhmän jäsen kertoo työn alla olevista tehtävistään, joiden kautta tarkastellaan sprintin tavoitteita ja mukautetaan sprintin kehitysjonoa ja tulevia suunnitelmia tarpeen mukaan niin, että tulevalle työpäivälle on olemassa selkeä suunnitelma. (Schwaber ja Sutherland, 2020)

Jokaisen sprintin lopussa pidetään kehitysryhmän ja asiakkaan välinen sprintin katselmointi. Sprintin katselmoinnissa asiakkaalle esitetään Definition of Done -määritelmän saavuttaneet toiminnot, mutta myös ne toiminnot, jotka eivät ole tätä määritelmää täyttäneet. Mikäli tehtävä täyttää Definition of Done -määritelmän vaatimukset, syntyy tästä inkrementti ohjelmistoon ja se poistetaan tuotteen kehitysjonosta. Muussa tapauksessa tehtävä jätetään tuotteen kehitysjonoon. Kokonaisuudessaan katselmoinnissa tarkastellaan tuotteen kehitysjonon tilaa ja mukautetaan sitä tarvittaessa. Tarkoituksena on myös selvittää tarvittavat muutokset seuraaviin sprintteihin ja keskustella mitä tehdään seuraavaksi. (Schwaber ja Sutherland, 2020)

Sprintti päätetään retrospektiivillä, jossa analysoidaan kehitysryhmän onnistumista sprintissä. Tarkoituksena on parantaa tuotteen laatua sekä työn tuottavuutta ja tehokkuutta. Retrospektiivissä keskustellaan siitä, mikä meni hyvin, mitä ongelmia kohdattiin ja kuinka nämä ongelmat ratkaistiin. Mikäli ongelmia ei ratkaistu, tulisi pohtia sitä miksi ne jätettiin ratkaisematta. Näiden pohjalta Scrum-tiimi suunnittelee muutoksia toimintaansa parantaakseen tehokkuutta ja tuottavuutta sekä tuotteen laatua. (Schwaber ja Sutherland, 2020)

3 Tuoteomistaja Scrumissa

Tämän luvun tarkoituksena on vastata päättökysymykseen sille määriteltyjen alaky-symyksiä avulla. Ensimmäisessä luvussa muodostetaan tuoteomistajan määritelmä, jonka jälkeen ryhdytään täydentämään tätä tarkastelemalla tarkemmin hyvän tuoteomistajan ominaisuuksia. Tämän jälkeen tutkielma etenee tuoteomistajan tehtävien kuvaamiseen Scrumin kontekstissa.

3.1 Tuoteomistajan määritelmä

Taulukko 1 näyttää kirjallisuuskatsauksen pohjalta havaitut määritelmät tuoteomistajalle. Tuoteomistaja nähdään yleisesti yhtenä henkilönä, ei komiteana (Sverrisdottir et al., 2014; Schwaber ja Sutherland, 2020), mutta akateemisesta kirjallisuudesta löytyy myös viitteitä siitä, että tuoteomistajia voi olla myös useampia. Esimerkiksi Paasivaara et al. (2012) ovat todenneet ettei tuoteomistaja pysty suorittamaan tehtäviä yksin, mikäli Scrumia skaalataan suurempiin projekteihin joissa ryhmiä voi olla useampia. Tässä tutkielmassa Scrum-projekti rajataan kuitenkin pienempään projektiin niin, että yksi tuoteomistaja kykenee suorittamaan kaikki rooliin liittyvät tehtävät.

Scrum-oppaan määritelmän mukaan tuoteomistaja edustaa asiakkaita ja muita sidosryhmiä Scrum-tiimille niin, että Scrum-tiimi tuottaa suurimman mahdollisen arvon tuotteesta (Schwaber ja Sutherland, 2020). Unger-Windeler ja Schneider (2019) ja Ghiba (2022) määrittelevät tuoteomistajan henkilöksi, joka edustaa asiakkaan sekä sidosryhmien tarpeita ja tavoitteita tekemällä tuotekehitykseen liittyviä päätöksiä. Tuoteomistaja toteuttaa tämän kommunikoimalla asiakkaan, sidosryhmien ja Scrum-tiimin kanssa (Matturro et al., 2018) sekä suorittamalla erilaisia vaativia tehtäviä (Bass et al., 2018). Näiden myötä hän voi edistää ymmärrystä liiketoiminnallisten ja teknisten näkökohtien välillä sekä tehdä ratkaisuja siitä, mitä ominaisuuksia ja toimintoja ohjelmistoon toteutetaan ja missä järjestyksessä ne rakennetaan (Sverrisdottir et al., 2014; Matturro et al., 2018).

Näiden päätelmien pohjalta, tässä kandidaatintutkielmassa tuoteomistaja määritellään yhdeksi henkilöksi, joka edustaa asiakasta ja muita sidosryhmiä Scrum-tiimille suorittamalla sellaisia tehtäviä, jotka varmistavat maksimaalisen arvon tuottamisen asiakkaalle.

Taulukko 1: Tuoteomistajan määritelmät kirjallisuudessa.

Artikkeli	Määritelmä
Sverrisdottir et al. (2014)	Tuoteomistaja omistaa tuotteen ja tekee päätöksiä siitä, mitä ohjelmistokehitysprojektissa tehdään perustuen organisaation asiakkaan ja sidosryhmien tarpeisiin.
Matturro et al. (2018)	Tuoteomistaja edustaa liiketoiminnallisia tavoitteita ja osallistuu Scrum-tiimin päivittäiseen toimintaan tehdäkseen päätöksiä siitä, mitä projektissa tehdään ja missä järjestyksessä.
Bass et al. (2018)	Tuoteomistaja on henkilö, joka kääntää liiketoiminnan tarpeet arvoa tuottavaksi ohjelmistoksi suorittamalla erilaisia haastavia tehtäviä.
Unger-Windeler ja Schneider (2019)	Tuoteomistaja on henkilö, joka edustaa asiakkaan tarpeita.
Schwaber ja Sutherland (2020)	Tuoteomistaja on yksi henkilö, jonka vastuulla on varmistaa, että Scrum-tiimin tekemä työ tuottaa suurimman mahdollisen arvon asiakkaalle ja muille sidosryhmille.
Ghiba (2022)	Tuoteomistaja edustaa asiakasta ja määrittelee kaikki tuotekehitykseen liittyvät päätökset.

3.2 Tuoteomistajan ominaisuudet

Kirjallisuudesta ilmenee, että tuoteomistajan tulee olla tiettyjen ominaisuuksien omaava henkilö, jotta hän sopii rooliin ja, jotta tuotteen tavoitteet voidaan saavuttaa parhaalla mahdollisella tavalla. McGreal ja Jocham (2018) esittelevät teorian, jonka mukaan tuoteomistajan tulee olla ominaisuuksiltaan CRACK eli *yhteistyökykyinen* (engl. **C**olloborative), *edustava* (engl. **R**epresentative), *valtuutettu* (engl. **A**uthorized), *sitoutunut* (engl. **C**ommitted) ja *asiantunteva* (engl. **K**nowledgeable).

Yhteistyökykyisyys viittaa tuoteomistajan kykyyn kommunikoida asianmukaisesti eri osapuolien kanssa. Tuoteomistajan tulee olla ektrovertti ja helposti lähestyttävä henkilö onnistuakseen tässä. (Sverrisdottir et al., 2014; McGreal ja Jocham, 2018) Yhteistyökykyisyys on tuoteomistajan arvostetuimpia ominaisuuksia (Matturro et al., 2018) ja tämän

voidaan nähdä johtuvan Scrumin luonteesta, mikä pohjautuu vahvasti yhteistyöhön ja vuorovaikutukseen.

Edustavuus ilmaisee tuoteomistajan kykyä luoda empatiaa sidosryhmien ja asiakkaiden kanssa. Tuoteomistajalla tulee olla kyky ymmärtää sidosryhmiä ja asiakkaita sekä tuoda heidän tarpeitaan esille, kun he eivät ole läsnä. (McGreal ja Jocham, 2018) Tämän voidaan nähdä vaativan hyviä kommunikaatiotaitoja. Sverrisdottir et al. (2014) mukaan tuoteomistajalla tulisikin olla sekä hyvät suulliset että kirjalliset kommunikaatiotaidot, sillä roolissa käydään jatkuvaa keskustelua eri osapuolien kanssa.

Valtuutettavuus tarkoittaa tuoteomistajan pätevyyttä tehdä tuotetta koskevia päätöksiä (McGreal ja Jocham, 2018). Kuten mainittu, Scrum-projekteissa on valtuutus tehdä muutoksia myös hyvinkin myöhäisessä vaiheessa kehitystä. Jotta muutoksia voidaan tehdä, tarvitaan hyviä suunnittelu- ja analysointitaitoja. Tuoteomistajalla tulee olla kykyä analysoida tilanteita, tehdä suunnitelmia ja ratkaisuja näiden perusteella, jotta muutoksiin kyetään valmistautumaan ja niihin pystytään vastaamaan onnistuneesti. (Sverrisdottir et al., 2014; Matturro et al., 2018)

Sitoutuneisuus on Matturro et al. (2018) tutkimuksen valossa tärkeä tuoteomistajan ominaisuus sekä scrum-mestarin että kehittäjien näkökulmasta. Sitoutunut tuoteomistaja seuraa Scrumin periaatteita hyväksymällä vain käyttökelpoisia ja arvoa tuottavia ominaisuuksia tuotteesta asiakkaalle. Samalla hän seuraa tuotteen visiota sekä varmistaa tuotteen olevan asiakkaan tarpeiden mukainen. (McGreal ja Jocham, 2018)

Asiantunteva tuoteomistaja tuntee oman toimialueensa ja pysyy ajan tasalla viimeisimmistä teknologiatrendeistä, markkinoista ja kilpailutilanteesta (McGreal ja Jocham, 2018). Tuoteomistajan tulee olla asiakasorientoitunut (Matturro et al., 2018) ja hänen tulee hallita ohjelmistokehityksen tekninen puoli, jotta hän pystyy ymmärtämään kehittäjien näkemyksiä ja tekemään päätöksiä projektin etenemisen suhteen sekä varmistamaan, että asiakas saa toimivan ohjelmiston mahdollisimman aikaisessa vaiheessa (Sverrisdottir et al., 2014).

Unger-Windeler ja Schneider (2019) tuovat tutkimuksessaan esille tärkeitä ominaisuuksia, jotka tuoteomistajalla tulisi olla. Nämä ominaisuudet muistuttavat vahvasti CRACK-teorian ominaisuuksia. Tutkijoiden mukaan kyseiset ominaisuudet tekevät tuoteomistajasta kokonaisuudessaan luotettavan. Voidaankin todeta, että tuoteomistajan tärkein ominaisuus on olla luotettava. Ollakseen luotettava, tulee tuoteomistajan ominaisuuksien rakentua CRACK -teorian mukaisista ominaisuuksista.

3.3 Tuoteomistajan tehtävät

Scrum-oppaan mukaan tuoteomistajan kaikkien tehtävien tarkoituksena on tuottaa arvoa asiakkaalle. Se miten tämä käytännössä toteutetaan voi vaihdella organisaatioiden, Scrum-tiimien ja yksilöiden välillä. (Schwaber ja Sutherland, 2020) Tämä voikin olla syynä sille, miksi empiiristen tutkimusten tulokset tuoteomistajan tehtävistä vaihtelevat, eikä roolille ole näin ollen kehittynyt vakiintunutta tehtävänkuvaa, kuten Taulukosta 2 voidaan todeta. Toisaalta myös itse tutkimukset aiheesta ollaan toteutettu eri tavoin. Esimerkiksi Unger-Windeler ja Schneider (2019) ja Ghiba (2022) ovat toteuttaneet tutkimukset kyselytutkimuksina, joissa suurin osa vastaajista on ollut rooliltaan muita kuin tuoteomistajia. Sverrisdottir et al. (2014), Oomen et al. (2017), Bass et al. (2018) ja Matturro et al. (2018) ovat taas toteuttaneet tutkimuksensa haastatteluina ja havainnointina suurimmaksi osaksi itse tuoteomistajille. Tämä voidaankin nähdä syynä sille, miksi Unger-Windeler ja Schneider (2019) ja Ghiba (2022) tuottamat tulokset eroavat muiden tutkimusten tuloksista huomattavan paljon.

Taulukko 2: Tuoteomistajan tehtävät kirjallisuudessa

<div> <div>Artikkeli</div> <div>Tehtävä</div> </div>	Sverrisdottir et al., 2014	Oomen et al., 2017	Bass et al., 2018	Matturro et al., 2018	Unger-Windeler ja Schneider, 2019	Schwaber ja Sutherland, 2020	Ghiba, 2022
Vaatimusten määrittely	✓	✓	✓	✓	x	✓	x
Tuotevision rakentaminen	✓	✓	x	✓	x	✓	✓
Julkaisun suunnittelu	x	✓	✓	✓	x	x	✓
Vaatimusten kirjoittaminen	✓	✓	✓	✓	✓	✓	x
Tuotteen kehitysjonon hallinta	✓	✓	✓	✓	✓	✓	✓
Osallistuminen ja yhteistyö Scrumin tapahtumissa	✓	✓	✓	✓	✓	✓	✓
Asiakassuhteiden hallinta	x	✓	✓	✓	x	x	x

Tutkimustulosten poikkeavuudesta huolimatta, kaikki Taulukon 2 sisältämät tutkimukset osoittavat yksimielisesti sen, että tuoteomistajan tehtäviä ovat tuotteen kehitysjonon hallinta sekä osallistuminen ja yhteistyö Scrumin tapahtumissa. Suurin osa tutkimuksista antaa osviittaa myös sille, että tuoteomistajan vakiintuneita tehtäviä ovat vaatimusten määrittely ja kirjoittaminen sekä tuotevision rakentaminen. Osa tutkimuksista osoittaa tuoteomistajan olevan vastuussa myös tuotteen lyhyen ja pitkän aikavälin suunnittelusta, jota voidaan kutsua julkaisun suunnitteluksi (engl. release planning) (Kantola et al., 2022). Lisäksi muutamat tutkimukset ottavat esille tuoteomistajan tärkeän roolin asiakassuhteiden hallinnassa. Asiakassuhteiden hallinnan voidaan kuitenkin nähdä tapahtuvan muiden tehtävien lomassa, joten tässäkin kandidaatintutkielmassa tämä aihealue käsitellään muiden tehtävien yhteydessä.

3.3.1 Vaatimusten määrittely ja tuotevision rakentaminen

Yksi tuoteomistajan tärkeimpiä tehtäviä on määritellä asiakkaan vaatimukset eli tarpeet ja tavoitteet tuotteelle (Sverrisdottir et al., 2014; Oomen et al., 2017; Bass et al., 2018; Maturro et al., 2018). Scrum-opas ottaa kantaa vaatimusten määrittelyyn, mutta ei tarkenna sitä miten se toteutetaan käytännössä. Tieteelliset julkaisut antavat kuitenkin osviittaa sille, että vaatimusten määrittely aloitetaan tuotevision rakentamisen pohjalta. Esimerkiksi Pichler (2010) ja McGreal ja Jocham (2018) mukaan tuotevisio luo Scrumille olennaista läpinäkyvyyttä ja määrittelee pitkän aikavälin suunnan eli sen mihin ohjelmistokehityksessä tulisi panostaa ja mitä tulisi jättää huomioimatta. Tämä auttaa tuoteomistajaa määrittelemään niitä vaatimuksia tuotteelle, jotka ohjelmistokehittäjien tulee toteuttaa.

Tuoteomistaja on päävastuussa tuotevision rakentamisesta (Ghiba, 2022) ja se on tehtävä perusteellisesti, sillä se vaikuttaa vahvasti ohjelmistokehityksen tehokkuuteen (Tessarolo, 2007). Scrum-projektissa tuoteomistajan ei ole kuitenkaan suotuisaa laatia tuotevisiota täysin yksin, vaan se tulee muodostaa vuorovaikutuksessa muiden projektin osapuolien kanssa. Tässä tuoteomistaja voi hyödyntää erilaisia metodeja, kuten visionrakentamistyöpajoja, haastatteluja, kyselyitä ja havainnointia. (Cohn, 2004; Blasquez ja Leblanc, 2018; Trießlinger et al., 2021) Empiiriset tutkimukset ovat osoittaneet, että etenkin visionrakentamistyöpajoilla on positiivinen vaikutus Scrum-projekteissa. Esimerkiksi Blasquez ja Leblanc (2018) ja Trießlinger et al. (2021) ovat havainneet visionrakentamistyöpajojen tuovan projektin osapuolet yhteen ja luovan yhteisymmärryksen tuotteen tulevasta suunnasta sekä määrittelevän tuotteen tavoitteen ja kehitettävät ominaisuudet.

Työpajoissa, haastatteluissa ja kyselyissä tuoteomistaja voi hyödyntää erilaisia työkaluja, joiden avulla voidaan selventää mitä tuotetta tehdään, kenelle tuote tehdään, miksi tuote tehdään ja miten tuote tehdään (Benassi et al., 2016; Blasquez ja Leblanc, 2018; Triefflinger et al., 2021). Kirjallisuudesta löytyy useampia esimerkkejä erilaisista työkaluista, mutta erityisesti *hissipuheiden* (engl. elevator pitch tai elevator test statement) laatiminen tuotteesta näyttää olevan ohjelmistokehitykseen keskittyvien organisaatioiden suosiossa. Esimerkiksi Triefflinger et al. (2021) toteuttamissa työpajoissa, hissipuheilla oli selkeä positiivinen vaikutus yritysten toimintaan. Hissipuheiden avulla tuotevisiota päästiin jalostamaan mahdollisimman tarkaksi ja kiinnostavaksi kuvaukseksi. (Blasquez ja Leblanc, 2018; Triefflinger et al., 2021)

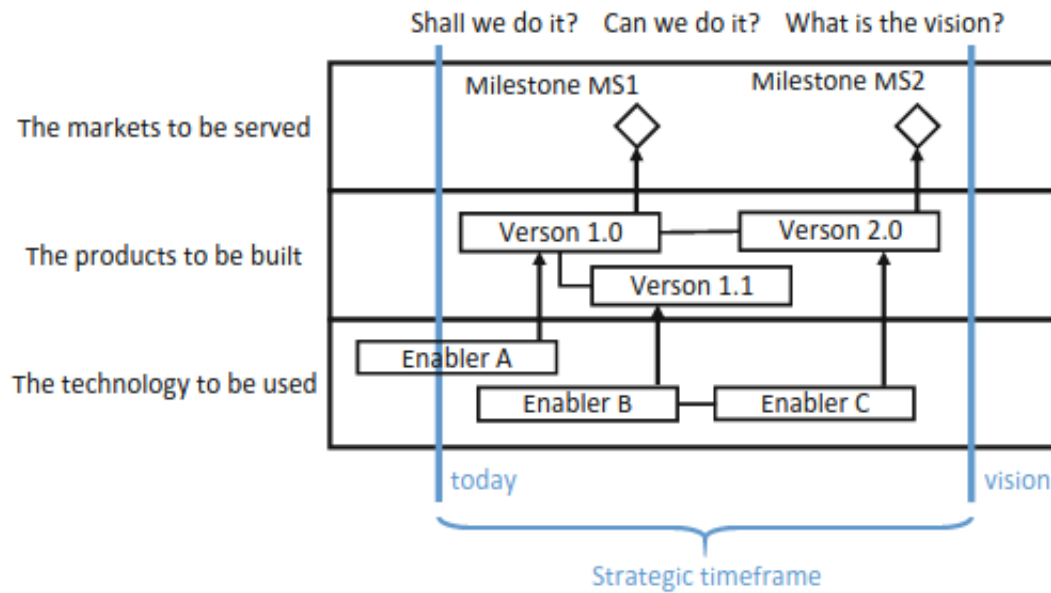
Scrum-projektissa ei yleisesti riitä, että tuotevisio ja tuotteen vaatimukset määritellään projektin alussa, vaan niitä tulisi tämentää läpi projektin elinkaaren. Näin ollen, tuoteomistajan tulee tehdä asiakasyhteistyötä eli kommunikoida sekä viettää aikaa asiakkaan ja sidosryhmien kanssa lisätäkseen ymmärrystä tuotteelle asetetuista tarpeista ja tavoitteista (Bass et al., 2018). Toisaalta tuoteomistajan tulee myös kommunikoida kehitystiimin kanssa ymmärtääkseen, miten tavoitteet ja tarpeet voidaan saavuttaa tai voidaanko niitä ylipäätensä edes toteuttaa (Vähäniitty et al., 2002; Bass et al., 2018).

3.3.2 Julkaisun suunnittelu

Kun tuotteen visio on selvillä ja tuoteomistaja on kerryttänyt tarpeeksi ymmärrystä tuotteen vaatimuksesta sekä määritellyt näitä, voi hän tehdä suunnitelman tuotteen kehityksestä sekä lyhyelle että pitkälle aikavälille eli julkaisun suunnittelun (Oomen et al., 2017; Bass et al., 2018; Matturro et al., 2018; Ghiba, 2022; Kantola et al., 2022; Layton, 2022). Akateemiset paperit suosittelevat tuoteomistajaa hyödyntämään tässä kohtaa *roadmap*-tekniikkaa (engl. roadmapping). Roadmap on visuaalinen suunnitelma siitä, mitä Scrum-tiimi toimittaa ohjelmistoon ajan mittaan, jotta tuotteen visio saavutetaan. Roadmappia tulee päivittää projektin kuluessa, sillä roadmap on nimenomaan vain suuntaa antava suunnitelma. Sen tarkoituksena on auttaa tuoteomistajaa laatimaan tarkempia vaatimuksia tuotteelle sekä tarjoamaan sidosryhmille, asiakkaille ja kehitystiimille mahdollisuuden tarkastella tietoa kehitystyöstä ja sen etenemisestä yhdellä silmäyksellä. Se auttaa etenkin varmistamaan, että tuotevisio on linjassa liiketoiminnallisten tavoitteiden kanssa ja, että kaikilla projektin osapuolilla on yhteisymmärrys kehitettävistä ominaisuuksista. Lisäksi se auttaa ylläpitämään tietoisuutta projektin laajemmasta kuvasta ja pitämään kiinni lyhyen- ja pitkän aikavälin tavoitteista. (Kittlaus ja Fricker, 2017; Layton, 2022)

Roadmapin visualisointi itsessään voidaan toteuttaa usealla eri tavalla. Kuva 2 näyttää esimerkin roadmapista, mikä on toteutettu kerrostetussa muodossa. Kerrostettu roadmap muodostuu eri tasoista ja niiden elementeistä, jotka sijoitetaan roadmappiin aikajanaalle näyttämään, missä vaiheessa minkäkin elementin on tarkoitus toteutua. Ylin taso eli markkinataso määrittelee tuotemarkkinoilla saavutettavat virstanpylväät eli saavutukset, joilla päästään lähemmäksi tuotteen visiota. Seuraava taso eli tuotetaso määrittelee karkeasti tuotteen kehitykseen liittyvät toimenpiteet eli ne tärkeimmät piirteet joita tuotteesta pyritään toimittamaan ajan mittaan, jotta virstanpylväät toteutuvat. Viimeinen taso eli teknologiataso määrittelee tärkeitä teknologisia mahdollistajia, kuten tuotteen komponentteja ja kehyksiä, joita käytetään tuotteen kehittämiseen ja toimittamiseen, jotta tuotetason elementit pystytään toteuttamaan. (Kittlaus ja Fricker, 2017)

Kuva 2: Roadmap (Kittlaus ja Fricker, 2017).



3.3.3 Vaatimusten kirjoittaminen

Tuotevision rakentamisen, vaatimusten määrittelemisen ja julkaisun suunnittelun pohjalta, tuoteomistaja voi kirjoittaa vaatimuksia, jotka kuvailevat ohjelmiston yksittäisiä toimintoja (Pichler, 2010; Sverrisdottir et al., 2014; Oomen et al., 2017; McGreal ja Jocham, 2018; Matturro et al., 2018). Unger-Windeler ja Schneider (2019) ovat havainneet, että Scrum-tiimi olettaa tuoteomistajan kirjoittavan vaatimukset täysin itsenäisesti ja lisäävän nämä tuotteen kehitysjonoon. Toisaalta Mahnic (2011) ja Matturro et al. (2018) toteuttamien tutkimusten mukaan tuoteomistaja kirjoittaa kaikki vaatimukset Scrum-tiimin avus-

tamana. Näin ollen voidaan todeta, että tuoteomistaja kirjoittaa vaatimukset ylös kuunnellen Scrum-tiimiä ja ottaen huomioon heidän näkemykset ohjelmiston vaatimuksista.

Vaatimuksia on useita eri tyyppisiä. Ne voivat olla esimerkiksi ei-toiminnallisia vaatimuksia, kuten suorituskykyyn tai käyttökokemukseen liittyviä vaatimuksia. Toisaalta vaatimukset voivat liittyä ohjelmiston ominaisuuksiin, kuten sidosryhmien pyyntöihin saada eri käyttöoikeuksia ohjelmistoon. Suurimman osan vaatimuksista muodostaa kuitenkin käyttäjätarinat, jotka kertovat ohjelmiston käyttäjän näkökulmasta halutut toiminnallisuudet. Huolimatta vaatimuksen tyypistä, tuoteomistaja kirjoittaa niitä pienelle kortille, jonka etupuolella on lyhyt kuvaus vaatimuksen sisällöstä ja takapuolella joukko hyväksymiskriteereitä (McGreal ja Jocham, 2018). Nykypäivänä organisaatiot hyödyntävät entistä enemmän myös erilaisia ohjelmistoja, kuten Jiraa, Taigaa, Treolla ja PivotalTrackeria vaatimusten kirjoittamiseen ja hallintaan. Tämä on tehnyt vaatimusten kirjoittamisesta tehokkaampaa ja nopeampaa. (Sarkan et al., 2011)

Kirjoittaessaan vaatimuksia, tuoteomistajan tulee huomioida, että kaikki kirjoitetut vaatimukset sisältävät tarpeeksi tietoa, jotta kehittäjät voivat arvioida niiden implementointiin tarvittavan työn määrän (McGreal ja Jocham, 2018). Kirjallisuudesta ei löydy tarkkaa määritelmää sille miten vaatimukset tulee kirjoittaa lukuun ottamatta käyttäjätarinoita. Tämä voi johtua siitä, että käyttäjätarinoista on tullut yleisimmin käytetty vaatimustyyppi ketterissä menetelmissä (Kassab, 2015). Sedano et al. (2019) ovat tutkimuksessaan vahvistaneet Scrumia käyttävien organisaatioiden hyödyntävän seuraavaa muotoa kirjoittaessaan käyttäjätarinoita:

As a <role/persona> I want <behavior> so that <the value>

Muotoilua ei pysty täysin kääntämään suomenkielelle. Tarkoituksena on kuitenkin kirjoittaa käyttäjätarinat sellaiseen muotoon, että ne kuvailevat kenelle toiminto tehdään, mitä tehdään ja miksi ne tehdään (Sedano et al., 2019). Näin käyttäjätarinoiden laatu voidaan taata, mikä parantaa samalla Scrum-projektin tuottavuutta ja työn tehokkuutta. Lucasen et al. (2016) mukaan tämä toteutuu, kun tuoteomistaja laatii käyttäjätarinan, mikä on ominaisuuksiltaan INVEST eli *riippumaton* (engl. **I**ndependent), *neuvoteltava* (engl. **N**egotiable), *arvoa tuottava* (engl. **V**aluable), *estimoitava* (engl. **E**estimable) ja *testattava* (engl. **T**estable). Näitä ominaisuuksia noudattamalla, tuoteomistaja voi helpottaa käyttäjätarinoiden ymmärtämistä ja hyväksymiskriteerien kirjoittamista. Hyväksymiskriteerit määrittelevät sen milloin vaatimus on valmis eli sen milloin vaatimus täyttää Definition of Done -määritelmän. Tuoteomistaja kirjoittaa hyväksymiskriteerit yhdessä kehittäjien

kanssa kullekin käyttäjätarinalle. Diebold et al. (2015) ovat tutkimuksessaan havainneet, että useimmissa organisaatioissa hyväksymiskriteereitä kirjoitetaan eri tavoin ja ne ovat yleensä enemmän tai vähemmän tarkkoja. Tärkeintä on kuitenkin se, että Scrum-tiimi pystyy ymmärtämään tarkat toimenpiteet vaatimusten suorittamiseen.

3.3.4 Tuotteen kehitysjonon hallinta

Ensimmäisen sprintin alussa tuoteomistaja laatii alustavan tuotteen kehitysjonon kirjoitetuista vaatimuksista. Tuotteen kehitysjoono on nimenomaan aina alustava lista, sillä se elää koko projektin ajan ja sitä voidaan muuttaa tarvittaessa. (Sedano et al., 2019; Schwaber ja Sutherland, 2020) Tuoteomistajan tehtävänä onkin nimenomaan aktiivisesti hoitaa tuotteen kehitysjoonoa varmistamalla, että se on aina DEEP eli *sopivan detaljoitu* (engl. Detailed appropriately), *estimoitu* (engl. Estimated), *ajankohtainen* (engl. Emergent) ja *priorisoitu* (engl. Prioritized). (Pichler, 2010) Esimerkki tuotteen kehitysjonosta DEEP-tilassa voidaan nähdä Kuvassa 3.

Kuva 3: Tuotteen kehitysjoono DEEP -tilassa (Deemer et al., 2010).

Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort	New Estimates of Effort Remaining as of Sprint...					
					1	2	3	4	5	6
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5						
As a buyer, I want to remove a book in a shopping cart	...	2	6	2						
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13						
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20						
Upgrade all servers to Apache 2.2.3	...	5	5	13						
Diagnose and fix the order processing script errors (bugzilla ID 14823)	...	6	2	3						
As a shopper, I want to create and save a wish list	...	7	7	40						
As a shopper, I want to add or delete items on my wish list	...	8	4	20						

Sopivan detaljoitu tuotteen kehitysjoono viittaa siihen, että korkeamman prioriteetin kohteet kuvataan yksityiskohtaisemmin kuin alemman prioriteetin kohteet. Tämä voidaan havaita myös Kuvassa 3. Tälläisen järjestelyn mukaan tuoteomistaja listaa aina epiikit (engl. epic) eli laajemman tason toiminnallisuudet listan loppuun. Epiikeistä muodostetaan myöhemmin tarkempia vaatimuksia, jotka sijoitetaan listan keskelle. Kun nämä vaatimukset tulevat prioriteetiltaan ajankohtaisiksi, pilkotaan ne vielä pienemmiksi vaatimuksiksi, jotta ne voidaan toteuttaa seuraavan sprintin aikana. Tämän ohjeen noudattaminen pitää kehitysjonon ytimekkäänä ja varmistaa, että seuraavassa sprintissä todennäköisesti toteutettavat asiat ovat toimivia. (Pichler, 2010)

Estimoitu tuotteen kehitysjohto tarkoittaa sitä, että kaikkien vaatimusten tulee olla esitettävissä eli niiden työmäärä tulee voida arvioida. Arviot ovat karkearakeisia ja ne ilmaistaan usein tarinapisteinä tai ihanteellisina päivinä. (Pichler, 2010) Tuoteomistaja suorittaa estimoinnin yhdessä kehitystiimin kanssa, mutta tuoteomistaja ei itsessään tee työmääräarviota vaatimuksille, vaan tästä vastaa kehitystiimi. Tuoteomistajan tehtävänä on olla enemmänkin mukana estimoinnissa selventäen vaatimuksiin liittyviä yksityiskoh- tia niin, että kehitystiimi ymmärtää mistä on kyse. Samalla hänen tehtävänä on kirjata estimoinnit tuotteen kehitysjohtoon. (Schwaber ja Sutherland, 2020) Osallistuminen vaati- musten estimointiin on tärkeää, sillä se auttaa tuoteomistajaa priorisoimaan vaatimuksia ja arvioimaan koko projektin kestoa (Pichler, 2010).

Ajankohtainen tuotteen kehitysjohto on aina päivitetty ajantasalle. Tuotteen kehitysjohtos- sa ei saa olla turhia elementtejä. Kaikkia vaatimuksia ei tarvitse myöskään kirjoittaa heti ensimmäisessä sprintissä vaan tyypillisesti tuoteomistaja kirjoittaa ylös vain ne vaatimuk- set, jotka ovat tarpeellisia tulevaan sprinttiin. Myöhemmin vaatimuksia voidaan kirjoittaa lisää, kun tuotteesta ja sen asiakkaasta opitaan lisää. Myös vaatimusten estimaatteja ja kokoa voidaan muokata. Vaatimuksia voidaan myös poistaa tarpeen tullen. (Pichler, 2010; Schwaber ja Sutherland, 2020)

Priorisoitu tuotteen kehitysjohto kuvaa sitä, että vaatimukset ovat listattu tärkeysjärjes- tykseen. Vaatimusten priorisointi on itsessään täysin tuoteomistajan vastuulla (McGreal ja Jocham, 2018; Unger-Windeler ja Schneider, 2019). Tuoteomistaja edustaa asiakasta prio- risoimalla tuotteen vaatimuksia niiden liiketoiminnallisen arvon, riskien, kustannuksien, koon ja riippuvuuksien perusteella varmistaen, että asiakkaille tärkeimmät ominaisuudet toteutetaan mahdollisimman nopeasti. Tarkoituksena on pyrkiä maksimoimaan asiakkaan saama arvo ja hyöty tuotteesta. (Pichler, 2010; Bass et al., 2018; McGreal ja Jocham, 2018; Ghiba, 2022) Scrumin menetelmäkehys ei itsessään kerro, miten priorisointi tapahtuu ja priorisointimenetelmiä löytyykin kirjallisuudesta useita. Esimerkiksi Borhan et al., 2019 ja Bukhsh et al. (2020) mukaan analyyttinen hierarkiaprosessi (AHP) on laajimmin käy- tetty menetelmä ketterissä menetelmissä. AHP:ssa vaatimukset kuvataan hierarkioiden avulla. Kukin vaatimus asetetaan hierarkiatasolle ja kunkin hierarkiatason vaatimuksia arvioidaan pareittain keskenään mittaamalla niitä ja laskemalla painoarvoja. Laskelmien jälkeen, tuoteomistaja voi asettaa vaatimukset paremmuusjärjestykseen. Vaikka AHP on osoittautunut erittäin hyväksi tekniikaksi, sekä Borhan et al., 2019 että Bukhsh et al. (2020) ovat tutkimuksissaan havainneet, AHP:n kärsivän skaalautuvuusongelmista. Toi- sekseen menetelmä on erittäin aikaa vievä prosessi, kun vaatimuksia on paljon. Tästä

johtuen tuoteomistajan onkin hyvä selvittää myös muiden priorisointimenetelmien soveltuvuutta projektiin. Toisaalta tuoteomistaja voi myös selvittää miten AHP:n ongelmia pystyttäisiin minimoimaan. Esimerkiksi Bukhsh et al. (2020) toteavat, että koneoppimisen hyödyntäminen on osoittautunut potentiaalisesti priorisointimenetelmien tehostamisessa.

3.3.5 Osallistuminen ja yhteistyö Scrumin tapahtumissa

On tärkeää, että tuoteomistaja osallistuu Scrumissa järjestettäviin tapahtumiin ja tekee niissä yhteistyötä Scrum-projektin osapuolien kanssa. Sprintin tapahtumat toimivat kommunikaatiokanavana tuoteomistajalle (Matturro et al., 2018; Ghiba, 2022). Saman aikaisesti tuoteomistaja voi osoittaa olevansa Scrum-tiimin käytettävissä ja motivoida tiimiä työskentelemään tehokkaammin (Sverrisdottir et al., 2014; Oomen et al., 2017; Matturro et al., 2018). Toisaalta tapahtumiin osallistumalla tuoteomistaja voi kerryttää tietämystä projektin tilanteesta Scrum-tiimiltä, jonka hän voi samalla kommunikoida myös asiakkaan suuntaan (Bass et al., 2018; Matturro et al., 2018).

Sprintin suunnittelupalaveri on yksi tärkeimpiä Scrumin tapahtumia tuoteomistajalle, sillä hän on vetovastuussa tapahtumasta (Matturro et al., 2018). Tuoteomistajan onkin syytä valmistautua tapahtumaan huolellisesti esimerkiksi käymällä materiaaleja läpi, kirjoittamalla uusia epiikkejä ja keskustelemalla asiakkaan kanssa (Ghiba, 2022). Tapahtumassa tuoteomistajan odotetaan selventävän tuotteen kehitysjonossa olevia vaatimuksia ja vastaamaan kysymyksiin näistä. Tehtävänä on auttaa tiimiä ymmärtämään, mitä seuraavassa sprintissä olisi järkevää tehdä. (Pichler, 2010; Schwaber ja Sutherland, 2020) Sprintin suunnittelupalaveri onkin oivallinen tapahtuma tuoteomistajalle, jossa tuoteomistaja voi muistuttaa tuotteen visiosta ja johdatella tiimiä toimimaan sen mukaisesti tuotteen kehitysjonon voimin (McGreal ja Jocham, 2018). Näiden perusteella Scrum-tiimi miettii, kuinka paljon voidaan tehdä ja miten työtehtävät järjestetään. Toisin sanoen, tuoteomistaja ei kerro tiimille, kuinka paljon työtä sprintissä tulee tehdä tai määrittää tehtäviä tiimin puolesta. Nämä ovat täysin kehitystiimin vastuulla. Kehitystiimin tulee kuitenkin ymmärtää milloin tehtävät täyttävät Definition of Done -määritelmän, ja tämä on täysin tuoteomistajan vastuulla selventää kehitystiimille. (Pichler, 2010; Schwaber ja Sutherland, 2020)

Matturro et al. (2018) mukaan tuoteomistajien osallistuminen sprintin aikana tapahtuviin päivittäispalaverihin on vähäistä. Scrum-opas mainitseekin, että tuoteomistaja voi osallistua päivittäispalaveriinkin vain, mikäli hän toteuttaa jotain sprintin kehitysjonon kohtia.

Osallistuminen olisi kuitenkin tästä huolimatta aina tärkeää, sillä se on loistava tilaisuus kommunikoida tiimin kanssa. Ghiba (2022) väittävätkin, että tuoteomistajien olisi suotuisaa ottaa jopa vetovastuu päivittäispalavereista, sillä tämä osoittaisi kehitystiimille, että tuoteomistaja olisi heidän käytettävissä aina tarpeen tullen. Toisaalta päivittäispalaverit ovat loistava tilaisuus tuoteomistajalle ymmärtää projektin edistymistä ja nähdä, tarvitseeko kehitystiimi apua. Hän voi myös jakaa tietoa projektista, kommunikoida tuotevisiota ja päivittää tietoa omista tekemisistään. Tuoteomistajan on kuitenkin tärkeää ottaa huomioon se, ettei hän saa häiritä liikaa tiimin itseorganisoitumista. (Pichler, 2010)

Tuoteomistajan tulee osallistua jokaiseen sprintin arviointitapaamiseen. Tässäkin tapah-
tumassa tuoteomistajalla on vetovastuu (Matturro et al., 2018) ja mahdollisuus vahvis-
taa tuotevisiota kaikille projektin osapuolille (McGreal ja Jocham, 2018). Tuoteomistaja
aloittaa tapaamisen vertaamalla tuotteen saavutettuja inkrementtejä sprintin tavoittee-
seen, jotta projektin edistyminen voidaan määritellä. Tuoteomistajan on tarkoitus tarkis-
taa kaikki inkrementit perusteellisesti. Hän hyväksyy tai hylkää inkrementit esimerkiksi
testaamalla toimintoja. Näistä tuoteomistajan tulee antaa selkeää ja rakentavaa palautetta
kehitystiimille. Kun edistyminen on selvitetty, tuoteomistajan tehtävänä on pyytää palau-
tetta myös asiakkaalta ja sidosryhmiltä. Kehittäjien, asiakkaiden ja muiden sidosryhmien
kanssa käytävän yhteistyön tarkoitus on etenkin hienosäätää tuotetta ja sen kehityspolkua.
(Pichler, 2010)

Matturro et al. (2018) tutkimuksessa ollaan havaittu, että tuoteomistajien osallistuminen
retrospektiiveihin on suhteellisen vähäistä. Tämän valossa voidaan tehdä toteamus sii-
tä, että tuoteomistajaa ei yleisesti pidetä täysin tiimin ydinjäsenenä. Tämä kuitenkin sotii
vastaan Scrum menetelmäkehyksen periaatteita ja arvoja, missä tuoteomistaja tekee jatku-
vaa yhteistyötä Scrum-tiimin kanssa ja osallistuu heidän toimintaansa. Näin ollen olisikin
tärkeää, että tuoteomistaja osallistuu säännöllisesti retrospektiiveihin. Retrospektiivissä
tuoteomistaja voi etenkin vahvistaa suhdettaan muun Scrum-tiimin kanssa ja kerätä pa-
lautetta omasta toiminnastaan tai antaa sitä muille tiimin jäsenille. Samoin hän voi myös
udella tiimiltä, miten esimerkiksi tuotevision viestintä on onnistunut heidän mielestään
ja parantaa tämän tehokkuutta erilaisten kysymysten avulla. (Pichler, 2010; McGreal ja
Jocham, 2018)

4 Yhteenvedo

Tässä kandidututkielmassa tutkittiin tuoteomistajan roolia Scrumin kontekstissa käyttäen lähteinä Scrum-opasta ja eri empiirisiä tutkimuksia aiheesta. Tavoitteena oli löytää vastaus päätutkimuskysymykseen: ”*Mikä on tuoteomistajan rooli Scrumissa?*”. Tämän perusteella muotoiltiin kaksi alakysymystä, ”*Mikä on tuoteomistajan määritelmä Scrumin kontekstissa?*” ja ”*Mitkä ovat tuoteomistajan tehtävät Scrumin kontekstissa?*”. Kunkin kysymyksen havainnot ja päätelmät esitetään alla.

Mikä on tuoteomistajan määritelmä Scrumin kontekstissa?

Tuoteomistaja on yksi henkilö Scrum-tiimissä, joka edustaa asiakasta ja muita sidosryhmiä Scrum-tiimille. Koska Scrumia käytetään eri tavoin eri organisaatioissa ja jopa saman organisaation eri osastojen ja Scrum-tiimien välillä, tulee tuoteomistajan olla henkilö kuka pystyy tekemään yhteistyötä sekä luomaan empatiaa kaikkien osapuolien välillä. Toisekseen hänellä tulee olla kyky tehdä vaikeita päätöksiä pohjautuen tuotteen visioon. Samanaikaisesti tuoteomistajan tulee tuntea oma toimialueensa ja omata teknillistä osaamista. Nämä ominaisuudet tekevät tuoteomistajasta luotettavan henkilön projektin osapuolien keskuudessa, joka pystyy suorittamaan tehtäviään missä tahansa Scrum-projektissa.

Mitkä ovat tuoteomistajan tehtävät Scrumin kontekstissa?

Tuoteomistajan tehtävät koostuvat vaativasta kokonaisuudesta Scrumissa, jotka varmistavat maksimaalisen arvon tuottamisen asiakkaalle. Näitä tehtäviä ovat vaatimusten määrittely ja tuotevision rakentaminen, julkaisun suunnittelu, vaatimusten kirjoittaminen, tuotteen kehitysjonon hallinta sekä osallistuminen ja yhteistyö Scrumin tapahtumissa.

Mikä on tuoteomistajan rooli Scrumissa?

Tuoteomistajan rooli Scrum-projektissa on keskeinen menestyksen kannalta. Hän on päävastuussa projektin onnistumisesta ja edustaa asiakasta ja muita sidosryhmiä Scrum-tiimille suorittaessaan vaativia tehtäviä, joiden avulla Scrum-tiimin työn tuottavuuden ja tehokkuuden kasvu tuottaa asiakkaalle maksimaalisen arvon tuotteesta. Onnistuakseen tehtävässään tuoteomistajan on kerättävä luottamusta projektin eri osapuolilta. Tämä voidaan saavuttaa tuntemalla oma toimialue, kommunikoimalla ja tekemällä yhteistyötä muiden osapuolien kanssa, luomalla empatiaa heidän välillä sekä tekemällä päätöksiä pohjautuen tuotteen visioon.

Lähteet

- Abrahamsson, P., Salo, O., Ronkainen, J. ja Warsta, J. (2017). "Agile software development methods: Review and analysis".
- Bass, J. M., Beecham, S., Razzak, M. A., Canna, C. N. ja Noll, J. (2018). "An empirical study of the product owner role in scrum". Teoksessa: *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings*, s. 123–124.
- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. ja Thomas, D. (2001). *Ketterän ohjelmistokehityksen julistus*. URL: <https://agilemanifesto.org/iso/fi/manifesto.html> (viitattu 13.09.2022).
- Benassi, J. L. G., Amaral, D. C. ja Ferreira, L. D. (2016). "Towards a conceptual framework for product vision". *International Journal of Operations & Production Management*.
- Blasquez, I. ja Leblanc, H. (2018). "A guideline to teach agile requirements". Teoksessa: *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, s. 360–361.
- Borhan, N. H., Zulzalil, H. ja Sa'adah Hassan, N. M. A. (2019). "Requirements prioritization techniques focusing on agile software development: A systematic literature". *International Journal of Scientific and Technology Research* 8.11, s. 2118–2125.
- Bukhsh, F. A., Bukhsh, Z. A. ja Daneva, M. (2020). "A systematic literature review on requirement prioritization techniques and their empirical evaluation". *Computer Standards & Interfaces* 69, s. 103389.
- Cho, J. (2008). "Issues and challenges of Agile Software Development with Scrum". *Issues in Information Systems* 9.2.
- Cohn, M. (2004). *User stories applied: for agile software development*. Addison-Wesley signature series. Boston, MA: Addison-Wesley. ISBN: 978-0-321-20568-1.
- Deemer, P., Benefield, G., Larman, C. ja Vodde, B. (2010). "The scrum primer".
- DeGrace, P. ja Stahl, L. H. (1990). *Wicked problems, righteous solutions*. Yourdon Press.
- Diebold, P., Ostberg, J.-P., Wagner, S. ja Zendler, U. (2015). "What do practitioners vary in using scrum?" Teoksessa: *International Conference on Agile Software Development*. Springer, s. 40–51.

- Dybå, T. ja Dingsøyr, T. (2008). "Empirical studies of agile software development: A systematic review". *Information and software technology* 50.9-10, s. 833–859.
- Ghiba, A. C. (2022). "The Product Owner Role in a Contemporary Agile Team." *Annals of the University Dunarea de Jos of Galati: Fascicle: I, Economics & Applied Informatics* 28.1.
- Kantola, K., Vanhanen, J. ja Tolvanen, J. (2022). "Mind the product owner: An action research project into agile release planning". *Information and Software Technology* 147.
- Kassab, M. (2015). "The changing landscape of requirements engineering practices over the past decade". Teoksessa: *2015 IEEE fifth international workshop on empirical requirements engineering (EmpiRE)*. IEEE, s. 1–8.
- Kittlaus, H.-B. ja Fricker, S. (2017). *Software product management*. Springer.
- Layton, M. C. (2022). *Scrum for dummies*. John Wiley & Sons.
- Lucassen, G., Dalpiaz, F., Werf, J. M. E. van der ja Brinkkemper, S. (2016). "The use and effectiveness of user stories in practice". Teoksessa: *International working conference on requirements engineering: Foundation for software quality*. Springer, s. 205–222.
- Mahnic, V. (2011). "A case study on agile estimating and planning using scrum". *Elektronika ir Elektrotechnika* 111.5, s. 123–128.
- Matturro, G., Cordovés, F. ja Solari, M. (2018). "Role of product owner from the practitioner's perspective. An exploratory study". Teoksessa: *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, s. 113–118.
- McGreal, D. ja Jocham, R. (2018). *The professional product owner: Leveraging scrum as a competitive advantage*. Addison-Wesley Professional.
- Oomen, S., De Waal, B., Albertin, A. ja Ravesteyn, P. (2017). "How can scrum be successful? Competences of the scrum product owner".
- Paasivaara, M., Durasiewicz, S. ja Lassenius, C. (2009). "Using scrum in distributed agile development: A multiple case study". Teoksessa: *2009 Fourth IEEE International Conference on Global Software Engineering*. IEEE, s. 195–204.
- Paasivaara, M., Heikkilä, V. T. ja Lassenius, C. (2012). "Experiences in scaling the product owner role in large-scale globally distributed scrum". Teoksessa: *2012 IEEE Seventh International Conference on Global Software Engineering*. IEEE, s. 174–178.
- Pichler, R. (2010). *Agile Product Management with Scrum: Creating Products That Customers Love*. Peaesons Education, Inc.
- Sarkan, H. M., Ahmad, T. P. S. ja Bakar, A. A. (2011). "Using JIRA and Redmine in requirement development for agile methodology". Teoksessa: *2011 Malaysian Conference in Software Engineering*. IEEE, s. 408–413.

- Schwaber, K. (1995). "Scrum development process". Teoksessa: *Business object design and implementation*. Springer, s. 117–134.
- (2004). *Agile project management with Scrum*. Microsoft press.
- Schwaber, K. ja Sutherland, J. (2020). "The scrum guide". *Scrum Alliance* 21.19, s. 3.
- Sedano, T., Ralph, P. ja Péraire, C. (2019). "The product backlog". Teoksessa: *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, s. 200–211.
- Stoica, M., Mircea, M. ja Ghilic-Micu, B. (2013). "Software development: agile vs. traditional." *Informatica Economica* 17.4.
- Sutherland, J. ja Schwaber, K. (2011). "The scrum papers: nut, bolts, and origins of an Agile framework". *Scrum inc*.
- Sverrisdottir, H. S., Ingason, H. T. ja Jonasson, H. I. (2014). "The role of the product owner in scrum-comparison between theory and practices". *Procedia-Social and Behavioral Sciences* 119, s. 257–267.
- Takeuchi, H. ja Nonaka, I. (1986). "The new new product development game". *Harvard business review* 64.1, s. 137–146.
- Tessarolo, P. (2007). "Is integration enough for fast product development? An empirical investigation of the contextual effects of product vision". *Journal of Product Innovation Management* 24.1, s. 69–82.
- Trieflinger, S., Münch, J., Roling, B., Voigt, Y. ja Lang, D. (2021). "Provide direction despite uncertainty: workshops for creating a compelling product vision". Teoksessa: *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, s. 212–215.
- Unger-Windeler, C. ja Schneider, K. (2019). "Expectations on the Product Owner Role in Systems Engineering-A Scrum Team's Point of View". Teoksessa: *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, s. 276–283.
- Vähäniitty, J., Lassenius, C. ja Rautiainen, K. (2002). "An approach to product roadmapping in small software product businesses". Teoksessa: *ECSQ2002, Conference Notes*, s. 12–13.

