

Scrum-viitekehyksen tuoteomistajan rooli Ticorporate-yrityssimulaatiossa

Jukka Riikonen

Opinnäytetyö
Marraskuu 2020
Liiketalouden ala
Tradenomi (AMK), Tietojenkäsittelyn tutkinto-ohjelma

Tekijä(t) Riikonen, Jukka	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 11/2020
	Sivumäärä 59	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Scrum-viitekehyksen tuoteomistajan rooli Ticorporate -yrityssimulaatiossa		
Tutkinto-ohjelma Tietojenkäsittelyn tutkinto-ohjelma		
Työn ohjaaja(t) Niko Kiviaho		
Toimeksiantaja(t) Tietojenkäsittelyn tutkinto-ohjelma, Jyväskylän ammattikorkeakoulun liiketoimintayksikkö		
<p>Tiivistelmä</p> <p>Korkeakoulut etsivät jatkuvasti ratkaisuja koulutusohjelmiinsa työelämän nopeasti muuttuviin ammattilatarpeisiin vastaamiseksi. Tietojenkäsittelyn tutkinto-ohjelma Jyväskylän ammattikorkeakoulussa on ottanut käyttöön ainutlaatuisen tavan tuottaa taitavia ja sopeutumiskykyisiä ammattilaisia Ticorporate Demo Lab-yrityssimulaation avulla, jossa 2. lukuvuoden opiskelijat oppivat ketterää ohjelmistokehitystä kehittämällä pelejä tai sovelluksia täysipäiväisesti pienissä tiimeissä.</p> <p>Simulaatioon osallistuvat oppilaat erikoistuvat valitsemalla päätoimen ja sivutoimen itselleen. Sivutoimeksi voi valita testauksen, liiketoiminnan, sisällöntuotannon, Scrummasterin tai tuoteomistajan roolin. Tuoteomistajan rooli Demo Labissa tuli mahdolliseksi vasta Ticorporate 7-toteutuksessa v.2020.</p> <p>Tutkimuksen tavoitteena oli tutkia ketterän Scrum-viitekehyksen tuoteomistajan roolia osana Ticorporate-yrityssimulaatiota kehittämistutkimuksen keinoin. Tarkoituksena oli selvittää, kuinka tuoteomistajan sivutoimea voitaisiin kehittää. Alkutilanne selvitettiin analysoimalla, kuinka tuoteomistajuutta on käsitelty ohjelmistotuotannon opintojaksolla, ennen Ticorporaten alkamista ja sivutoimen valitsemista. Lisäksi Demo Labin rakenne ja tarpeet tuoteomistaja-sivutoimeen liittyen tutkittiin. Tuoteomistajan rooliin liittyvää materiaalia analysoitiin vertaamalla sitä Scrumin teoriaan ja Demo Labin tarpeisiin, ja Ticorporate 7-toteutuksen tuoteomistajien kokemuksia selvitettiin ryhmäteemahaastattelun keinoin.</p> <p>Tutkimuksen tuloksena tuotettiin konkreettisia kehittämissuhteita Ticorporateen, sivutoimen käytänteisiin ja rakenteeseen tulevien toteutusten parantamiseksi. Jatkotutkimus on tarpeen edistyneempien tuotekehitysmenetelmien tuottamiseksi tuoteomistajan sivutoimessa.</p>		
Avainsanat (asiasanat) Scrum, ketterät menetelmät, tuoteomistaja, Ticorporate, ohjelmistotuotanto, tuotekehitys		
Muut tiedot (Salassa pidettävät liitteet)		

Author(s) Riikonen, Jukka	Type of publication Bachelor's thesis	Date 11/2020 Language of publication: Finnish
	Number of pages 59	Permission for web publication: x
	Title of publication Scrum Product Owner's role in Ticorporate business simulation	
Degree programme Business Information Systems		
Supervisor(s) Kiviaho, Niko		
Assigned by Business Information Systems degree programme, JAMK University of Applied Sciences		
Abstract <p>Universities are constantly searching for ways to improve upon their degree programmes to respond to the rapidly changing requirements of businesses' needs for professionals. The Business Information Systems degree programme of JAMK University of Applied Sciences has adopted unique approach to produce skilled and adaptable professionals by using Ticorporate Demo Lab business simulation, in which second-year bachelor students learn agile product development by developing games or software full-time in small teams.</p> <p>Students participating in the simulation choose a main subject and a side subject to specialize in. Side subjects consist of Scrummaster, tester, business, content producer and Product Owner. The role of a Product Owner in Ticorporate was made available in 2020.</p> <p>The objective of this study was to explore agile Scrum framework's role of the Product Owner in Ticorporate business simulation via action research and how to develop it even further. The established theory of Scrum from the Product Owner's perspective was studied as well as the needs and the structure of the Ticorporate business simulation. Existing Product Owner related material was analyzed, and current Product Owners' experiences of the side subject were collected by group interview. Scrum theory, subject material, and interviewees' experiences in relation to simulations' Product Owner structure were cross-examined.</p> <p>As a result of the analysis, development and implementation recommendations of basic product development practices were produced to existing material and subject structure to enhance the future iterations of Product Owner side subject. Further study is required for more advanced product development techniques for the Product Owner side subject.</p>		
Keywords/tags (subjects) Scrum, Agile, Product Owner, product development, Ticorporate, software development		
Miscellaneous (Confidential information)		

Sisältö

1	Johdanto	3
2	Tutkimusasetelma	4
2.1	Tavoitteet ja tutkimusongelmat	4
2.2	Tutkimusmenetelmät.....	5
3	Scrum -viitekehys	9
3.1	Pääperiaatteet ja roolit	9
3.2	Scrumin tuotokset.....	11
3.3	Tapahtumat	12
3.4	Säännöt ja arvot	14
3.5	Ketterän ohjelmistokehityksen historiaa	16
3.6	Tuoteomistaja	19
4	Tuoteomistajuus osana ohjelmistotuotantoa ja Ticorporatea	27
4.1	Ohjelmistotuotanto	27
4.2	Ticorporate Demo Lab.....	32
5	Tuoteomistajan sivutoimi Demo Labissa	35
5.1	Viikkotapaamiset	36
5.2	Luennot	37
5.3	Asiantuntijahaastattelu.....	40
5.4	Käyttäjäkertomus- ja Zenhub-työpaja	42
5.5	Manuaali.....	43
5.6	Julkaisut ja tuotantovaiheet	47
5.7	Tuoteomistajien kokemuksia Demo Labista	48
5.8	Muita sivutoimen puutteita	49
6	Kooste kehittämis ehdotuksista	50
7	Pohdinta	51
7.1	Tulokset	51
7.2	Tutkimuksen luotettavuustarkastelu	51
7.3	Jatkotutkimus	53

Lähteet	54
Liitteet	56
Liite 1. Teemahaastattelun teemat	56

Kuviot

Kuvio 1. Kehittämistutkimuksen prosessi	9
Kuvio 2. Käyttäjäkertomus Zenhub-sovelluksessa.	23
Kuvio 3. Kuvakaappaus Zenhub-sovelluksesta.....	43

1 Johdanto

Ammattikorkeakoulut etsivät jatkuvasti uusia keinoja työelämään valmentavien opintokokonaisuuksien kehittämiseksi ja työnantajien ja työelämän muuttuviin vaatimuksiin vastaamiseksi. Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelman opetussuunnitelma uudistui vuonna 2013, jonka seurauksena tutkinto-ohjelma otti käyttöön Ticorporate-yrityssimulaation oppilaiden sisäisen ja ulkoisen yrittäjyyden, projektityöskentelyn ja teknisen osaamisen oppimisen tukemiseksi. Ticorporate-yrityssimulaation tarkoituksena on mallintaa työelämässä tapahtuvaa tiimityöskentelyä, opettaa tarvittavia vuorovaikutustaitoja, syventää asiantuntijuutta ja valmentaa opiskelijoita projektiluontoiseen tapaan työskennellä kehittämällä sovelluksia tai pelejä. (Bister, Immonen & Tuikka 2016, 202–204). Ticorporatessa opiskelijat valitsevat yhden päätoimisen ja sivutoimisen roolin, jotka määrittävät opiskelijan ydinosamisalueen ja vastuut projekteissa. Sivutoimisia rooleja ovat testaaja, liiketoiminta, sisältötuotanto ja Scrum-viitekehikseen kuuluvat Scrummaster ja tuoteomistaja. (Pölkki 2020.)

Menestyvien sovellusten tai pelien kehittäminen vaatii vahvaa visiota tuotteesta ja ymmärrystä markkinoista ja asiakkaista. Ticorporaten tuotekehitysprojekteissa käytetään ketterien sovelluskehitysmenetelmien Scrum-viitekehystä, joka on empiirinen ja iteratiivinen tuotekehityksen menetelmä. Scrumin tunnistama Product Owner -rooli eli tuoteomistaja huolehtii tuotteen vision toteutumisesta ja siitä, mitä tuotteen kehityssyklin aikana kehitetään. Tuoteomistajan tarkoituksena on maksimoida tuotteen sidosryhmille tuottama arvo tuotannon aikana.

Tutkijan aloittaessa työharjoittelunsa tammikuussa 2020 Ticorporate 7 -toteutuksessa tuoteomistajuus päätettiin ottaa osaksi Ticorporatea. Opinnäytetyön päätavoitteena on siten kehittää tuoteomistajuutta sivutoimena Ticorporatessa. Tutkija on aiemmin toiminut v.2015-2016 Ticorporate 3 -toteutuksessa Scrummasterin roolissa.

2 Tutkimusasetelma

Tutkimuksen toimeksiantajana toimii Jyväskylän ammattikorkeakoulun liiketoimintayksikkö. Jyväskylän ammattikorkeakoulu on Suomen 7. suurin korkeakoulu, jossa työskentelee 700 henkilökunnan jäsentä ja opiskelee 8500 opiskelijaa. Ticorporate on tietojenkäsittelyn tutkinto-ohjelmaan kuuluva 25 opintopisteen opintokokonaisuus 2. vuoden opiskelijoille, ja siihen osallistuu vuosittain n. 40 opiskelijaa, joista 6–8 toimii tuoteomistajina. Vuoden 2022 toteutuksessa opiskelijamäärä todennäköisesti kaksinkertaistuu, sillä syksyllä 2020 tietojenkäsittelyn tutkinto-ohjelmaan avautui peliteollisuuteen keskittyvä, englanninkielinen tutkinto-ohjelma.

2.1 Tavoitteet ja tutkimusongelmat

Scrum-viitekehukseen kuuluu kolme virallista roolia: kehitystiimin jäsen, Scrummaster ja tuoteomistaja. Ticorporate 7 -toteutuksessa sivutoimen vaihtoehdoksi lisättiin tuoteomistaja. Mitä tuoteomistajuuden suhteen tulee huomioida Ticorporaten kontekstissa? Kuinka Scrumin tuoteomistajan roolia voidaan soveltaa Ticorporaten tarpeet huomioon ottaen? Jotta Scrum -viitekehystä voidaan opettaa kokonaisuutena Ticorporatessa, tulisi tuoteomistajuuden roolia kehittää. Opinnäytetyön tavoitteena on tuottaa siten kehitysehdotus siitä, kuinka tuoteomistajan roolia voidaan jatkokehittää 2021 alkavaan Ticorporate toteutukseen. Tutkija on toiminut kohdeorganisaatiossa tutkimuksen ajan.

Opinnäytetyössä pyritään vastaamaan seuraaviin kysymyksiin:

1. Mikä on tuoteomistajan rooli Scrum-viitekehyksessä?
2. Mikä on Ticorporate -yrityssimulaatio?
3. Miten tuoteomistajan roolia voidaan kehittää Ticorporatessa?

2.2 Tutkimusmenetelmät

Opinnäytetyön tutkimusmenetelmäksi soveltui parhaiten kehittämistutkimus, sillä kehittämistutkimuksen tavoitteet ovat linjassa opinnäytetyön tavoitteiden kanssa. Bisterin (2019) mukaan kehittämistutkimuksella on kaksi ydintavoitetta: kehittäminen ja vaikuttaminen. Kehittämisellä tavoitellaan käytänteiden, toimintatilanteen ja niihin osallistuvien henkilöiden ymmärryksen lisäämistä, kun taas vaikuttamisella pyritään yhdistämään teoria ja käytäntö. (Mts. 44.)

Kehittämistutkimukselle ominaista on konkreettisen edun ja hyödyllisten ratkaisujen aikaansaaminen toimeksiantajalle ja muutos alkutilanteeseen. Tutkijan tulee löytää ilmiöön liittyvä teoriapohja, johon tutkimuskohde liittyy, ja jonka kautta tutkija voi löytää uutta tietoa kehittämisen toteutumiseksi. (Bister 2019, 26.) Kanasen (2016, 33) mukaan kehittämistutkimuksen lähtökohtana on muutostarve parempaan. Kanasen mielestä kehittämistutkimukselle ominaista on kehittämisen ja tieteellisen tutkimuksen synteesi syklisessä muodossa. Tämä kehityssykli muodostuu alkutilan ongelmasta, ratkaisuehdotuksesta, toteutuksesta ja lopputuloksesta. (Mts. 33.) Opinnäytetyön tapauksessa alkutilan ongelmaksi voidaan määrittää tuoteomistajan rooli: kuinka sitä voidaan kehittää ja järjestää käytännössä Ticoporateen kuuluvana sivutoimena?

Kehittämistutkimus ei ole erillinen tutkimusmenetelmä, vaan tarkkaan valittu kokonaisuus soveltuvia tutkimusmenetelmiä. Kehittämistutkimus voidaan määritellä monimenetelmäisenä tutkimusstrategiana, jossa hyödynnetään kvalitatiivisia ja kvantitatiivisia tutkimusmenetelmiä tutkimusongelman ratkaisemiseksi, ja muutoksen aikaansaamiseksi tavoitteen mukaisesti. (Kananen 2016, 33–34.) Pernaan (2013) määrittelyn mukaan kehittämistutkimuksessa pyritään vastaamaan kolmeen kysymykseen: miten kehittämisessä edetään, millaisia tarpeita ja mahdollisuuksia voidaan tunnistaa, ja mihin tuotoksiin kehittäminen johtaa. Nämä kysymykset jakavat siten kehittämistutkimuksen kolmeen osaan: kehittämisprosessiin, ongelma-analyysiin ja kehittämistuotokseen. (Mt.)

Ongelma-analyysissä pyritään tunnistamaan muutostarpeet, mahdollisuudet ja haasteet. Ongelma-analyysivaihe voi olla teoreettinen, empiirinen tai yhdistelmä molempia muotoja. Ongelma-analyysin seurauksena tutkimuksen kehittämistavoitteet selkeytyvät, jolloin niiden toteuttamiseksi luodaan kehittämissuunnitelma tutkimuksen ohjaamiseksi. Kehittämissuunnitelmaa parannetaan ja päivitetään koko tutkimuksen ajan, sillä kehittämistutkimus tutkimusmenetelmänä on joustava. (Pernaa 2013.)

Kvalitatiivinen aineistonkeruu kehittämistutkimuksen tukena

Ennen kehittämistyön aloittamista tutkijan tulee ymmärtää tutkittavaa ilmiötä ja ympäristöä tutkimuskohteesta tehtävän kartoituksen avulla. Ilman riittävää alustavaa tiedonkeruuta ilmiöstä tutkija ei voi saavuttaa tarvittavaa ymmärryksen tasoa tutkimuskohteeseen vaikuttamisesta ja kehittämistyöstä. (Bister 2019, 44.) Kehittämistutkimuksen tueksi valikoitui kvalitatiivinen aineistonkeruu, sillä laadullisen tutkimuksen aineiston keruumenetelmien tarkoituksena on ymmärtää tutkimuksen kohdetta syvästi ja laajasti. (Mts. 43–44.) Kanasen (2016, 34–35) mielestä kvalitatiivisessa eli laadullisessa tutkimuksessa ilmiötä ei tunneta, ja tutkittavaa ilmiötä pyritään ymmärtämään havainnoista ja haastatteluista saatujen aineistojen avulla.

Laadullisessa tutkimuksessa tutkimuksen kohteena olevaa ilmiötä, tapahtumaa tai toimintaa yritetään kuvata ja ymmärtää. Tutkittavalle kohteelle pyritään löytämään sopiva teoreettinen tulkinta. Siten tutkimukseen käytetyillä tiedonantajilla tulisi olla kokemusta tai vahva käsitys tutkimuskohteesta. Täten tiedonantajien valintaan tulee käyttää harkintaa, ja valinnan tulee tukea tutkimuksen tarkoitusta. Tutkimusraportissa harkinta ja valinnat tulee perustella, jotta lukija voi itse päätellä niiden tieteellisen validiteetin. Tarkkaan valituista tiedonantajista voidaan käyttää nimitystä eliittiotanta. (Tuomi & Sarajärvi 2018, luku 3.4.)

Aineistonkeruumenetelmäksi valittiin teemahaastattelu, sillä Ticorporate 7 -toteutuksessa tutkimukseen soveltuvia tiedonantajia eli tuoteomistajaopiskelijoita oli kuusi ja henkilökunnan avainhenkilöitä kaksi. Kvantitatiiviset menetelmät eivät siten sovi aineistonkeruumenetelmäksi, koska avainhenkilöistä koostuva haastattelu-

joukko on saturaation saavuttamiseksi kvantitatiivisesti liian pieni. Saturaatiossa tiedonantajat eivät enää tuota tutkimuskohteen kannalta oleellista tietoa, ja aineisto alkaa toistua (Tuomi & Sarajärvi 2018, luku 3.4.1).

Havainnointia voidaan myös käyttää tutkimuksessa muiden tutkimusmenetelmien tukena, sillä se on kaikille tieteenhaaroille yhteinen ja tarpeellinen menetelmä (Hirsjärvi & Hurme 2015, 37). Havainnoinnin lisäksi Ticorporateen ja tuoteomistajan sivutoimeen liittyviä dokumentteja voidaan analysoida. Aineistoa analysoidaan teoriaohjaavan analyysimenetelmän mukaisesti, jossa teoria voi toimia analyysin tukena, mutta analyysi ei ole täysin teoriapohjainen. Aikaisempi tieto ohjaa analyysia, ja päättelyketjuja tehdään yhdistelemällä teoriaa ja aineistoa. (Tuomi & Sarajärvi 2018, luku 4.2.)

Teemahaastattelu

Teemahaastattelu on lähempänä strukturoimatonta kuin strukturoitua haastattelua. Teemahaastattelu luokitellaan puolistrukturoiduksi menetelmäksi, sillä haastattelun aihepiirit – teema-alueet – ovat kaikille samat. Muissa puolistrukturoiduissa haastatteluissa esimerkiksi kysymykset, tai jopa kysymysten muoto, ovat kaikille samat. Teemahaastattelusta puuttuu strukturoidulle lomakehaastattelulle luonteenomainen kysymysten tarkka muoto ja järjestys, mutta se ei ole täysin vapaa niin kuin syvähaastattelu. (Tuomi & Sarajärvi 2018, luku 3.1.1.) Teemahaastattelu eroaa muista haastattelumenetelmistä siten, että haastattelu rakentuu ennen haastattelua määriteltyjen teemojen ja kysymysten ympärille. Keskustelua voidaan tarkentaa ja ohjata sen mukaan, kuinka haastateltava kysymyksiin vastaa. Siten teemahaastattelun etuna on menetelmän joustavuus. Keskiössä ovat haastateltavien antamat merkitykset ja tulokset. (Mts. luku 3.1.1.)

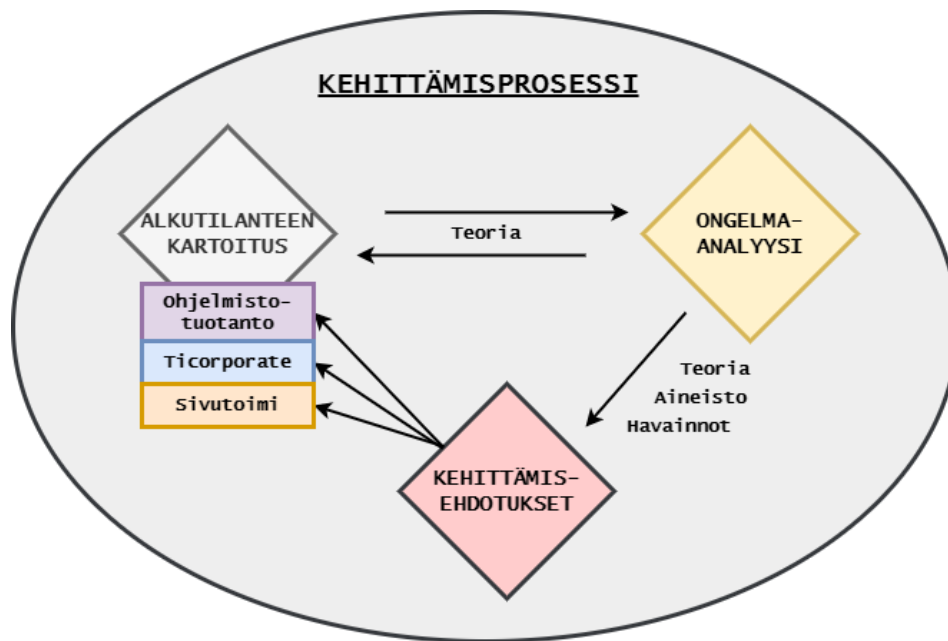
Tuomi ja Sarajärvi (2018) vertaa teemahaastattelua Merton, Fiske ja Kendallin (1956) kohdennettuun haastatteluun. Molemmissa menetelmissä haastateltavat ovat olleet osallisina jossakin tietyssä tilanteessa. Tutkija on perehtynyt tutkittavaan ilmiöön perusteellisesti, jonka perusteella hän on tehnyt oletuksia tilanteen pääpiirteisiin liittyen, ja tilanteen seurauksien vaikutuksiin haastateltavissa. Analysoimalla ilmiötä ja

olettamuksiaan tutkija rakentaa haastattelurungon, jonka avulla haastateltavien kokemuksia tilanteista voidaan tutkia. (Tuomi & Sarajärvi 2018, luku 4.2.3.)

Haastattelun tarkoitus on maksimoida tiedon saanti valitusta aiheesta. Tätä edesauttaa teemojen tai haastattelukysymysten luovuttaminen haastateltaville etukäteen, jotta he voisivat tutustua aiheeseen. Eettisesti hyvä käytäntö on kertoa tiedonantajille etukäteen, mistä on kyse. On todennäköisempää, että tiedonantajat haluavat osallistua haastatteluun, kun tutkimuksen aihe on tiedossa. (Tuomi & Sarajärvi 2018, luku 3.1.) Ryhmähaastattelussa keskustelu ja kommentointi on melko vapaamuotoista. Haastattelijan tehtävänä on pitää keskustelu valittujen teemojen ympärillä ja edesauttaa keskustelua. Haastattelijan tulee huomioida, että kaikilla osallistujilla on yhtäläiset mahdollisuudet osallistua keskusteluun. Ryhmähaastattelu valittiin, koska siten on mahdollista taltioida avainhenkilöiden yhteiset tai eriävät näkemykset ja tulokset aiheeseen liittyen. (Hirsjärvi & Hurme 2015, 61–63.)

Kehittämissykli

Tutkimuksen alussa selvitettiin Ticorporaten alkutilanne. Ticorporateen opiskelijoita valmistelevalle opintojaksolle on ohjelmistotuotanto. Miten tuoteomistajuutta on käsitelty ohjelmistotuotannon opintojaksolla, ja Ticorporaten alkuvaiheessa, ennen kuin opiskelijat valitsevat sivutoimensa? Alkutilakartoituksen jälkeen analysoitiin tuoteomistaja-sivutoimen rakennetta Ticorporaten ja Scrumin teorian näkökulmasta. Tämän vaiheen jälkeen Ticorporaten 7 -toteutuksen päättäneitä tuoteomistajaopiskelijoita teemahaastateltiin. Teemoina käsiteltiin heidän kokemuksiaan tuoteomistajuudesta ennen Ticorporatea, tuoteomistajan roolia ja kuinka he kokivat sivutoimen rakenteen. Lopuksi luotiin kehittämissuositukset sivutoimen parantamiseksi saatujen tulosten perusteella (ks. Kuvio 1).



Kuvio 1. Kehittämistutkimuksen prosessi

3 Scrum -viitekehys

3.1 Pääperiaatteet ja roolit

Scrum on tuotteiden kehittämiseen, hallintaan, ylläpitämiseen ja toimittamiseen tarkoitettu viitekehys, jolla pyritään maksimoimaan tuotteen tuottamaa lisäarvoa tehokkaalla ja luovalla tavalla (Schwaber & Sutherland 2017, 3). Viitekehyksellä tarkoitetaan sitä, että sen sääntöjen sisällä voidaan käyttää muita käyttötarkoituksen soveltuvia prosesseja tai tekniikoita. Scrum ei siis itsessään ole yksityiskohtainen menetelmä tai prosessi. Scrumille ominaista on kokonaisvaltainen läpinäkyvyys tuotehallinnassa, jotta sen eri osa-alueiden tehokkuutta voidaan seurata ja parantaa. (Mts. 3.)

Schwaberin ja Sutherlandin (2017) mukaan Scrumiin kuuluu

- Scrum-tiimi ja siihen sisältyvät roolit
- viitekehysten tuotokset eli artefaktit
- Scrumin tapahtumat
- Scrumin säännöt ja arvot.

Jokaisella osa-alueella on tarkkaan harkittu tarkoitus ja vuorovaikutussuhde, jota Scrumin määrittämät säännöt ohjaavat. (Mts. 3.) Scrum perustuu empirismiin: tieto voidaan johtaa kokemuksesta ja tekemällä tiedettyyn tietoon perustuvia päätöksiä. Viitekehyksenä Scrum pyrkii maksimoimaan ennustettavuutta ja hallitsemaan riskejä olemalla sekä iteratiivinen, eli useaan otteeseen toistuva, että inkrementaalinen, eli vähittäin kasvava tai additiivinen. Empiirisen prosessin kolme peruspilaria ovat läpinäkyvyys, tarkastelu ja sopeutuvuus Schwaberin ja Sutherlandin mielestä. Nämä peruspilarit ovat avainasemassa viitekehyksen sisällä toimimisessa. (Mts. 4–5.)

Scrumin roolit

Scrum määrittelee kolme roolia, jotka muodostavat Scrum-tiimin: Scrummaster, tuoteomistaja ja kehitystiimi (Clinton 2010, 44–45; Schwaber & Sutherland 2017, 5–7). Scrummasterin vastuulla on viitekehyksen sääntöjen edistäminen ohjaamalla ja auttamalla muita ottamaan haltuun Scrumin säännöt ja toimintaperiaatteet. Scrummaster toimii palvelevana johtajana Scrum-tiimille ja auttaa tarvittaessa muokkaamaan toimintaperiaatteita tiimin sisä- ja ulkopuolella. (Sutherland 2017, 5–7.)

Tuoteomistajan päävastuulla on kehitystiimin tuottamista tuloksista muodostuvan tuotteen arvon maksimointi. Avainasemassa on myös **tuotteen kehitysjonon** hallinta erilaisin kehitysjonoon kohdistuvien operaatioiden avulla, jotta se on selkeästi ymmärrettävissä, riittävän tarkka ja priorisoitu tavoitteet huomioon ottaen. (Mts. 5–7.)

Kehitystiimi muodostuu asiantuntijoista, joiden vastuulla on kehitysjonon sisällön toteuttaminen julkaisukelpoiseksi tuotteen osaksi rajatun aikaikkunan eli sprintin sisällä. Ominaista kehitystiimeille on itseohjautuvuus eli tietyntasoinen vapaus toimintaan, monitaitoisuus ja jaettu vastuu kehityksestä. Scrummastereita ja tuoteomistajia on aina yksi henkilö per Scrum-tiimi, mutta kehitystiimiin koko voi vaihdella kolmesta yhdeksään. (Mts. 5–7.)

3.2 Scrumin tuotokset

Scrum-tiimin toiminnan seurauksena syntyvät tuotokset eli **artefaktit** ovat tärkeä osa viitekehystä, joita ilman viitekehystä ei voi kutsua Scrumiksi. Näitä artefakteja ovat tuotteen kehitysjojo, sprintin kehitysjojo ja **inkrementti** (Schwaber & Sutherland 2017, 15–17).

Tuotteen kehitysjojo on käytännössä yksinkertaisimmillaan järjestetty lista kaikista tiedossa olevista tuotteeseen kohdistuvista vaatimuksista, korjauksista ja muutoksista. Näitä kutsutaan tuotejonon kohdiksi. (Schwaber & Sutherland 2017, 15.) Tuoteomistaja on vastuussa kaikista tuotteen kehitysjonon toimenpiteistä, kuten vaatimusten ja muutoksien sisällöstä, kehitysjonon kohtien tärkeysjärjestyksestä ja dokumentin läpinäkyvyydestä. Tuotteen kehitysjojo on jatkuvasti muuttuva, elävä dokumentti, eikä koskaan valmis: sen sisältöä muutetaan tarpeen mukaan vastaamaan ympäristön muuttuvia liiketoiminnan, markkinoiden ja teknologian vaatimuksia Scrumin empiirisen luonteen takia (Mts. 15–16).

Sprintin kehitysjojo muodostuu tuotteen kehitysjonon valituista ja suurinta arvoa tuottavista kohdista sen hetkisen arvion mukaan ja suunnitelmasta niiden muuttamiseksi inkrementiksi sprintin päättyessä. Sprintin kehitysjojo on siten tuoteomistajalle annettu arvio, lupaus ja tavoite kehitystiimin toimesta seuraavan inkrementin toteutuneesta sisällöstä. Kehitystiimin tekemän työn edistymisen tulee olla havaittavissa päivittäisesti, joten sprintin kehitysjonon tulee olla riittävän yksityiskohtainen sen mahdollistamiseksi. Kehitysjojoa muokataan tarpeen mukaan sprintin tavoitteen saavuttamiseksi: kehitysjojoon kirjattua työtä voidaan muokata, lisätä tai poistaa, ja ainoastaan kehitystiimillä on valtuudet tehdä näitä toimenpiteitä. (Mts. 16–17.)

Kaikki sprinttien aikana toteutuneet tuotteen kehitysjonon kohteet muodostavat sprintin päättyessä inkrementin eli lisäyksen tuotteeseen pienen, 'valmiin' kokonaisuuden muodossa. 'Valmiilla' tarkoitetaan sitä, että kaikilla Scrum-tiimin jäsenillä on yhteinen ja jaettu ymmärrys siitä, mitä 'valmis' tarkoittaa: tätä kutsutaan valmiin määritelmäksi. Yhteinen määritelmä on tärkeä siksi, että sana 'valmis' voidaan tul-

kita hyvin subjektiivisesti. Määritelmä on käytännössä ennalta sovittu laadunvarmistuksen kriteeristö, joka pitää huolen siitä, että inkrementti on sen läpäistessään julkaisukelpoinen. (Mts. 17.)

3.3 Tapahtumat

Scrum määrittelee useita siihen kuuluvia säännöllisesti toteutuvia, aikarajattuja tapahtumia, joiden tarkoituksena on pitää muiden kuin Scrum-palaverien tarve mahdollisimman pienenä. Schwaberin ja Sutherlandin (2017, 9) mukaan Scrumiin kuuluvat tapahtumat ovat

- sprintti
- päivittäispalaveri
- sprintin katselmointi
- sprintin retrospektiivi.

Scrumin tärkeimpänä tapahtuma on muut tapahtumat sisäänsä kapseloiva **sprintti**. Sprintti on maksimissaan kuukauteen rajattu ja kestoltaan vakioitu aikaikkuna, jonka sisällä tuotekehitystä tehdään. Sprintin tavoitteena on 'valmis' ja julkaisukelpoinen inkrementti. Sprintin päättyessä uusi sprintti alkaa, ja jokaiselle sprintille tulee olla valituista kehitysjonon kohdista muodostettu sprintin tavoite, johon Scrum-tiimi pyrkii. Mahdolliset muutokset sprintin sisällä eivät saa vaarantaa asetettua tavoitetta. (Mts. 9–10.)

Kehitystiimi voi kuitenkin neuvotella sprintin sisällöistä, ja sitä voidaan tarkentaa tuoteomistajan kanssa. Tuoteomistajalla on myös valta keskeyttää sprintti ennen sen päättymistä, jos tavoitteen toteuttaminen ei ole kannattavaa esimerkiksi markkinoiden muutosten seurauksena. (Mts. 9–10.)

Sprintti aloitetaan sprintin suunnittelupalaverilla, jossa sprintin aikana tehtävät kehitystyöt suunnitellaan Scrum-tiimin kesken. Suunnittelussa keskustellaan, mitä sprintti

tin aikana voidaan tuottaa, ja miten se voitaisiin toteuttaa. Tuoteomistajan ja kehitystiimin neuvottelun tuloksena tuotejonosta valitaan ne kohdat, joiden toteuttamiseen sitoudutaan. Näistä muodostuu sprintin kehitysjono. (Mts. 10–11.)

Scrummaster pitää huolen siitä, että suunnittelu tapahtuu ja osallistujat ymmärtävät tapahtuman tarkoituksen. Suunnittelupalaveri on aikarajattu maksimissaan kahdeksaan tuntiin kuukauden mittaiselle sprintille, ja skaalautuu alaspäin, mikäli sprinttien valittu kesto on tätä lyhyempi. (Mts. 10–11.)

Päivittäispalaveri on 15 minuuttiin rajattu tapahtuma, joka toistuu jokaisena sprintin päivänä. Sen tarkoituksena on suunnitella tehtävät työt ennen seuraavaa päivittäispalaveria ja keskustella kehitystiimin edistymisestä kohti sprintin tavoitetta tarkastelemalla tehtyä työtä, tulevaa työtä ja mahdollisia esteitä. Kehitystiimillä on oikeus päättää, kuinka palaverin sisältö järjestetään. Palaveriin osallistuu kehitystiimi ja Scrummaster: mikäli muita osallistujia on paikalla, he eivät saa häiritä tilaisuutta. Päivittäispalaveri onkin hyvä erinomainen tilaisuus jakaa informaatiota nopeasti ja tehokkaasti tehtävän työn suhteen koko kehitystiimin kesken. (Mts. 12–13.)

Sprintin katselmoinnissa tarkastellaan yhteistyössä Scrum-tiimin ja sidosryhmien kanssa Sprintin aikaisen kehityksen tuloksesta eli inkrementistä ja tarvittaessa tehdään muutoksia tuotteen kehitysjonoon. (Schwaber & Sutherland 2017, 13–14.) Keskustelun tavoitteena ja olennaisena osana on pitää yllä vuoropuhelua ja saada palautetta. Tapahtuma on rajattu noin neljään tuntiin kuukauden pituiselle sprintille, ja kesto skaalautuu alaspäin lyhyemmille sprinteille. Scrummasterin vastuulla on tapahtuman järjestäminen, osallistujien ymmärrys sekä tarkoituksesta että sisällöstä ja aikarajaaminen. (Mts. 12–13.)

Schwaber ja Sutherlandin (2017, 12–13) esimerkin mukaan katselmointi etenee seuraavasti: Katselmoinnissa tuoteomistaja esittää niin kutsutut 'valmiit' ja 'ei valmiit' työn osat. Kehitystiimi käy läpi, mitkä asiat menivät hyvin, mitä ongelmia sprintin aikana kohdattiin ja kuinka ne ratkaistiin. 'Valmis' osuus työstä esitellään kehitystiimin toimesta ja vastaa nouseviin kysymyksiin. Tuoteomistaja raportoi tuotteen kehitysjono-

non nykytilan ja antaa arvioi valmistumisajankohdasta tapahtuneen edistymisen perusteella. Osallistujat miettivät yhdessä, mitä toimenpiteitä voidaan tai kannattaa tehdä alkavan sprintin suunnittelupalaverin pohjustamiseksi. Myös markkinatilannetta, budjettia ja aikataulua tarkastellaan. Koko tapahtuman lopputuloksena on tämänhetkistä toimintaympäristöä ja sen muuttujia vastaava tuotteen kehitysjono. (Mts. 12–13.)

Sprintin viimeinen tapahtuma on katselmoinnin jälkeen tapahtuva **sprintin retrospektiivi**: sen tarkoituksena on antaa Scrum-tiimin tarkastella ja kehittää omia työskentelytapojaan sekä tuottaa suunnitelma kehitysprosessin parantamisesta seuraavan sprintin aikana (Schwaber & Sutherland 2017, 14). Sen kesto on rajattu kolmeen tuntiin kuukauden sprintille, skaalautuen lyhyemmissä sprinteissä alaspäin. Scrum-masterin vastuulla on palaverin tapahtuminen, osallistujien ymmärrys sen tarkoituksesta ja aikarajaaminen. Scrummaster myös itse osallistuu tapahtumaan ikään kuin Scrum-prosessin omistajana. (Mts. 14.)

Schwaber ja Sutherlandin (2017, 14) mielestä retrospektiivissä sprintin toimintaa tarkastelemalla pyritään löytämään parannuksia toimintatapoihin. Palaverissa analysoimalla työkaluja, prosessia, yhteistyötä ja ihmisten välistä vuorovaikutusta yritetään tunnistaa ne asiat, jotka sujuivat hyvin ja jotka eivät. Näiden perusteella luodaan suunnitelma työskentelytapojen kehittämiseksi, jota voidaan toteuttaa seuraavassa sprintissä. Scrummaster tarvittaessa valmentaa ja kannustaa Scrum-tiimiä Scrum-viitekehikseen liittyvissä asioissa. (Mts. 14.)

3.4 Säännöt ja arvot

Schwaber ja Sutherland (2017, 4–5) esittävät, että Scrum koostuu kolmesta peruspiilarista: läpinäkyvyys, sopeutuminen ja tarkastelu. Prosessin tulee olla näkyvä lopputuloksesta vastuussa oleville, sillä heillä tulee olla yhteinen jaettu näkemys ja ymmärrys siitä, mitä toimintaympäristössä tapahtuu. Tällä viitataan erityisesti 'valmiin' määritelmään ja jaettuun yhteiseen kieleen prosessissa työtä tekevien ja sen lopputulosta tarkastelevien henkilöiden osalta. (Mts. 5.)

Edistystä ja Scrumin tuotoksia tulee tarkastella säännöllisesti ei-toivotun vaihtelun havaitsemiseksi. Tarkastelua ei kuitenkaan tulisi tehdä liian usein, ettei siitä tule työkentelyn estettä. (Schwaber & Sutherland 2017, 5.) Mikäli yksi tai useampi prosessin osa poikkeaa liikaa hyväksytyistä raja-arvoista, ja johtaisi ei-toivottuun lopputulokseen, tulee prosessin tai sen materiaaleja muokata mahdollisimman nopeasti poikkeavuuden minimoimiseksi. Scrumin neljä virallista sprintin sisällä tapahtuvaa tapahtumaa on edellä mainituista syistä tarkoitettu sopeutuvuuden ja tarkastelun säännölliseksi ylläpitämiseksi. (Mts. 5.)

Scrum-viitekehyksen käyttämisen onnistumiseksi, tulee Scrum-tiimin noudattaa Scrumin arvoja ja toimia niiden mukaan. Scrum määrittää viisi arvoa, joita ovat

- sitoutuminen
- rohkeus
- keskittyminen
- avoimuus
- kunnioittaminen.

Scrum-tiimi oppii ymmärtämään näiden arvojen merkityksiä ja soveltamaan niitä käyttäessään viitekehyksen eri osa-alueita. Esimerkiksi yksilöiden täytyy sitoutua tiimin tavoitteiden saavuttamiseksi, ja Scrum-tiimin jäsenillä tulee olla rohkeutta työkentällä vaikeiden haasteiden parissa. Jokaisen tulee keskittyä sprintin aikana työkentelyyn ja sprintin tavoitteisiin. Työstä ja siihen liittyvistä haasteista keskustellaan avoimesti, ja Scrum-tiimi kunnioittaa toinen toisiaan ammattitaitoisina yksilöinä. (Schwaber & Sutherland 2017, 5.)

Ketterän ohjelmistokehityksen julistuksessa painotetaan samankaltaisia arvoja kuin Scrumissa (Beck, Beedle, Van Bennekum, Cockburn, Cunningham, Fowler, Grenning, Highsmith, Hunt, Jeffries, Kern, Marick, Martin, Mellor, Schwaber, Sutherland & Thomas, 2001). Ketterän kehityksen pääperiaatteiksi määritellään mm. asiakkaan tyytyväisyys toimittamalla ohjelmistotuotetta aikaisin ja säännöllisesti. Toimiva tuote mainitaan pääasiallisena edistyksen mittarina. Yksilöjen ja vuorovaikutusten arvoa korostetaan prosessien ja työkalujen ylitse. Myös muutosvalmiuden tulisi olla tärkeämpää kuin suunnitelman seuraamisen. (Mt.)

3.5 Ketterän ohjelmistokehityksen historiaa

Scrumia viitekehyksenä on käytetty 1990-luvun alkupuolelta asti (Schwaber & Sutherland 2017, 3). Ensimmäinen kirjallinen maininta Scrumista tuotehallinnan kontekstissa löytyy Harvard Business Review -lehden vuonna 1986 tammikuun numerossa julkaistusta artikkelista, jossa artikkelin kirjoittaneet Hirotaka Takeuchi ja Ikujiro Nonaka vertaavat tuotekehitystä Rugby-urheilulajiin. (Ks. Takeuchi & Nonaka 1986.)

Artikkelissaan Hirotaka ja Takeuchi (1986) väittävät, että perinteiset tuotekehityksen toimintatavat eivät enää riitä onnistumiseen markkinoilla. Ratkaisuksi he esittävät holistista lähestymistapaa, jossa vaiheittaisen toimintatavan sijaan tuotekehityksessä edettäisiin limittäisesti. Vaiheittaisella toimintatavalla tarkoitetaan kehitystyötä, jossa jokaisen osa-alueen spesialisti, esim. insinöörit tekevät oman osuutensa kehityksestä ja siirtävät sen jälkeen tuotteen seuraavalle osa-alueelle tai yrityksen osastolle, kuten markkinointiin. Limittäisessä toimintatavassa jokaisen tuotteen osa-alueen asiantuntijat työskentelevät samanaikaisesti tuotteen kehityksessä, jolloin eri vaiheet limittyvät päällekkäin, ja siten kestävät pidemmälle tuotekehityksen elinkaarissa. Hirotakan ja Takeuchin mukaan jälkimmäinen lähestymistapa on elintärkeä uusien tuotteiden tuottamiseen joustavasti ja nopeasti. Kuten rugbyssä, heidän esittämässä toimintamallissa joukkue yrittää päästä vertauskuvallisesti eteenpäin yksikönä, syöttäen palloa edestakaisin eri specialistien välillä. (Mt.)

Clinton (2010, 6–7) esittää, että peliteollisuuden syntyminen alkupuolella 1970-luvulla pelilaitteistojen kehittyessä yksittäinen henkilö ei enää riittänyt pelin kehittämiseen, vaan julkaisemiseen tarvittiin useampaa spesialistia. Pienen tiimin kehitystyö nosti niin kuluja kuin riskejäkin, mikäli tuote ei menestynyt markkinoilla. Ratkaisuna tähän ongelmaan monet yritykset ottivat käyttöön Waterfall-mallin tyyliä metodologisia ratkaisuja riskien hillitsemiseksi. Waterfall eli vesiputoukselle keskeistä on vaiheittainen eteneminen tuotekehityksen vaiheissa, ja ohjelmistoprojektin suuri koko. (Mts. 6–7.) Malli näyttää olevan hyvin samankaltainen Takeuchin ja Nonakan esittämä vanhentunut, vaiheittainen tuotekehityksen tuotantotapa.

Vesiputousmallien vastustus alkoi kasvaa 1980-luvulla, sillä jopa suuret puolustus- ja informaatioteknologiset projektit epäonnistuivat kiihtyvällä tahdilla. Useita erilaisia ns. iteraatiomalleja alkoi ilmestyä niin kirjallisuudessa kuin artikkeleissakin. (Clinton 2017, 13.) Iteraatiolla tarkoitetaan useampaan otteeseen tapahtuvaa toistumista. Iteraatiomalleissa jokaisessa toistossa on pieni osa jokaista tuotekehityksen osa-aluetta (mts. 13). On huomionarvoista, että Takeuchin ja Nonakan 1986 esittämä limitäinen malli jakaa paljon yhtäläisyyksiä iteratiivisen mallin idean ja ominaisuuksien kanssa.

Clintonin (2017) mukaan näitä uusia iteratiivisia ja inkrementaalaisia – ns. vähittäin kasvavia tai additiivisia – malleja kutsuttiin kollektiivisesti nimellä *lightweight methods* eli kevyiksi menetelmiksi. Vuonna 2001 asiantuntijoista koostuva ryhmä (ks. Beck, ym. 2001) kokoontui yhteen keskustelemaan aiheesta ja julkaisi *Manifesto for Agile Software Development* – nimisen dokumentin, jossa kevyitä menetelmiä kutsuttiin nimellä *Agile methodologies* eli *ketteriksi* menetelmiksi. Manifesti tiivistä yhteisen tärkeimmät arvot ja periaatteet, joihin kevyet mallit pohjautuivat. Manifesti loi pohjan ja mahdollisti siten yhteneväisen filosofian ja periaatteet *ketterille viitekehyksille*, kuten *Scrum*, *Lean* ja *XP*. (Clinton 2017, 13–14; Beck, ym. 2001.)

Manifestin julkaisun jälkeen monilla *ketterillä* menetelmillä on ollut aikaa kehittyä ja vastata muuttuvan maailman – ja siinä tehokkaasti toimivan tuotekehityksen – haasteisiin. Kniberg ja Skarin (2010) mukaan esimerkkeinä *ketterille viitekehyksille* on mm. *Kanban* ja *Scrum*-menetelmän yhdistelmä *ScrumBan*, joka yhdistelee periaatteita ja menetelmiä molemmista. *Scrum* ja *Kanban* ovat molemmat *prosessityökaluja*: prosessi määrittelee, kuinka työskennellään, ja työkalu on mikä tahansa asia, jota käytetään tietyn lopputuloksen tavoittelemiseksi. Knibergin ja Skarinin mielestä työkaluja tulisi ottaa käyttöön ja hyödyntää tarpeen mukaan, sen sijaan, että sitoisi itsensä yhteen tiettyyn prosessiin tai työkaluun. Prosessin tai työkalun kustomointi omiin tarpeisiin on hyödyllistä ja sallittua, niin kauan kuin sillä voidaan päästä haluttuun lopputulokseen. (Mts. 7–10.)

Muita Scrum ja Kanban yhdistelmämallin esimerkin kaltaisia löytyy myös 2010-luvulta. Esimerkiksi Singer (2019) esittelee Basecamp-yrityksen kehittämää omaa mallia, jota on kehitetty yrityksen sisäisesti yli 15 vuoden ajan. Singer jakaa mallin karkeasti kolmeen osaan tai vaiheeseen: muotoiluun, vedonlyöntiin ja rakentamiseen. Muotoilulla tarkoitetaan projekteissa tehtävää esi- ja selvitystyötä, jonka perusteella projektia esitellään vedonlyöntiosassa. Vedonlyönnissä valitaan esitellyistä projekteista potentiaalisimmat ja päätetään, kuinka projektin suhteen tehdään seuraavan kuuden viikon aikana. Rakennusvaiheessa määritellään tiimeihin kohdistuvat odotukset ja käytänteet. Näitä hyödyntäen tiimit löytävät, mitä kehitystyötä täytyy tehdä, kuinka ohjelmointi ja muotoilu voidaan yhdistää, ja kuinka hallita mitä tiedetään tai ei tiedetä. (Mt.)

Singerin (2019) mukaan mallissa painotetaan kuuden viikon mittaisia syklejä, esityötä ja työn muotoilua ennen sen luovuttamista tiimeille ja halukkuuden määrää. Halukkuudella tarkoitetaan sitä, että kuinka paljon aikaa projektiin ollaan halukkaita käyttämään tai kuinka arvokkaana idea nähdään. Tiimeille annetaan täysi autonominen vastuu ja vapaus toteuttaa projekti esityössä määritellyissä rajoissa. Riskienhallintaa toteutetaan mm. yhdistämällä muotoilu ja ohjelmointi aikaisessa vaiheessa ja arvioimalla potentiaalisia riskejä ennen projektin tuomista vedonlyöntiin. (Mt.)

Kaikki yritysten kehittämät sisäiset tuotekehitysviitekehyykset eivät kuitenkaan ole olleet aivan ongelmattomia. Leen (2020) katsaus maailman suosituimman musiikkistriimauspalvelu Spotifyn kehittämään sisäiseen ketterään viitekehyykseen tuo esiin sen, miksi kyseinen malli ei lopulta toiminut räjähdysmäisesti kasvavan yrityksen tarpeisiin. Spotifyn mallissa tiimejä kutsuttiin joukkueiksi ja ryhmää joukkueita kutsuttiin puolestaan heimoksi. Jokaisen joukkueen tuoteomistaja raportoi ja oli vastuussa heimojohtajalle. Ohjelmistoinisöörit tekivät poikkeuksen muihin tiimirooleihin siinä, että he muodostivat tiettyjen erityisosa-alueiden, kuten esimerkiksi palvelinrajapintojen osaston. Näiden lisäksi käytettiin myös kiltoja, osastonjohtajia ja misiona. Joukkueista, heimoista, heimojohtajasta ja osastosta muodostuvaa mallia kutsuttiin matriisimalliksi. (Lee 2020.)

Lee tunnistaa neljä kriittistä virhettä, miksi malli epäonnistui: matriisimalli ratkaisi väärän ongelman, se oli liian keskittynyt tiimiautonomiaan, heimojen rajat ylittävä yhteistyöosaaminen oli otettu itsestäänselvyytenä ja matriisimallin sanasto aiheutti kommunikaatiohaasteita. Lee esittää, että mallissa on loppujen lopuksi hyvin paljon yhteneväisyyksiä mm. Scrumiin, mutta keksityillä, uusilla synonyymeillä Scrumin käyttämälle sanastolle. (Mt.)

3.6 Tuoteomistaja

Clinton (2010, 51) määrittelee tuoteomistajan tehtävään kuuluvan mm. tuotteen vision muodostaminen, vision kommunikoiminen ja kehitettävien toiminnallisuuden asettaminen tärkeysjärjestykseen. Tuoteomistajan vastuita ovat

- tuotteen yhteisen vision muodostaminen asiakkaiden ja kehittäjien välille
- asiakkaiden ja loppukäyttäjien edustaminen
- tuotteeseen sijoitetun pääoman tuotto ja kannattavuus
- julkaisujen ja julkaisuajankohtien suunnittelu
- sprinttien suunnittelu- ja katselmointipalaverien tukeminen
- toiminnallisuudet ja niiden kehittämisjärjestys.

Suurimmat tuoteomistajan sitoumukset sprintin aikana ovat tuotteen kehitysjonon päivittäminen ja sen tärkeysjärjestykseen asettaminen. Tuoteomistaja osallistuu sprintin suunnittelupalaveriin ja sprintin katselmointiin, ja hyväksyy tai hylkää sprintin tulokset. (Mts. 51.)

Visio

Tuoteomistaja toimii tuotteen yhteisen ja jaetun vision kulminoitumispisteenä tiimissä. Jaettu visio on tärkeää tuotteen onnistumisen kannalta, sillä suurissa tiimeissä eri osa-alueiden kehittäjien visio tuotteesta voi alkaa poiketa toisistaan. Lopputuloksena tuote ei välttämättä toimi eheänä kokonaisuutena tai kokemuksena loppukäyttäjän näkökulmasta. (Clinton 2010, 52.) Tuoteomistajan vastuulla on nähdä tuote asiakkaan tai loppukäyttäjän näkökulmasta ja ymmärtää heidän toiveensa ja tarpeensa tuotteen suhteen. (Mts. 52–53.)

Julkaisut ja tuotantovaiheet

Tuoteomistaja käsittelee ja hallitsee tuotteen julkaisuja, julkaisusuunnitelmia ja julkaisupäivämääriä (Clinton 2010, 53). Julkaisuilla tarkoitetaan usean perättäisen sprintin toteutuneesta sisällöstä muodostuvasta kokonaisuudesta. Esimerkiksi videopeleissä tämä aikajakso on tyypillisesti kahdesta neljään kuukautta. Julkaisut voidaan rinnastaa perinteisempien projektien niin kutsuttuihin virstanpylväisiin tai välitavoitteisiin. (Clinton 2010, 43.) Julkaisut asettavat pidemmän aikavälin tavoitteita tiimille ja sidosryhmille, ja siten vaativat tuotteen toiminnallisuuksilta tarkempaa vianmäärittystä ja laadunhallintaa. Tämä vähentää epätietoisuutta jäljellä olevasta työstä tuotteen julkaisemiseksi. Jokaiselle sprintille asetetut tavoitteet johdetaan julkaisusuunnitelmasta. (Mts. 44.)

Julkaisut aloitetaan ja suunnitellaan julkaisusuunnittelupalaverissa, joihin osallistuu tuoteomistaja, sidosryhmät, Scrum-tiimin jäsenet ja muut tarvittavat erityisasiantuntijat. Palaverissa suunnitellaan tärkeimmät julkaisutavoitteet, mahdolliset sprinttitaavoitteet, julkaisusuunnitelma ja julkaisupäivämäärä. Julkaisu edistyy sprinttien edistymisen mukaan. (Clinton 2010, 117–118.) Julkaisusuunnitelmia päivitetään sprinttien tavoitteiden tai kehityksen edistymisen perusteella projektin aikana (ks. Clinton 2010, 53). Palaverissa tarkastellaan edellisen julkaisun aikana tapahtunutta kehitystä ja tuotteen kehitysjonoa, joiden perusteella suurimmat ja tärkeimmät julkaisun tavoitteet määritellään. Nämä tavoitteet edustavat haastetta koko Scrum-tiimille ja auttavat ominaisuuksien kehitysjärjestyksessä ja vision selkeyttämisessä niiden toteuttamiseksi. (Clinton 2010, 118.) Ehdotetut sprinttitavoitteet ovat arvioita työn edistymisestä, eivät sitoumuksia (mts. 120).

Erityisesti videopelituotteissa kehittämisen jakaminen useampaan vaiheeseen voi olla tarpeellista. Yksityiskohtainen konseptointivaihe on tarpeen projektin alkuvaiheessa ennen kuin julkaisijat ovat valmiita sitoutumaan projektiin. Tarinakeskeiset videopelit tarvitsevat paljon sisältöä, joita tuotetaan pelimekaniikkojen avulla. Pelimekaniikat taas kehitetään usein esituotantovaiheessa. Suurissa peliprojekteissa on yksi julkaisupäivämäärä useamman vuoden tuotannon päätteeksi, josta ei voi joustaa. Yleensä raskas tuotteen soveltuvuustestaaminen laitteistojen suhteen jätetään projektin loppuvaiheeseen. (Clinton 2010, 130.)

Vaikka ketterissä videopeliprojekteissa kehittämisen eri osa-alueet on jaettu tasaisemmin sen elinkaaren ajalle, voidaan suurin osa projekteista jakaa eri tuotantovaiheisiin, joilla on oma painopisteensä tuotannossa. **Konseptointivaiheessa** ideoita kehitetään, prototyypataan tai niistä luovutaan nopealla tahdilla. Vaihe on täysin iteratiivinen ja tapahtuu ennen esituotantoa. Konseptointivaihe on monesti aikarajattu yhden tai useamman konseptisuunnitelman tuottamiseksi julkaisijalle tai lisensoijalle. **Esituotantovaiheessa** tutkitaan, mikä konseptissa on mielekästä ja hauskaa, ja kuinka niitä voidaan ilmentää tai tukea pelimekaanisesti. Tuotantotason peli-asetteja eli erilaista sisältöä tuotteeseen kehitetään tämän vaiheen aikana. Esituotantovaiheelle on tyypillistä iteratiivisuus ja inkrementaalisuus. (Mts. 131–132.)

Tuotantovaiheessa keskitytään sisällön tuottamiseen esituotantovaiheessa valittujen pelimekaniikkojen ja prosessien avulla. Ydinmekaniikkoja ei muuteta merkittävästi, sillä suurin osa sisällöstä tulee perustumaan niihin, jolloin niiden muuttaminen tulee hyvin kalliiksi ja paljon työtä voi valua hukkaan. Täten esituotantovaiheessa ydinmekaniikkojen kehittäminen ja tarkkaan valitseminen on ratkaisevan tärkeää. **Jälkituotantovaiheessa** parannellaan tuotetta inkrementaalisesti ehostamalla olemassa olevia ominaisuuksia, optimisaatiolla tai korjaamalla ongelmia. Laitteistoihin liittyvät soveltuvuustestit ajetaan, mikäli niitä ei ole voinut tehdä aikaisemmissa vaiheissa. Tuotannon vaiheet eivät kuitenkaan ole täysin aikarajattuja tiettyyn kohtaan projektin edistymistä, vaan jatkuvat projektin elinkaaren ajan, mutta vähemmällä panostuksella vaiheelle ominaisen pääpainotuksen sijaan. (Mts. 131–132.)

Asiakkaat ja sidosryhmät

Scrum-tiimin suhteet asiakkaisiin ja sidosryhmiin ovat tärkeitä, sillä useat tuotteen kehitysjonon kohdat ja niiden tärkeysjärjestys muodostuvat heidän toiveidensa perusteella (Clinton 2010, 54). Tuoteomistaja voidaan käsittää ns. pääasiakkaana, joka toimii Scrum-tiimissä asiakkaiden ja sidosryhmien edustajana, ja samalla päättää kehitysjonon kohtien tärkeysjärjestyksestä. Sidosryhmillä tarkoitetaan henkilöitä Scrum-tiimin ulkopuolella, joilla on osuus tuotteen kehittämisessä. Tällaisia ovat esimerkiksi tuottajajulkaisija, markkinointi ja muut mahdolliset organisaation eri osa-alueiden päälliköt tai edustajat. Sidosryhmäläiset voivat esitellä ominaisuusehdotuksia tuotteen kehitysjonoon. (Mts. 54–55.)

Ketterä suunnittelu

Ketterän suunnittelun tarkoituksena on jakaa suunnittelun työmäärä koko projektin elinkaaren ajaksi. Ketterissä tiimeissä suunnittelua toteutetaan suhteessa enemmän verrattuna perinteisiin tiimeihin, mutta se ei tapahdu pelkästään projektin alkuvaiheessa. Eniten arvoa asiakkaalle tuottavilla ominaisuuksilla on prioriteetti, joten ne kehitetään mahdollisimman aikaisin. Ketterällä suunnittelulla pyritään minimoimaan teknistä velkaa tai kehitysvelkaa käyttämällä sprinttiä: kokonaista kehityssykliä, jossa viänmääritys, optimoiminen ja hienosäätö tapahtuvat saman syklin aikana kehityksen ohessa. Myös tehtyjä suunnitelmia hienosäädetään projektin edetessä. Jatkuva yhteydenpito sidosryhmän ja kehitystiimin välillä mahdollistaa selkeiden tavoitteiden määrittämisen. (Clinton 2010, 107–108.)

Projektien onnistuminen on kiinni asiakkaiden, käyttäjien, kehittäjien ja liiketoiminnan tai organisaation asiantuntijoiden hallussa olevasta informaatiosta. Nämä henkilöt edustavat karkeasti jaoteltuna kahta näkökulmaa: tekniikkaa tai liiketoimintaa painottavaa. Mikäli toinen näistä näkökulmista on hallitsevassa asemassa projektin informaation ja päätösten suhteen, projekti epäonnistuu. Ongelmaa voidaan ehkäistä tekemällä päätöksiä ja keräämällä tietoa aikaisin ja usein. Tähän käyttötarkoitukseen voidaan käyttää käyttäjäkertomuksia. (Cohn 2004, 3–4.) Käyttäjäkertomusten etuna on ymmärrettävyys niin kehittäjien kuin liiketoiminnan näkökulmasta (ks. Cohn 2004, 104).

Käyttäjäkertomus on lyhyt kirjoitettu kuvaus kehitettävästä asiasta, työkalusta tai ominaisuudesta, jolla on selkeää arvoa sen käyttäjälle (Clinton 2010, 87–88). Käyttäjäkertomuksilla pyritään havainnollistamaan ominaisuuksien arvoa ja herättämään keskustelua aiheesta. Käyttäjäkertomuksia voidaan käyttää edustamaan käyttäjän vaatimuksia käyttäjän näkökulmasta tuotteen tai sen osa-alueen suhteen. Ominaisuuksien yksityiskohtia ei kuvailla yksityiskohtaisesti, vaan tarkoitus on keskustella yksityiskohdista. (Clinton 2010, 87–88; Cohn 2004, 4.) Käyttäjäkertomus sisältää testejä, joiden tarkoituksena on välittää ja dokumentoida informaatiota, jonka perusteella käyttäjäkertomus voidaan todeta valmiiksi (Cohn 2004, 4).

Käyttäjäkertomuksia on mahdollista tehdä mallin perusteella. Malliin kuuluu käyttäjän rooli, käyttäjän tavoite ja käytön syy (ks. kuvio 2). Käytön syy voidaan jättää kirjoittamatta, mikäli syytä pidetään itsestäänselvyytenä. Käyttäjäkertomuksien tulee olla tarvittavan pieniä kokonaisuuksia, jotta ne on mahdollista toteuttaa sprintin aikana. (Clinton 2010, 88.) Jokaista käyttäjäkertomusta ei kuitenkaan kannata rakentaa sprintin aikana kehitettävän kokoiseksi, sillä tuoteomistaja ei pysty hallitsemaan suurta määrää niitä kerralla. Ominaisuuden tärkeys määrittää sen, kuinka tarkkaan käyttäjäkertomuksen yksityiskohdat on merkitty. Mitä tärkeämpi käyttäjäkertomus tai ominaisuus projektissa on, sitä aiemmin sitä aloitetaan työstämään. Tärkeämpiä käyttäjäkertomuksia voidaan myös **pilkkoa** pienemmiksi kokonaisuuksiksi. Pilkkominen voi auttaa tarkentamaan yksityiskohtia ja käyttäjäkertomusten hallintaa. (Mts. 90.) Käyttäjäkertomukset, jotka ovat liian suuria kokonaisuuksia sprintin aikana toteutettaviksi, voidaan erotella muista kutsumalla niitä **eepoksi** (epic) (Clinton 2010, 89; Cohn 2004, 6).

This Issue is **blocked** by Pelihahmon hyppääminen

This Issue is **high priority** in the Mario Revised (example backlog) Workspace

Open **Blocked**

Pelaajahamona voin hyppäämällä aktivoida kysymysmerkkilaatikoita, jotta voin kerätä pisteitä tai poweruppeja #9

Conditions of Satisfaction // Requirements

- Kysymysmerkkilaatikoista saa pisteitä tai sieniä, kun pelaajahahmo osuu alareunaan
- Pisteitä saa jokaisesta osumasta laatikkoon
- Pisteitä on ennaltamäärätty määrä
- Sienet irtoavat laatikosta ja liikkuvat nopeasti
- Sienet kimpoilevat esteistä
- Sienet kasvattavat pelaajan elämää yhdellä
- Sienet kasvattavat pelaajahahmon kokoa pienestä isoksi

Pelaajahamona voin hyppäämällä aktivoida kysymysmerkkilaatikoita, jotta voin kerätä pisteitä tai poweruppeja is **blocked**

This Issue is blocked by

Pelihahmon hyppääminen

Hide Dependency List

Pipelines

- Mario Revised (example backlog)
- Product Backlog **High priority**

Labels

- Story

Assignees

Milestone

No Milestone

Estimate

No estimate yet

Epics

Not inside an Epic

Kuvio 2. Käyttäjäkertomus Zenhub-sovelluksessa.

Käyttäjäkertomuksia voidaan selventää hyödyntämällä **tyytyväisyyskriteerejä** pilkkomisen sijaan. Tyytyväisyyskriteerit voivat olla esimerkiksi lauseen muotoon kirjoitettuja, testattavia yksityiskohtia. (Clinton 2010, 90.) Samaa käytäntöä käytetään hyväksymiskriteereissä ja hyväksymistesteissä. (vrt. Cohn 2004, 12, 67.) Käyttäjäkertomuksia usein havainnollistetaan kirjoittamalla ne viesti- tai muistilapuille. Muistilappujen koko rajoittaa liiallista yksityiskohtaisuutta käyttäjäkertomuksen suhteen, ja niitä on helppo järjestellä ja muokata tarpeen mukaan. Käyttäjäkertomuksen laatua voidaan arvioida **INVEST**-mallin mukaan. (Clinton 2010, 92.) Käyttäjäkertomuksen tulisi siten olla

- Itsenäinen (Independent)
- Neuvoteltava (Negotiable)
- Arvokas (Valuable)
- Arvioitava (Estimable)
- Pieni (Sized appropriately)
- Testattava (Testable).

INVEST-mallin mukainen itsenäinen käyttäjäkertomus ei ole riippuvainen toisen käyttäjäkertomuksen kehittämisestä. Yksityiskohdista ja niiden toteuttamisesta täytyy myös pystyä neuvottelemaan. Käyttäjäkertomuksesta tulee käydä ilmi sen lisäämä arvo helposti ymmärrettävällä tavalla. Laadukas käyttäjäkertomus on myös helppo arvioida työmäärältään ja tarpeeksi pieni, jotta se voidaan kehittää sprintin aikana. Sprintin päättyessä käyttäjäkertomuksen valmius on pystyttävä todentamaan erilaisin testein tai asiantuntijan varmistamana. (Clinton 2010, 93–97.)

Käyttäjärooleja tai käyttäjän näkökulmaa voidaan tarkentaa käyttäjäkertomuksen suhteen erottamalla eritasoisia käyttäjiä. Käyttäjäkertomuksen arvoa voidaan havainnollistaa tai analysoida paremmin esimerkiksi ammattipelaajan ja peruspelaajan näkökulmasta, sillä poikkeavuus niiden välillä voi olla suurta. (Clinton 2010, 97–98.) Mikäli käyttäjäkertomuksen työmäärää ei voida arvioida puuttuvan informaation tai muusta syystä aiheutuvan suuren riskin takia, voidaan ratkaisuna käyttää piikki -käyttäjäkertomusmallia. Piikki (spike) on aikarajattu käyttäjäkertomus, jonka tarkoituk-

sena on vähentää toiseen käyttäjäkertomukseen liittyvää epäselvyyttä ja tuottaa tietoa tutkimalla piikin aihetta. Piikki auttaa tuoteomistajaa arvioimaan muiden käyttäjäkertomusten kehittämisen hintaa, kokoa tai teknologisten ratkaisujen valitsemista. Piikin aikarajauksen päättyessä kerrotaan, mitä piikin aikana tutkittavasta asiasta on opittu. (Clinton 2010, 96.)

Tuotteen kehitysjono ja sen operaatiot

Sprintin aikana työstetyt käyttäjäkertomukset täytyy pystyä toteamaan 'valmiiksi' sprintin päättyessä. Siten ennen käyttäjäkertomuksen kehittämisen aloittamista Scrum-tiimillä tulee olla yhteinen ymmärrys siitä, mitä 'valmis' tarkoittaa väärinymmärrysten välttämiseksi. Käytännössä tämä voidaan toteuttaa kirjoittamalla vaatimuksia tai hyviä käytänteitä, jotka jokaisen käyttäjäkertomuksen täytyy läpäistä, jotta se voidaan todeta 'valmiiksi'. Määrittelyt ehkäisevät teknistä velkaa, selventävät tekemätöntä työtä ja parantavat tuotteen laadunhallintaa. Mikäli yksittäinen käyttäjäkertomus tarvitsee tarkempia määrittelyjä, ne voidaan lisätä tyytyväisyyskriteereinä. (Clinton 2010, 99.)

Projektin alussa Scrum-tiimi ja sidosryhmät kokoontuvat luomaan korkea tason käyttäjäkertomuksia tuotteen suhteen. Käyttäjäkertomuksia voidaan kerätä hyödyntäen markkinoinnista ja kohderyhmästä saatua tietoa tai hyödyntäen työpajaa käyttäjäkertomusten keräämiseen. Työpajassa tuoteomistaja kutsuu koolle kaikki henkilöt, joiden on mahdollista osallistua ideoimiseen. Työpajan tavoitteena on kerätä seuraavan julkaisun verran käyttäjäkertomuksia. Tämä edellyttää keskustelua käyttäjäkertomusten tärkeysjärjestyksestä, kehitystiimin kapasiteetista ja toteuttamisen mahdollisuuksista. Mikäli kehittämiseen tarvittavia erityisasiantuntijoita on saatavilla projektissa käyttäjäkertomusten toteuttamiseksi, ne puretaan pienemmiksi ja yksityiskohteisemmiksi. Kehitystiimin ja muiden asiantuntijoiden mielipiteet ovat erittäin tärkeitä tässä vaiheessa ideointia teknisten ratkaisujen tai niistä johtuvien rajoitteiden tunnistamiseksi. (Clinton 2010, 101–102.)

Käyttäjäkertomusten työmäärän arviointiin voidaan käyttää **tarinapisteitä** (story points). Tarinapisteet ovat suhteellisia työmäärän tai sen monimutkaisuuden mittaa-

misen yksikköjä. Mittayksikön suhteellisuus mahdollistaa tarkempien arvioiden tekemisen, sillä käyttäjäkertomuksia voi vertailla suhteessa toisiin käyttäjäkertomuksiin. (Clinton 2010, 114.) Cohnin (2004, 11) esimerkin mukaan, neljän tarinapisteen käyttäjäkertomuksen toteuttamiseen kuluu kaksi kertaa enemmän aikaa kuin kahden tarinapisteen käyttäjäkertomukseen. Yhteen sprinttiin mahtuvien käyttäjäkertomusten määrää voidaan arvioida laskemalla, kuinka monta tarinapistettä yhden sprintin aikana tiimi on toteuttanut (Clinton 2010, 111). Kehitystiimi voi myös arvioida, kuinka monta tarinapistettä he pystyvät toteuttamaan sprintin aikana. (ks. Cohn 2004, 11). Tarinapisteiden toteuttamisen kapasiteettia sprintin aikana tiimin toimesta kutsutaan termillä **vauhti** (velocity) (Clinton 2010, 110; Cohn 2004, 11). Täten pidemmän aikavälin tavoitteiden saavuttamisajankohtaa projektissa voidaan arvioida mittaamalla Scrum-tiimin keskimääräistä vauhtia (Clinton 2010, 111).

Tarinapisteitä voidaan arvioida tarinatyöpajassa. Tarinapisteiden arvioimiseen tulisi hyödyntää käyttäjäkertomukseen liittyvien eri osa-alueiden asiantuntijoiden mielipiteitä paremman kokonaisarvion saavuttamiseksi. Koon arvioimista helpottaa myös triangulaatio, eli käyttäjäkertomuksen arvioiminen suhteessa pienempään ja suurempaan tai osa-alueeltaan samankaltaisiin käyttäjäkertomuksiin. Suuremmat käyttäjäkertomukset ovat hankalampia arvioida verrattuna pieniin, joten niiden pilkkominen pienemmiksi tarkentaa myös arviota. (Clinton 2010, 114.)

Nämä osatekijät yhdistävä tekniikka käyttäjäkertomusten arviointiin on **suunnittelupokeri** (planning poker). Suunnittelupokeria voidaan käyttää tarinatyöpajassa tai julkaisupalaverissa. Tekniikassa osallistujat keskustelevat käyttäjäkertomuksesta, ja sen päätteeksi antavat sille tarinapistearvon samanaikaisesti paljastamalla sitä vastaavan pelikortin tai sitä vastaavan merkin. Eniten keskimääräisestä arviosta poikkeavat henkilöt perustelevat arvionsa. Tämä paljastaa mahdolliset käyttäjäkertomukseen liittyvät epävarmuudet ja olettamukset. Tätä toistetaan, kunnes kaikki osallistujat tuottavat saman tarinapistearvon. Mikäli arvot ovat muutamaa poikkeusta lukuun ottamatta samat, poikkeushenkilöt voivat luovuttaa kantansa suhteen ja hyväksyä vallitsevan arvion. Kaikkia julkaisun käyttäjäkertomuksia ei usein analysoida yhdeltä istumalta suuren työmäärän takia, vaan kaikista tärkeimmät. Jäljelle jääneiden arvioimista voidaan jatkaa tapaamalla kerran sprintissä. (Clinton 2010, 115.)

Projekteilla tulisi olla ennalta määritetty **asteikko** tarinapisteiden arvioimiseksi. Yleinen käytäntö on, että asteikon sisällä valitut pistearvot ovat kahden suuruusluokan sisällä, esimerkiksi 1–100. Pistearvoja tulisi esiintyä enemmän asteikon alapäässä, sillä arvopisteiden 1 ja 2 koko suhteessa toisiinsa on paljon suurempi kuin esimerkiksi arvopisteiden 20 ja 21. **Fibonaccin lukujonon** ensimmäisiä arvoja käytetään usein asteikon alkupäässä ja pyöristämällä loppupäässä, kun asteikko ja pistearvot valitaan. Fibonaccin lukujonossa jokainen luku on kahden edellisen lukujonon luvun yhteenlaskettu summa, kuten 1, 2, 3, 5, 8 ja niin edelleen. Kun käyttäjäkertomuksien tarinapisteitä arvioidaan, täytyy sovittuja asteikon lukuja käyttää, eikä niistä saa poiketa. (Clinton 2010, 116.)

4 Tuoteomistajuus osana ohjelmistotuotantoa ja Ticorporatea

Tässä luvussa tutkitaan aikaa ennen tuoteomistajan sivutoimen alkamista. Luvun 4 tarkoituksena on tutkia, mitä tietoja opiskelijoilla on tuoteomistajuudesta ennen sivutoimen valintaa. Luvussa 4.1 tutkitaan, kuinka tuoteomistajuutta on käsitelty, ennen kuin opiskelijat saapuvat Ticorporateen. Materiaalia verrataan teoriaan, ja analyysin perusteella annetaan myös mahdollisia kehitysehdotuksia. Tuoteomistajien ryhmäteemahaastattelun ohjelmistotuotantoon liittyviä tuloksia verrataan aineistoon. On huomionarvoista, että luvussa 4.1 analysoinnin kohteena oleva luentomateriaali ei kerro mitään siitä, kuinka materiaalia esitetään verbaalisti. Luvussa 4.2 selvitetään, mikä Ticorporate on ja mitä tarpeita ja vaatimuksia se esittää tuoteomistajan sivutoimen järjestämisen suhteen.

4.1 Ohjelmistotuotanto

Ohjelmistotuotanto on tietojenkäsittelyn tutkinto-ohjelman 10 opintopisteen laajuisen opintojakso, joka edeltää Ticorporatea ja antaa valmiudet opiskelijoille Ticorporateen. Sen osaamistavoitteina ovat ohjelmistokäsitteiden, ohjelmistoliiketoiminnan

kokonaisuuden ja asiakasnäkökulman sisäistäminen. Opiskelija kykenee määrittelemään tuotteen kehittämisen vaatimuksia ja käyttäjäkertomuksia yhteistyössä tuotteen omistajan kanssa. (Ohjelmistotuotanto HTT201101-3001 2020, luku Osaamista-voitteet.)

Ohjelmistotuotanto -opintojaksolla käsitellään ketteryyden määritelmää, Scrum-viitekehystä ja sen soveltamista. Kiviaho (2019) käsittelee luentomateriaalissaan Manifesto for Agile Software Development -dokumentin perusperiaatteet ja ketterät arvot yksi kerrallaan, ja tuo esiin havainnollistavia esimerkkejä sen jokaisesta osasta. Scrumin tapahtumat, toiminnot ja artefaktit esitellään yleisellä tasolla. (Mt.)

Tuotteen kehitysjo

Tuotteen kehitysjonosta mainitaan, että se perustuu tuoteomistajan visioon. Tuoteomistaja on vastuussa vision realisoimisesta. Kehitysjonon toteuttamiseen käytetään käyttäjäkertomuksia. Projektin alussa tuotteen kehitysjo määritellään, sen sisältämät kohdat laitetaan tärkeysjärjestykseen ja tärkeimmät kehitysjonon kohdat pisteytetään. (Kiviaho 2019.) Tuotteen kehitysjo mainitaan muodostuvan käyttäjäkertomuksista, ja siihen merkitään kaikki kehittämiseen liittyvät ajatukset. Luentomateriaalissa tärkeimmät tuotteen kehitysjonon toiminnallisuudet käydään läpi, mutta niiden toteutustapaan ei oteta kantaa. (Mt.) Mainintaa tuotteen kehitysjonosta elävänä ja muuttuvana dokumenttina tai läpinäkyvyyden tärkeydestä ei ole. Tuotteen kehitysjo on visuaalisesti havainnollistettu listana. Inkrementtiä ei mainita. Toisaalta on huomionarvoista, että pelkkä luentomateriaali ei kerro mitään siitä, kuinka aiheesta keskustellaan luennon aikana, tai tuleeko dokumentin muuttuva luonne tai muut vastaavat seikat esille verbaalisti.

Visio

Tuotevisio määritellään tavoitteeksi tai unelmaksi, jota tavoitellaan, ja joka on lähtöisin tuoteomistajasta tai asiakkaasta (Kiviaho 2019). Vision tarkoituksena on ohjata tuotekehitystä ja selittää, miksi tuotetta kehitetään. Visio kertoo asiakkaan tai käyttäjän kohderyhmän ja kuinka kohderyhmän ongelma tai tarve voidaan ratkaista onnistuneesti. Huomionarvoista on myös kilpailu, markkinat ja käytössä olevat resurssit, kuten aika ja pääoma. Vision tulisi olla lyhyt, selkeä ja inspiroiva, mutta jättää samalla

tilaa luovuudelle. Hyvässä visiossa ydinominaisuudet käyvät selviksi konkreettisesti, ja sen tulee herättää tunteita, ja olla samaistuttava sidosryhmien kesken. Visio täytyy voida kiteyttää yhteen lauseeseen. (Mt.)

Tuoteomistaja yhteisen jaetun vision kantajana sidosryhmien, kehitystiimin ja asiakkaiden tai loppukäyttäjien suhteen ei käy selväksi Kiviahon esityksestä. Kiviaho tosin painottaa visiota tarkkaan harkittuna, kiteytettynä tuotteen potentiaalisen olemassaolon kokonaisuutena. Visiota tarkastellaan laajemmin, havainnollisemmin ja konkreettisemmin verrattuna Clintonin (2010, 52) määritelmään. Clintonin vision määritelmä on siis abstraktimpi ja Kiviahon määritelmä konkreettisempi.

Visiosta voisi materiaalissa olla mainintaa kaikkien sidosryhmien, asiakkaiden, kehittäjien ja loppukäyttäjien kollektiivisen mielikuvituksen tai potentiaalin ilmentymänä. Siinä missä Scrummasteria voidaan kutsua vertauskuvallisesti ”prosessin vartijaksi”, voidaan tuoteomistajaa siten verrata ”vision vartijaan”. Tuoteomistaja on vastuussa siten vision keräämisestä, yhdistämisestä ja toteuttamisesta, mutta omaa myös viimeisen sanan.

Valmiin määritelmä

Kiviaho (2019) esittää, että definition of done eli ’valmiin määritelmä’ kertoo, mitä ’valmis’ tarkalleen tarkoittaa. Sen tarkoituksena on olla apuna analysoidessa sitä, mitä on potentiaalisesti julkaisukelpoinen tuote. ’Valmiin määritelmä’ on määritelty eri tavalla eri tasoille, kuten käyttäjäkertomuksille, sprinteille ja julkaisuille. Määritelmät muuttuvat ja niitä tarkennetaan sitä mukaa, kun projekti etenee. Käytännön toteutus määritelmän suhteen voi vaihdella. Tuoteomistajan hyväksyntä voi olla yksi testi määrittelemään, milloin tehty työ on potentiaalisesti julkaisukelpoinen. Muita esimerkkejä voi olla esim. versionhallinnassa lähdekoodiin käytetty tagi eli merkintä, dokumentaation päivittäminen tai koodikatselmointi ja siihen liittyvät toimenpiteet. (Mt.)

Kiviahon määritelmällä on yhtäläisyyksiä Schwaberin ja Sutherlandin (2017, 17) ’valmiin määritelmään’. Schwaberin ja Sutherland painottavat inkrementin tarkastelua määritelmän avulla, jotta se voidaan todeta julkaisukelpoiseksi. Scrum-tiimillä tulisi

olla yhteinen ymmärrys, mitä 'valmis' tarkoittaa. Kiviaho ei käytä termiä inkrementti, mutta referoi käytännössä täysin samaan asiaan "potentiaalisesti julkaisukelpoisella" tuotteella. Yhtäläisyyksiä löytyy myös tiimin yhteisen ymmärtämisen tärkeyden suhteen. Toisin kuin Kiviaho, Schwaber ja Sutherland ei puolestaan ota kantaa määritelmän erilaisuudelle ei tasoilla, esim. käyttäjäkertomuksille, sprinteille ja julkaisuille. Kiviaho antaa konkreettisia esimerkkejä määritelmän toteuttamisesta käytännössä. Myös Clinton (2019, 99) mainitsee määritelmän toteuttamisesta käytännössä, että se voidaan toteuttaa kirjoittamalla vaatimuksia tai hyviä käytänteitä. Kiviahon määritelmä on kokonaisuutena kattava.

Käyttäjäkertomukset

Käyttäjäkertomuksien Kiviaho (2019) kuvailee olevan lauseita, joista käy ilmi kuka tekee, mitä tekee ja miksi tekee. Kiviahon käsitys käyttäjäkertomuksesta on hyvin pitkälti sama, kuin Clintonin (2010, 88) määrittelemät käyttäjärooli, tavoite ja syy. Samoin kuin Clinton, Kiviaho määrittelee käyttäjäkertomusten koon epiciksi eli eepokseksi ja normaaliksi eli sprintin aikana toteutettavan kokoiseksi. Eepoksista mainitaan, että ne ovat laajoja kertomuksia, joita usein käytetään projektin alkuvaiheessa, ja että eepoksen pilkotaan pienemmiksi viimeistään suunnittelupäivässä. Suunnittelupäivällä Kiviaho referoi Scrumin suunnittelupalaveriin, mutta eri termillä.

Kiviaho (2019) esittää, että käyttäjäkertomus sisältää monia muitakin yksityiskohtia, kuten toiminnallisen kertomuksen, hyväksymiskriteerit, tekniset ratkaisut tai materiaalit ja kertomuspisteet. Toiminnallinen kertomus on kuvailu käyttäjän toiminnasta järjestelmässä, jota voidaan verrata käyttötapaukseen. Hyväksymiskriteerit ovat tarkempia yksityiskohtia, joiden avulla testaamista voidaan suunnitella. Tekniset ratkaisut ja materiaalit ovat mahdollisia kuvia, linkkejä, arkkitehtuuriviittauksia, käyttöliittymäsuunnittelua tai muuta relevanttia sisältöä käyttäjäkertomuksen toteuttamiseksi. Kertomuspisteillä Kiviaho referoi tarinapisteisiin: kyseessä on vain erilainen käännös englannin kielen termistä Story point. Kiviaho painottaa, että käyttäjäkertomuksen sisältämä dokumentaatiota on ikään kuin pöytäkirja keskustelusta, ja tärkeämpää on käyttäjäkertomuksesta käyty keskustelu. (Ks. mt.)

Kiviahon malli ei merkittävästi eroa Clintonin (2010, 87–90) mallista. Clinton painottaa enemmän käyttäjäkertomuksen tarkoitusta havainnollistamaan ominaisuuksien arvoa käyttäjälle. Clinton, Cohn ja Kiviaho käyttää termiä eepos havainnollistamaan isokokoista tai laajaa käyttäjäkertomusta. Tarinapisteet on mainittu materiaalissa, mutta niiden käyttämiseen ei laajemmin oteta kantaa. Materiaalissa painotetaan nimenomaan sitä, että kaikista tärkeintä käyttäjäkertomuksissa on niistä syntyvä keskustelu. Kiviaho käy läpi käyttäjäkertomuksen päätehtävät, kuinka käyttäjäkertomuksia tehdään ja niiden tarkoituksen osana Scrumia, mutta ei syvenny aiheeseen niitä pidemmälle.

Ryhmähaastattelussa kävi ilmi, että yksi tuoteomistajista koki käyttäjäkertomukset irrallisiksi, eikä ymmärtänyt niiden tarkoitusta. Yhden tuoteomistajan mielestä käyttäjäkertomuksien kirjoittamista ei ohjattu. Eräs haastateltavista koki, ettei saanut käyttäjäkertomuksista hyötyä irti, ja oli ”hukassa sen kanssa”, kunnes niiden käyttö alkoi Ticorporatessa. Kaksi haastateltavaa neljästä koki osuuden sekavana. Selkeitä johtopäätöksiä käyttäjäkertomuksista ryhmähaastattelun perusteella on kuitenkin hankalaa vetää luotettavasti, sillä tieto aiheesta on vähäistä.

Tuoteomistaja

Tuoteomistajan roolista osana Scrumia on mainittu, että tuoteomistajan tulisi nähdä projektin status viikkoraportin tai burndown -kaavion muodossa (Kiviaho 2019). Tuoteomistajan tulee olla läsnä demojen eli inkrementtien esittelyissä sprinttien päätteeksi ja tietoinen sprinttien kehitysjonosta ja valmiin määritelmästä. Kehitystiimin vauhti ja arvio käyttäjäkertomusten valmistumisesta ovat myös olennaista tietoa, jota tuoteomistajalle on tarpeen välittää. Retrospektiiveissä tuoteomistajan läsnäolo olisi suotavaa. Myös testauksesta saatavaa kvantitatiivista tietoa tulisi välittää tuoteomistajalle. (Mt.)

Kiviahon materiaaleissa ei tosin suoraan mainita tarkemmin tuoteomistajan muista vastuista projektiin liittyen, kuten asiakkaiden, sidosryhmien ja loppukäyttäjien äänen edustamisesta. Julkaisuista ja tuotantovaiheista ei myöskään ole mainintaa. Toisaalta ohjelmistotuotannon opintojakso voidaan määritellä konseptointivaiheeksi, ja Ticorporaten puolestaan sisältävän esituotanto, tuotanto- ja jälkituotantovaiheet.

Julkaisuista tai julkaisunhallinnasta ei ole mainintaa. Toisaalta voidaan esittää, että Ticorporaten demotilaisuudet ja Kiviahon esittämät ”demot” tarkoittavat käytännössä julkaisuja, mutta eri termillä. Kiviaho myös painottaa näkökulmaa, että tuoteomistajan tulee olla tietoinen projektin etenemisestä tavalla tai toisella.

Tuoteomistajien ryhmäteemahaastattelussa käsiteltiin ohjelmistotuotannon antamia valmiuksia Scrum-viitekehukseen ja tuoteomistajuuteen. Kokemukset olivat hajanaisia: yhden haastateltavan mielestä opintojakso ei antanut ”minkäänlaisia” valmiuksia. Toinen mainitsi, että tieto oli kovin sekalaista. Kolmannen mielestä opiskelijat saivat taustatietoa viitekehuksesta, sen tarkoituksesta ja toiminnasta käytännössä. Kaikki informantit olivat yhtä mieltä siitä, että opintojakso ei antanut ollenkaan valmiuksia tuoteomistajuuteen, eikä ”termiä edes mainittu”. Kaksi haastateltavista ei osannut yhdistää tuoteomistajuutta Scrum-viitekehukseen. Yhden tuoteomistajan mielestä Scrumin teoriaa oli liikaa ja käytännön harjoitusta liian vähän. Yksi toivoi teorian lisäksi käytännön harjoitteita.

Konseptointiin toivottiin myös enemmän aikaa. Kolme neljästä haastateltavasta koki, että ideointiin olisi voinut kohdentaa enemmän aikaa. Yhden mielestä olisi parempi, mikäli ideoita olisi mahdollista kehittää useampia ennen Ticorporatea. Yksi haastateltavista mainitsi, että visio ei ollut tarpeeksi selkeä Ticorporateen siirryttäessä.

4.2 Ticorporate Demo Lab

Ticorporate yrityssimulaatio on pakollinen osa Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelmaa. Sen tavoitteena on opettaa opiskelijoita toimimaan projektiryhmässä tietyissä kehitystyön rooleissa ja oppia, kuinka tietojärjestelmäprojekteja toteutetaan. Ticorporate jaettiin kahteen osaan vuonna 2017: keväisin järjestettävään Ticorporate Demo Labiin ja vapaavalintaiseen, syksyllä järjestettävään Ticorporate Product Labiin. Demo Labissa opiskelijat tuottavat tiimeissä videopelejä, verkkopalveluja tai verkkotuotteita. (Pölkki 2020.) Demo Labin tuotantoprosessi koostuu neljästä eri jaksosta: konseptoinnista, esituotannosta, tuotannosta ja jälkituotannosta (mt.).

Demo Lab alkaa kolmipäiväisellä konseptointivaiheella, jonka alussa esitellään ohjelmistotuotannon opintojaksolla kehitetyt peli- tai ohjelmistokonseptit opiskelijoille ja henkilökunnalle. Opiskelijat vastaavat kyselyyn, joka kartoittaa heidän henkilökohtaisia mielenkiintonsa kohteita, jotta he voivat alkaa pohtia omaa rooliaan tiimeissä ja sivutoimen valintaa. Henkilökunta valitsee esitellyistä konsepteista sopivan määrän suhteessa opiskelijamäärään. Valittujen konseptien tuoteomistajat esittelevät konseptiaan opiskelijoille, ja opiskelijat voivat neuvotella konseptin sisällöistä ja omasta halusta osallistua konseptin toteuttamiseen. Ryhmättömät opiskelijat antavat näkemysensä dokumentin muodossa konsepteista, joihin heillä oman arvionsa mukaan on mielenkiintoa ja potentiaalista osaamista. Tuoteomistajat voivat myös antaa ehdotuksia projektiin sopivista henkilöistä. Henkilökunta jakaa ryhmättömät opiskelijat konsepteihin parhaan arvionsa mukaan edellä mainittuihin tietoihin perustuen. (Pölkki 2020.)

Konseptointivaiheen päättymisen jälkeen alkaa esituotantovaihe, jossa muodostetut ryhmän henkilöt tutustuvat keskenään, kertovat tarkemmin osaamisestaan ja sopivat roolituksesta. Projektin tuotantomalli vaikuttaa roolien määräytymiseen. Jokaisessa projektissa tulisi kuitenkin olla vähintään yksi Scrummaster, tuoteomistaja, testaaja ja liiketoimintavastaava. Esituotantovaiheeseen käytetään sen verran aikaa, mitä konsepti tarvitsee, mutta mielellään yhden työviikon verran. Esituotannon aikana ryhmät tuottavat High Level Test Planin, projektisuunnitelman ja joko Game Design -dokumentin tai esitutkimuksen. Lisäksi ryhmät pystyttävät tarvittavat työympäristöt, tutustuvat työkaluihin ja niiden käyttämiseen, esimerkiksi projektinhallinnan ja testauksen osalta. (Pölkki 2020.)

Tuotantovaihe alkaa noin kahden viikon jälkeen Ticorporaten ajankohdasta, jolloin projektien toteuttaminen alkaa edetä projektisuunnitelmien mukaisesti. Tuotantovaihe jaetaan tuotantojaksoihin Scrumin mukaisesti. Käytännössä tämä tarkoittaa pääsääntöisesti sprinttejä. Tuotantojaksoilla pääpaino on tuotteen toiminnallisuuksien kehittämiseen tehokkaasti ja suunnitelmallisesti. Tuotantovaiheeseen sisältyy kolme demotilaisuutta, jossa projektin edistymistä esitellään julkisesti, ja jossa projektit saavat kehittämis ehdotuksia ja palautetta (Mt.). Demotilaisuuksia voidaan käytännössä verrata Clintonin (2010, 43) esittämiin julkaisuihin, jotka ovat useamman

perättäisen sprintin sisällöstä muodostuva kokonaisuus ja samalla välitavoite projektin elinkaaren aikana. Tuotantovaiheen aikana on tarkoitus julkaista projekti ryhmän prosessin ja valintojen mukaan vaiheen alussa tai lopussa. Henkilökunnan tarkoituksena on auttaa ja ohjata ryhmiä Demo Labin aikana. (Pölkki 2020.)

Demo Lab päättyy jälkituotantovaiheeseen, jossa projektin toteutus ja parhaat löydettyt käytänteet esitellään projektissa opittujen asioiden lisäksi, jotta tulevat opiskelijat voisivat oppia niistä. Jälkituotannossa projektien jäsenet esittelevät loppudemonsa ja koko projektin elinkaaren aikaisen tuotannon. Projekteista kirjoitetaan loppuraportit, projektit varmuuskopioidaan ja tuotokset toimitetaan henkilökunnalle. Jälkituotantoon kuuluu myös julkaiseminen projektin valitsemassa kanavassa, jonka projektin jäsenet saavat päättää. Jokainen opiskelija luo oman näyteportfolionsa, joka esitellään näyttöpäivänä. Loppuesityksessä kaikki projektit esitellään. (Mt.)

Esituotannon alussa opiskelijat valitsevat päätoimen ja sivutoimen. Opiskelijat voivat valita useampia sivutoimia tai pelkän päätoimen, mikäli he pystyvät todistamaan tehneensä päätoimea laajasti näyteportfoliossaan. Päätoimeksi voi valita tekniikan, sisällöntuotannon tai liiketoiminnan. Sivutoimen vaihtoehtoja ovat tekniikka, sisällöntuotanto, liiketoiminta, Scrummaster, testaaja ja tuoteomistaja. (Mt.)

Jokainen opiskelija tuottaa Demo Labin aikana oman henkilökohtaisen näyteportfolionsa, jonka tarkoituksena on havainnollistaa Demo Labin aikana hankittua osaamista. Näyteportfolioon valitaan laaja ja hyvin osaamista edustava kokoelma opiskelijan näytteistä. Näyteportfolion näytteet valitaan perusportfoliosta, joka koostuu opiskelijan kaikesta Demo Labissa tuottamasta aineistosta, kuten raportit, kirjoitelmat, luonnokset, välivaiheiden tuotokset ja tuotosten lopputulokset. Perusportfolioon suositellaan talletettavan kaikki arvioinnit, palautteet, itsearviot, valokuvat, muistiinpanot tai vastaavat dokumentaatiot. Näyteportfolio on arvioitava kokonaisuus, mutta perusportfoliota ei arvioida. Näyteportfoliossa opiskelija tuo esiin omia näkemyksiään ja vahvuuksiaan omasta osaamisesta. Esimerkiksi sisällöntuotannon työprosessi ja välituotokset on tärkeää tuoda esiin lopputulosten lisäksi. Näyteportfolion toteutustapa on vapaa, mutta useimmiten se rakennetaan verkkoon tavalla tai toisella. (Mt.)

Näyttöpäivässä opiskelijat esittelevät näyteportfolionsa. Näyttöpäivän tarkoituksena on havainnollistaa opiskelijan osaamista ja sen kehittymistä, jotta sitä voidaan arvioida. Arviointi koostuu kolmesta pääaihealueesta: tekniikasta, sisällöntuotannosta ja sekalaisesta. Opiskelija osallistuu yhteen näistä perustuen siihen, mitä on pääosin Demo Labin aikana tehnyt. Portfolion arviointikriteerit ovat tyydyttävä, hyvä ja kiitettävä. Portfolion arvioitavia kohteita ovat

- portfolion sisältö ja toteutustapa
- osaaminen
- kontribuutio projektiin
- itsearviointi ja oma analyysi
- esitys ja lisäarvo portfoliokeskustelulle.

Arviointiraati koostuu relevanteista tutkinto-ohjelman henkilökunnan jäsenistä. Raati antaa arviointipäätöksen näyteportfoliosta kriteerien ja arviointikohteiden perusteella. (Pölkki 2020.)

Työskentely projekteissa tapahtuu maanantain ja perjantain välisenä aikana, ja työmäärä on 30 tuntia viikkoon. Työmäärään sisältyy kaksi kahvitaukoa, ja opiskelijat voivat hyödyntää liukuvaa työaikaa. Jokaisesta työskentelykuukaudesta opiskelijat saavat yhden lomapäivän. (Mt.)

5 Tuoteomistajan sivutoimi Demo Labissa

Tuoteomistajan sivutoimen kokonaisuuden rakentaminen aloitettiin Ticorporate Demo Lab 7 – toteutuksen alkaessa tammikuussa 2020. Tästä syystä aiempaa oppimateriaalia sivutoimesta ei ollut, joten materiaalia kehitettiin toteutuksen aikana. Tässä luvussa kyseisiä tuotettuja materiaaleja ja rakenteita esitellään ja verrataan Scrumin teoriaan. Ryhmäteemahaastattelun aineistoa käytetään analysoinnin ja johtopäätösten tukena. Lisäksi Demo Labin aikana tehtyjä havaintoja käytetään analyysin ja tulosten tukena. Analyysissa otetaan huomioon myös luvun 4 johtopäätökset.

Sivutoimen rakenne

Sivutoimeen pyrittiin saamaan laajasti erilaista materiaalia. Tuoteomistaja -sivutoimen rakenne Ticorporate 7 -toteutuksen aikana muodostui seuraavanlaiseksi

- Luento 1 (tuoteomistajuus, ketteryys, visio ja työkalut)
- Luento 2 (käyttäjäkertomukset, tarinapisteet ja tarinapisteiden arviointi)
- 10 viikkotapaamiskertaa
- asiantuntijahaastattelu (Product Leader)
- käyttäjäkertomus- ja Zenhub työpajat
- manuaali (Product Owner Survival Guide Ticorporate Edition)
- henkilökohtaista ohjaamista projektille tai tuoteomistajopiskelijoille pyydettyessä.

Ticorporaten portfolioa ei otettu huomioon rakennetta suunniteltaessa. Sivutoimea arvioitiin asteikolla hyväksytty tai hylätty.

5.1 Viikkotapaamiset

Sivutoimeen valittiin 1-2 viikon välein tapahtuvat viikkoluennot tai tapaamiset, joissa käsiteltiin tuoteomistajuuteen liittyviä aiheita tai koettuja haasteita. Tapaamiset valittiin siksi, että monet muutkin sivutoimet, kuten Scrummasterit, hyödyntävät viikkotapaamisia kokoontumalla yhden tai kahden viikon välein. Menetelmä on siis todettu ja testattu toimivaksi jo useamman vuoden ajan. Usein viikkotapaamisiin liittyy jokin ennalta määrätty aihe, josta keskustellaan. Olennainen viikkotapaamisien etu on myös yhteisön tuki haastavissa tilanteissa ja vertaisoppiminen kuuntelemalla toisten haasteita. Viikkotapaamisissa tuoteomistajaopiskelijat pääsivät siten vertaisoppiin ja antamaan vertaistukea toisilleen ideoiden tai ratkaisuehdotusten muodossa, ja muodostamaan pientä yhteisöä. Demo Labin alkuvaiheessa tapaamisia oli kerran viikossa, jotta opiskelijoiden kokemiin haasteisiin projekteissa pystyttiin vastaamaan nopeasti. Tuotannon puolivälissä tapaamisia järjestettiin kahden viikon välein. Tarvetta lyhyemmälle syklille ei ollut, sillä opiskelijat alkoivat sisäistää tuoteomistajuutta ja toiminta oli rutiininomaista.

Ryhmähaastattelusta kävi ilmi, että kaikki neljä tuoteomistajaa kokivat viikkotapaamiset hyödyllisiksi. Näkemys muiden tuoteomistajien kokemuksiin ja työskentelytapoihin koettiin opettavaiseksi kahden mielestä. Erityisesti omien ongelmien purkamisen ja vertaistuen koettiin helpottavan stressiä. Kokemus siitä, että ei ollut ainoa, jolla oli ongelmia jossakin aihealueessa, koettiin helpottavana. Omaa toimintaa pystyi vertaamaan toisten tuoteomistajien toimintaan ja analysoimaan omaa kehitystään. Viikoittainen tapaamisfrekvenssi koettiin sopivaksi.

5.2 Luennot

Demo Labin aikana järjestettiin kaksi luentoa, joissa tuoteomistajuutta kokonaisuutena esiteltiin ja kuinka tuoteomistajan rooliin liittyviä operaatioita voidaan tehdä käytännössä.

Luento 1

Ensimmäisellä luennolla Product Ownership – Introduction to Demo Lab käsiteltiin sitä, mitä ketteryys tarkoittaa, miksi Demo Labissa käytetään Scrumia ja mitä vastuita tuoteomistajalla sekä Scrum-viitekehyksessä että Demo Labin projekteissa (Product Ownership Ticorporate 7 Demolab edition 2020). Ketteryys määritellään luentomateriaalissa hyvin yksinkertaisella tasolla mukailten Dave Thomasin (2015) määritelmää: selvitä lähtötilanteesi, ota askel kohti tavoitettasi, muuta toimintatapaasi juuri oppimasi asioiden perusteella ja toista prosessi. Tämän lisäksi, jos kahdesta tai useammasta arviolta yhtä paljon arvoa tuottavasta vaihtoehdosta täytyy valita, valitse se reitti, joka tekee muuttumisen tulevaisuudessa helpommaksi. Lisäksi Manifesto for Agile Software Development -dokumentin ensimmäiseen osaan viitataan. (Product Ownership Ticorporate 7 Demolab edition 2020)

Luentomateriaalissa käsitellään myös sitä, miksi Scrumia käytetään ja mitä ketteryys tarkoittaa Scrumin kontekstissa. Luentomateriaali esittää argumentin Scrumin käyttämisestä: Scrumin perusteita opetellaan, jotta tiedetään, milloin se toimii, milloin se ei toimi tai miksi se toimii. Opiskelijoiden pitää pystyä ymmärtämään, milloin viitekehys toimii tai ei toimi. Lisäksi heidän täytyy oppia hahmottamaan, miksi viitekehys

toimii tai ei toimi, ja perustelemaan kantansa. Argumentin ideana on herätellä opiskelijoita ajattelemaan ketterästi sen sijaan, että he oppivat seuraamaan pelkkää prosessia, ja muuttamaan sääntöjä, kun ne eivät toimi heidän eduksensa. (Product Ownership Ticorporate 7 Demolab edition 2020.) Toisaalta tämä ajattelutapa edellyttää Scrumin peruseriaatteiden hallitsemista, kuten argumentti esittää. Tuoteomista- jaopiskelijoiden kannalta voisi olla parempi, että tämä ketterän ajattelun teema siirrettäisiin lähemmäksi Demo Labin loppuvaihetta, kun heillä on enemmän kokemusta roolissa toimimisesta. Muuten on riskinä, että tämä sekoittaa ajattelua liikaa sivutoimen alussa.

Luentomateriaalissa mainitaan Schwaberin ja Sutherlandin (2017, 6) esittämä tuoteomistajan vastuu arvon maksimoimisesta kehitystiimin kehittämän tuotteen osalta. Lisäksi vision kommunikoiminen, kehitysjonon hallinta ja sidosryhmien edustaminen mainitaan tuoteomistajan tärkeimmiksi tehtäviksi. Visio määritellään kykynä ajatella tai suunnitella tulevaisuutta hyödyntäen mielikuvitusta tai viisautta. Visio voi olla myös lauseen muotoon kirjoitettu lyhyt selostus halutusta tulevaisuuden tilasta, joka voidaan saavuttaa kehittämällä ja käyttämällä tuotetta. Määritelmä eroaa Clintonin (2010, 52) määritelmästä, sillä tuoteomistajaa ei referoida tuotteen jaetun ja yhteisen vision kulminoitumispisteenä. Tuoteomistajan kykyä nähdä tuote asiakkaan tai loppukäyttäjän näkökulmasta ja heidän edustamisestaan tulisi painottaa. Tuoteomistajan tehtävää jaetun vision kommunikoijana on tarpeen myös tuoda esille. Materiaalissa havainnollistavina esimerkkeinä visionääreistä käytetään Steve Jobsia ja Elon Muskia. Tärkeimmiksi tuoteomistajan taidoiksi mainitaan erinomaiset kommunikointitaidot, rohkeus sanoa ei, kyky viestiä eri asiantuntijoiden kanssa ja erinomaiset kuuntelutaidot. (Mt.)

Tuotteen kehitysjonosta mainitaan, että tuoteomistajan tulee järjestellä kehitysjonoa arvon ja sidonnaisuuksien perusteella, jotta asetetut tavoitteet voidaan saavuttaa. Tärkeää on myös kehitysjonon läpinäkyvyys, ja että kehitystiimi ymmärtää kehitysjonon kohdat tarvittavalla tasolla. Hyödynnettäväksi työkaluiksi mainitaan Definition of Ready eli INVEST-malliin perustuva tarkistuslista käyttäjäkertomuksen valmiudesta tai laadunvarmistuksesta. Muiksi työkaluiksi mainitaan Conditions of Satisfaction eli hyväksymiskriteerit, ja 'valmiin määritelmä' osana laadunhallintaa. (Mt.)

Kompleksien työkalujen esittäminen ensimmäisellä luennolla Demo Labin alkuvaiheessa on kyseenalaista, sillä monet niistä käsittelevät käyttäjäkertomuksia, joiden rakentamiseen ei ole käytetty aikaa. Käyttäjäkertomuksien käyttämisen, keräämisen ja kirjoittamisen tulisi tapahtua ennen työkalujen esittelyä, ja työkaluja tulisi esitellä vasta, kun tuoteomistajaopiskelijat hallitsevat käyttäjäkertomusten perusteet.

Luento 2

Luennolla 2 (Product Owner Weekly 2 2020) käsitellään käyttäjäkertomuksia: tarinapisteitä, arvioimista, priorisointia, riippuvaisuussuhteiden käsittelyä ja taskeja. Tarinapisteet on määritelty käyttäjäkertomusten arvioimisen yksikkönä, joka mittaa monimutkaisuutta, työmäärää tai kestoja. Luentomateriaalissa esitetään, että tarinapisteillä ei ole absoluuttisia arvoja, ja jokainen tiimi tulkitsee niitä eri tavalla. Arviointi tapahtuu suunnittelupalaverissa, ja siihen voidaan käyttää Fibonaccin lukujonoa tai ideaalia päivää. Vauhtia mitataan tarinapisteillä, eli kuinka monta pistettä työtä sprintin aikana voidaan toteuttaa. (Mt.) Materiaalin mukaiset tarinapisteiden, arvioimisen ja vauhdin mallit vastaavat Clintonin (2010, 111–114) esittämiä malleja ja Cohnin (2004, 11) esimerkkejä vauhdin ja suhteellisen arvioimistavan osalta.

Materiaalissa absoluuttista ja suhteellista arvioimistapaa käsitellään konkreettisten esimerkkien kautta. Esimerkiksi on helpompaa arvioida, onko matka kotoa lähikauppaan pidempi vai lyhyempi verrattuna kotoa työpaikalle, kuin antaa täsmällistä metrimäärää lähikaupan etäisyydestä. Ihmiset tekevät helpommin virhearvioita, kun käytetään absoluuttista arvioimistapaa verrattuna suhteelliseen. (Product Owner Weekly 2 2020.) Suhteellinen arvioimistapa on käytännössä sama kuin Clintonin (2010, 114) esittämä triangulaatio. Suunnittelupokeri ja Fibonaccin lukujono menetelminä käydään materiaalissa läpi, ja molemmat vastaavat Clintonin (2010, 115–116) malleja.

Tuotteen kehitysjonon tärkeysjärjestyksestä mainitaan, että sen tulisi tapahtua ylhäältä alaspäin, ja mitä korkeammalla kohta on, sitä tarkemmin se tulisi olla määriteltyä. Suosituksena mainitaan 2–3 sprintin työmäärän verran yksityiskohtaisia kohtia kehitysjonon yläpäässä. (Product Owner Weekly 2 2020.) Samaa esittää myös Clinton

(2010, 41–42), tosin suositusta tarkasta työmäärästä ei mainita. Kohtien riippuvuussuhteet voivat vaikuttaa järjestykseen, ja niiden tulisi selvittää kehitysjonoa ope-roidessa. Riippuvuussuhteet voivat aiheuttaa ongelmia suunnittelussa ja tehdä arvi-oinnista vaikeaa. Useimmiten ongelmat johtuvat käyttäjäkertomusten liian pienestä koosta tai kehnosta pilkkomisesta. Usein tämä voidaan ratkaista yhdistämällä pie-nempiä tarinoita suuremmiksi tai vaihtoehtoisella pilkkomiskäytännöllä. (Product Owner Weekly 2 2020.) Samaa esittää myös Cohn (2010, 18) kohtien riippuvuussuh-teista. 'Valmiin määritelmää' ei mainita, mutta sitä on käsitelty luennolla 1 (Product Owner Weekly 2 2020).

Luentomateriaalin lopussa käsitellään, miten käyttäjäkertomukset eroavat taskeista eli tehtävistä. Tehtävät -menetelmässä yksittäinen tehtävä edustaa pientä määrää työtä sprintin kehitysjonossa. Tehtäviä voidaan johtaa käyttäjäkertomuksista tai muista tuotteen kehitysjonon kohdista, eivätkä ne suoranaisesti ole tuoteomistajan vastuulla. (Product Owner Weekly 2 2020.) Tehtävät (tasks) eivät ole osa Schwaberin ja Sutherlandin esittämää Scrum-viitekehystä. Clinton (2010, 61) mainitsee tehtävät osana suunnittelupalaveria. Tehtävät-menetelmää ei oletusarvoisesti käytetä tai esi-tellä Demo Labissa, mutta sen käyttäminen ei ole kiellettyä. Ticorporate 7 -toteutuk-sessa tehtävä -menetelmän esittelemisen havaittiin aiheuttavan huomattavaa se-kaannusta tuoteomistajaopiskelijoiden toiminnassa, joten sen poistaminen luento-materiaalista olisi aiheellista. Mikäli jokin ryhmä käyttää kyseistä menetelmää tulevai-suudessa, voi sivutoimesta vastuussa oleva henkilö sen sijaan antaa henkilökohtaista tukea menetelmän hyödyntämisessä.

5.3 Asiantuntijahaastattelu

Asiantuntijahaastattelua kokeiltiin osana sivutoimen rakennetta, sillä myös Scrum-master -sivutoimessa asiantuntijahaastattelua hyödynnetään osana opiskelijoiden ohjaamista. Kyseisessä haastattelutilaisuudessa paikallisen IT-suuryrityksen kokenut Scrummaster kutsuttiin paikan päälle. Haastateltava kertoi työstään ja Demo Labin Scrummasterit pystyivät esittämään kysymyksiä tai kokemiaan ongelmatilanteita, jo-hon haastateltava pystyi vastaamaan. Näin opiskelijat saivat katsauksen myös yritys-maailman toimintamalleihin. Asiantuntijahaastattelun havaittiin olevan hyödyllinen

ja miellyttävä tilaisuus niin opiskelijoille kuin kyseiselle asiantuntijallekin, joten samaa päätettiin kokeilla tuoteomistajien sivutoimessa.

Asiantuntijahaastattelun tavoitteeksi asetettiin saada tuoteomistajat ajattelemaan tuotekehitystä kokonaisuutena ja roolin ulkopuolella. Lisäksi tarkoituksena oli saada tuoteomistajille näkemystä yritysmaailman tuotekehityksestä ja mahdollisuus esittää kysymyksiä tuotekehityksen ammattilaiselle. Haastattelua ei strukturoitu etukäteen, vaan se tapahtui vapaamuotoisesti.

Demo Labin tuotantovaiheen lopussa paikallinen product leader paikallisella ja kansallisella tasolla merkittävä tuotekehityksen erityisasiantuntija suostui haastattelupyyntöön. Tilaisuutta ei voitu järjestää fyysisessä tilassa pandemiarajoitusten takia, joten haastattelu tapahtui etäyhteydellä. Ennen haastattelua, opiskelijat pääsivät käyttäjätestaamaan asiantuntijan kehittämää virtuaalista projektinhallinta -sovellusta, jossa tuoteomistajat pääsivät vastaamaan tuotekehitykseen ja tuoteomistajuuteen liittyviin kysymyksiin, teemoihin ja kertomaan omista haasteistaan. Asiantuntija kommentoi opiskelijoiden haasteita ja antoi ratkaisuehdotuksia. Haastattelun aikana asiantuntija vastasi esitettyihin kysymyksiin ja pyysi opiskelijoilta palautetta tuotteesta.

Haastattelun aikana havaittiin, että opiskelijat eivät olleet kovin aktiivisia kysymysten suhteen. On mahdollista, että etäyhteys oli osatekijä passiivisuuteen, sillä monelle etäyhteyden kautta tapahtuva kommunikointi ryhmänä oli uutta. Päätös etäyhteydestä myös jouduttiin tekemään nopealla aikataululla. Toisaalta etäyhteys voi myös auttaa henkilöitä, joille fyysisessä tilassa tapahtuva kommunikointi on haasteellisempaa. Kysymyksiä voi olla myös haasteellista esittää, mikäli opiskelija ei niitä ole miettinyt etukäteen. Haastattelun jälkeen havaittiin peliprojektien tuoteomistajien osittainen tyytymättömyys tilaisuuteen. Syyksi he mainitsivat, että vaikka haastattelu oli mielenkiintoinen, ei sovelluskehitykseen liittyvä tuotekehitys merkittävästi auttanut peliprojektien tuoteomistajien ongelmia. Kyseiset tuoteomistajat olivat samaa mieltä siitä, että mikäli kyseessä olisi ollut pelituottaja, olisi tilaisuudesta voinut olla heille enemmän hyötyä.

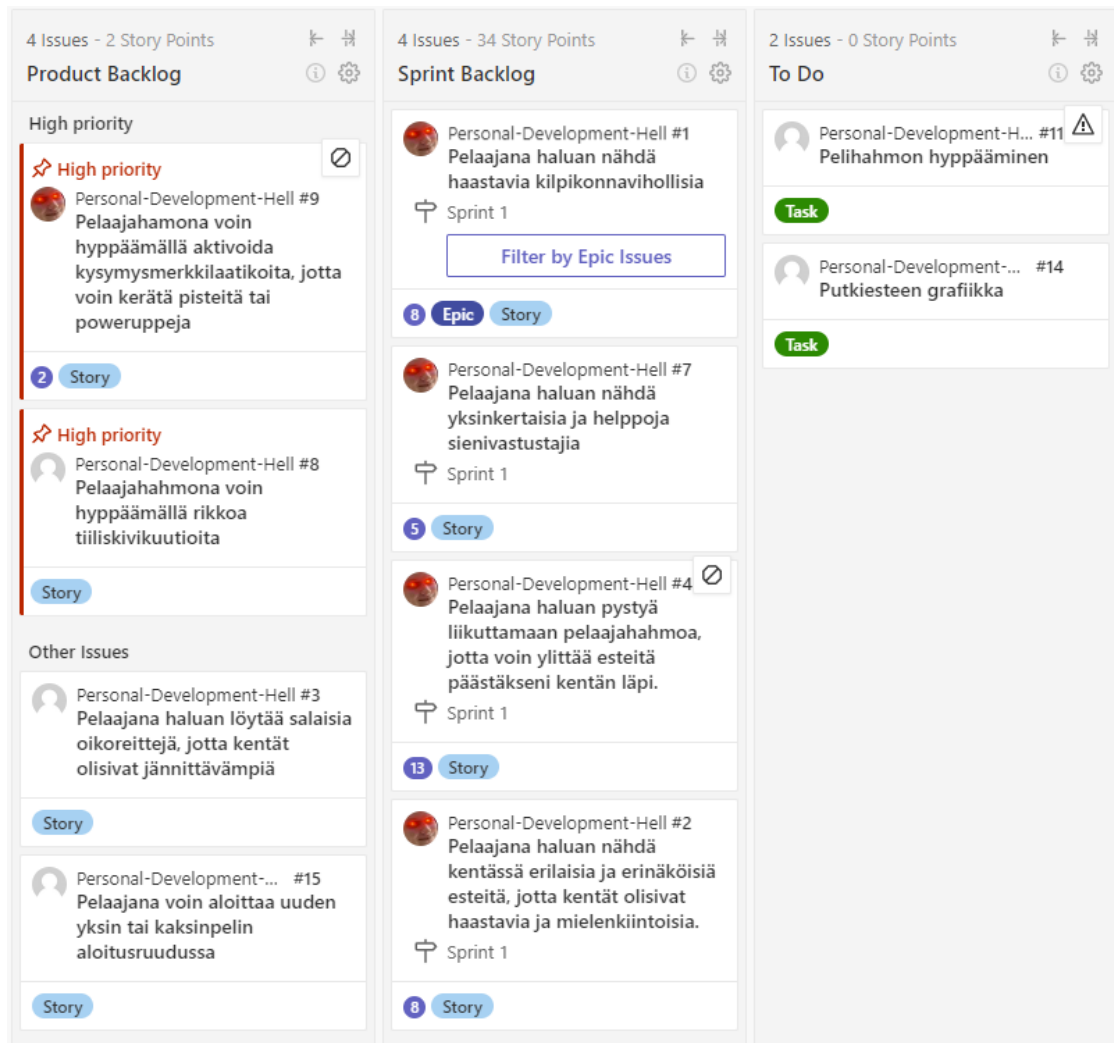
Täten on perusteltua, että asiantuntijahaastatteluja voisi sivutoimessa olla useampia, mutta eri osa-alueilta, mikäli mahdollista. Voidaan myös esittää, että haastattelutilaisuutta tulisi strukturoida hieman etukäteen. Esimerkiksi opiskelijoiden kysymykset voitaisiin kerätä ennen haastattelua, jotta heillä olisi aikaa pohtia niitä.

5.4 Käyttäjäkertomus- ja Zenhub-työpaja

Demo Labin alkuvaiheen jälkeen havaittiin, että käyttäjäkertomukset aiheuttivat haasteita opiskelijoille. Käyttäjäkertomusten kirjoittaminen malliin oli monen mielestä hankalaa, eikä menetelmää tarkoitusta ollut ymmärretty. Lisäksi havaittiin, että tuoteomistajat ja muut opiskelijat kokivat Scrumin ja Zenhubin termistöjen väliset ristiriidat sekaviksi. Tämä aiheutti paljon haasteita suunnittelupalavereissa ja tuotteen kehitysjonon hallitsemisessa.

Havaintojen perusteella päätettiin kokeilla viikkotapaamisen yhteydessä työpajaa, jonka tavoitteena oli auttaa Zenhubin ja Scrumin termistöjen väliseen sekaannukseen ja näyttää esimerkkitapaus siitä, kuinka Zenhubia voidaan soveltaa Scrumissa. Työpajassa käsiteltiin kuvitteellista Zenhub-sovellukseen rakennettua pelituotteen esimerkkiä, jossa havainnollistettiin tuotteen ja sprintin kehitysjonoa, käyttäjäkertomuksia ja tehtäviä (kuvio 3).

Työpajan jälkeisten havaintojen ja opiskelijoiden palautteen perusteella työpajasta oli selkeää hyötyä projekteille. Tätä tukee myös ryhmähaastattelun tulokset. Ryhmähaastattelun perusteella kävi ilmi, että työpajan koettiin selkeyttävän sekaannusta Zenhubin termistön suhteen yhden tuoteomistajan mielestä. Kaksi tuoteomistajaa koki käyttäjäkertomusten havainnollistamisen hyödylliseksi. Näiden tietojen perusteella voidaan esittää, että käyttäjäkertomus-työpaja voitaisiin järjestää heti Demo Labin esituotantovaiheessa tai tuotantovaiheen alussa sekaannusten välttämiseksi.



Kuvio 3. Kuvakaappaus Zenhub-sovelluksesta.

5.5 Manuaali

Demo Labin alussa aloitettiin rakentamaan tuoteomistajille tarkoitettua tuoteomistajuutta käsittelevää manuaalia, jota kutsuttiin työnimellä Product Owner Survival Guide. Manuaalin toiminta-ajatuksena oli toimia sivutoimen päämateriaalina, ja että tuoteomistajat voivat tarvittaessa lukea ja tarkistaa manuaalista, kuinka toimia ongelmatilanteissa. Manuaalin tavoitteena oli sisältää tuoteomistajuuteen liittyvät ydinosaamisalueet kirjoitetussa muodossa, ja antaa havainnollistavia esimerkkejä esimerkiksi käyttäjäkertomusten kirjoittamisesta ja muista hyödynnettävistä pakollisista tai vapaaehtoisista työkaluista ja menetelmistä.

Manuaali koostui seuraavista aihealueista

- tuoteomistajan rooli, vastuut ja tehtävät
- tuotteen kehitysjono
- käyttäjäkertomukset
- miksi käyttäjäkertomuksia?
- hyväksymiskriteerit ja työkalut: Definition of Ready, Conditions of Satisfaction, Definition of Done.

Tuoteomistajan vastuiksi manuaalissa mainitaan liiketoiminnan tavoitteiden saavuttaminen, lisäarvon maksimoiminen ja sidosryhmien edustaminen. Lisäksi tuoteomistaja on vastuussa vision rakentamisesta ja sen kommunikoimisesta sidosryhmille, kehitystiimille ja asiakkaille. Tuoteomistaja on vastuussa projektin rahoituksesta, vaatimuksista ja tavoitteista. Tärkeimmäksi tuoteomistajan yksittäiseksi toiminnoksi mainitaan tuotteen kehitysjonon ylläpitäminen, hoitaminen ja priorisointi. Vision määrittelyä edustavan ”projektin motiivia ja haluttua lopputulosta”. Vision tulisi olla tarkka kuvailu tuotteen ydinominaisuuksista ja projektin tavoitteista, mutta sopivan lavea, ettei se estä luovuutta. Visiossa asiakkaat ja heidän tarpeensa kuvaillaan, sekä vastataan siihen, kuinka ne voidaan tyydyttää.

Tuoteomistajan vastuut manuaalissa vastaavat Clintonin (2010, 52) mallia sillä poikkeuksella, että manuaalissa julkaisuista ja niihin liittyvistä operaatioista ei ole mainintaa, joten manuaali on siltä osin puutteellinen. Visio on hyvin samankaltainen kuin Kiviahon (2019) esittämä määritelmä visiosta. Tosin verrattuna Clintonin näkemykseen visiosta, jossa tuoteomistajan tulisi nähdä tuote loppukäyttäjän ja asiakkaan näkökulmasta, sekä ymmärtää heidän toiveensa että tarpeensa tuotteen suhteen, ei manuaalissa ole mainintaa. Manuaalin visioon voisi lisätä Clintonin näkemyksen painotuksen, sillä se on yksi tuoteomistajan tärkeimmistä vastuista. Lisäksi julkaisunhallinta ja siihen liittyvät operaatiot tulisi käydä läpi manuaalissa.

Manuaalissa tuotteen kehitysjonon mainitaan olevan lista ”tuotteen elinkaaren aikana kehitettävistä, lisäarvoa tuottavista ominaisuuksista tai tavoitteista käyttäjän näkökulmasta”. Maininta tuotteen kehitysjonon muuttuvasta luonteesta ja muokkautu-

vuudesta liiketoiminnan vaatimukseen vastaamiseksi löytyy Schwaberin ja Sutherlandin (2017, 15–17) mukaisesti. Visuaalista havainnollistamista ei kehitysjonosta tosin ole. Tuotteen kehitysjonon kerrotaan koostuvan tuotteen kehitysjonon kohdista, kuten käyttäjäkertomuksista ja piikeistä. Piikille ei anneta määritelmää, joka on selkeä puute. Tähän on mahdollista käyttää Clintonin (2010, 96) esimerkkiä piikistä.

Käyttäjäkertomusten määritelmä näyttää perustuvan Clintonin (2010, 87–88) määritelmään, sillä erotuksella, että käyttäjäkertomusten mainitaan olevan ”placeholde-reita” eli fyysisiä representaatioita aiheen keskustelulle. Clintonin mallipohjaa käyttäjäkertomuksen käyttäjäroolille, tavoitteelle ja käytön syyille havainnollistetaan. Tosin manuaalissa mainitaan, että käyttäjäkertomuksissa tulisi välttää erityisen teknisiä termejä, jotta myös sidosryhmät voivat ymmärtää, mistä on kyse. Käyttäjäkertomusten käytöstä on kirjoitettuun muotoon laitettu, havainnollistava esimerkki, mutta ei visuaalista esimerkkiä. Clintonin (2010, 89) esittämää tapaa pilkkoa käyttäjäkertomuksia käytetään manuaalissa, tosin termillä ”vertical slice”.

Manuaalissa mainitaan ’valmiin määritelmä’, INVEST-malli, tyytyväisyyskriteerit (Conditions of Satisfaction tai acceptance criteria) ja Definition of Ready. INVEST-malli perustuu Clintonin (2010, 92) malliin. Definition of Ready (myöh. DoR) määritellään työkaluna tai menetelmänä selvittämään, onko käyttäjäkertomus tarpeeksi määritelty, jotta se voidaan ottaa suunnittelupalaveriin. Sen tarkoitus on toimia yksinkertaisena tarkistuslistana ja apuna käyttäjäkertomusten määrittelemisessä. DoR voi sisältää mitä tahansa tuoteomistajan valitsemia yksinkertaisia ’valmiin määritelmän’ kaltaisia testejä. Manuaalin mukaan esimerkiksi INVEST-malli voi toimia yhtenä näistä testeistä.

Tyytyväisyyskriteerit perustuvat Clintonin (2010, 90) määritelmään ja se vastaa Cohnin (2004, 12) käyttämää määritelmää, jota Cohn kutsuu termillä hyväksymiskriteerit (acceptance criteria). Materiaalissa tyytyväisyyskriteerit ovat lupaus toteutuneista ominaisuuksista ja lopputuloksista. Tyytyväisyyskriteerien mainitaan toimivan ikään kuin ’valmiin määritelmänä’, mutta käyttäjäkertomuskohtaisesti.

’Valmiin määritelmä’ materiaalissa kuvaillaan laadunvarmistuksen tarkistuslistana, joka varmistaa, että kaikilla on yhteinen näkemys siitä, mikä on valmis tai laadukas.

Sille on myös useita tasoja, kuten julkaisutaso, sprinttitaso ja käyttäjäkertomustaso. Sprinttitasolla sen tarkoituksena on varmistaa, ettei käyttäjäkertomukseen liity teke-
mätöntä työtä eli teknistä velkaa, jotta se voidaan todeta 'valmiiksi'. Manuaalin mal-
lin ydin on samankaltainen kuin Clintonin (2010, 99) malli, mutta eri tavalla kuvail-
tuna. Manuaalissa 'valmiin määritelmän' mainitaan olevan tuote- ja tiimikohtainen,
johon Clinton puolestaan ei ota suoranaisesti kantaa. 'Valmiin määritelmästä' on kir-
joitettuun muotoon rakennettu havainnollistava esimerkki.

Manuaalin suurimpia puutteita on tekstin keho ja ajoittain epäjohtonmukainen for-
matointi, joten visuaalinen ulkoasu on sekava. Visuaalisia esimerkkejä juuri ole, muu-
ten kuin tekstin muodossa. Tekstissä käytetään paljon lihavoitua, kursivoitua ja eri-
kokoisia otsikkoja, mikä tekee lukemisesta hankalaa. Manuaali perustuu pitkälti Clin-
tonin, Schwaberin ja Sutherlandin ja ajoittain Cohnin esittämiin malleihin, joten voi-
daan todeta, että sisältö on luotettavaa, mutta ei kovin käyttäjäystävällistä. Tätä tu-
kee myös tuoteomistajien kokemukset manuaalista, erityisesti visuaalisten esimerk-
kien puutteen suhteen. Lisäksi esimerkit ovat osin englanniksi, osin suomeksi. Voi-
daan siis perustella, että tähän tulisi kiinnittää erityistä huomiota ei pelkästään op-
paassa, vaan kaikissa materiaaleissa. Tulevaisuudessa Ticorporate voi toimia suurim-
malta osin englanniksi, sillä 2020 syksyllä ensimmäinen vuosiryhmä aloitti opintonsa
englanninkielisessä pelituotannon tradenomikoulutuksessa.

Tuoteomistajien mielipiteet

Oppaasta keskusteltiin myös haastattelussa. Kolme neljästä tuoteomistajasta ei ollut
lukenut manuaalia kuin "pari kertaa". Yksi haastateltavista mainitsi oppaan olevan
liian pitkä luettavaksi ja lukemisen haastavaksi. Manuaalista mainittiin, että paperi-
nen versio olisi voinut olla parempi, ja yhden tuoteomistajan mielestä kokonaisuuk-
sien loppuun olisi voinut tiivistää, että "mitä sun pitää tehdä, nyt kun olet tuoteomis-
taja". Yleisimpien kysymysten ja vastausten osiota pyydettiin. Visuaalisesti havainnol-
listavia esimerkkejä toivottiin, vaikkapa kuvallisten tuotteen kehitysjojen tai käyt-
täjäkertomusten muodossa.

Kysyttäessä manuaalin hyödyllisyydestä, kolme neljästä tuoteomistajasta oli sitä mieltä, että manuaalista voisi olla hyötyä. Yksi haastateltavista koki sen etenkin alkuvaiheessa hyödylliseksi. Toinen tuoteomistaja mainitsi, että manuaalista oli erityisesti hyötyä käyttäjäkertomuksia kirjoittaessa. Lisäksi yksi haastateltavista esitti, että ongelmatilanteissa manuaalista voisi tarkistaa, kuinka toimia. Tämä olikin yksi manuaalin tarkoitusperistä.

5.6 Julkaisut ja tuotantovaiheet

Julkaisuja tai tuotantovaiheita Clintonin (2010, 53) mallin mukaisesti ei sivutoimen rakenteessa otettu huomioon. Ticorporate Demo Lab yhdessä ohjelmistotuotannon kanssa muodostavat Clintonin esittämät tuotantovaiheet. Demo Labin aikana on kolme demotilaisuutta, jossa tuotteen edistystä eli inkrementtiä esitellään. Demotilaisuudet soveltuvat Clintonin esittämiin julkaisuihin, jotka muodostavat pidemmän aikavälin tavoitteita ja kehityskokonaisuuksia.

Voidaan esittää, että tuoteomistajien tulisi tuottaa Demo Labin aikana kolme julkaisusuunnitelmaa, joista jokaiseen julkaisuun valitut käyttäjäkertomuskokonaisuudet tai painotukset käyvät ilmi. Esitetty toimintatapa voisi vähentää Demo Labin aikana havaittua projektien virhearviointien määrää, koska Clintonin esittämä julkaisusuunnitelma pakottaa tuoteomistajan tarkastelemaan edistymistä suhteessa seuraavaan julkaisuajankohtaan eli Demotilaisuuteen.

Julkaisusuunnitelmat ja niiden toteutumisen arviointi voisivat toimia myös portfolio-materiaalina tai arvioitavina kokonaisuuksina. Demotilaisuuksien jälkeen projektit voisivat kokoontua julkaisupalaveriin Clintonin mallin mukaisesti ja muokata julkaisusuunnitelmaa. Tämä edellyttää tuotantovaiheiden esittelyä ja opettamista tuoteomistajille, ja muutoksia sivutoimen rakenteeseen. Mikäli julkaisuhallinta otetaan osaksi sivutoimen rakennetta, tulee tuoteomistajien opetella julkaisusuunnitelmien tekeminen ja julkaisupalaverien pitäminen.

5.7 Tuoteomistajien kokemuksia Demo Labista

Kaikki kuusi tuoteomistajaa kutsuttiin ryhmäteemahaastatteluun. Kaikki suostuivat haastatteluun, mutta vain neljä osallistui. Haastattelun teemat oli jaettu ohjelmistotuotantoon ja Ticorporateen, tuoteomistajuuteen ja sivutoimen rakenteeseen.

Kolme neljästä haastateltavasta koki oman vision kommunikoinnin haastavaksi. Kahden mielestä haasteet johtuivat vision epätarkkuudesta. Käyttäjäkertomusten mallin kirjoittaminen koettiin hankalaksi. Myös Scrumin ja Zenhub -projektinhallintasovelluksen termistöt ”meni hieman sekaisin siinä aluksi”. Termistön sekavuus helpotti, kun tiimin kanssa sovittiin selkeät säännöt siitä, kuinka termistöt vastaavat toisiaan.

Tärkeimmiksi tuoteomistajan taidoiksi mainittiin kommunikointi tiimin kanssa, rohkeus sanoa ei, päätöksien perustelu ja projektin kokonaiskuvan hahmottaminen. Yksi haastateltavista toivoi, että projektien tilaa seurattaisiin enemmän, jotta tuoteomistajien haasteista voitaisiin olla tietoisia. Kaksi haastateltavaa tuoteomistajaa koki, että toiminnan tarkastelua ja palautteen antamista olisi Demo Labin alussa ollut enemmän:

Nyt oli vähän se, että välillä oli kiva tehdä omassa rauhassa ja jätettiin vähän omalle vastuulle kaikki, mutta toisaalta, olisi ollut kiva jos olis tullut sellainen ”tarkastaja” niin sanotusti, joka olisi tullut katsomaan, että mitäs olette tehneet ja missäs teillä on [ongelmaa] ja näyttäkää, mitä olette saaneet aikaiseksi. (Haastateltava 2.)

Kaikki 4 haastateltavaa olivat samaa mieltä siitä, että tuoteomistajana toimiminen projekteissa on ollut palkitsevaa. Yhden tuoteomistajan mielestä Demo Lab on ollut opiskelun parasta aikaa. Yksi tuoteomistajista koki projektien päättymisen surullisena:

On ollut kiva tehdä tällaista projektia, jossa on ollut yksi asia, mihin on keskitytty ja sitten kun ei yleensä noilla kursseilla ole tutustunut muuhun porukkaan, tai mä ainakaan. Oon saanut tutustua tuohon tiimiin ihan kunnolla, ja sitten myös muihinkin, kun oltiin kaikki tuolla yhdessä. Oli niin hauskaa. (Haastateltava 1.)

Kysyttäessä, kokevatko tuoteomistajaopiskelijat sivutoimen vaikuttavan työllistymiseen, yksi haastateltavista koki tuoteomistajuuden valmistelevan yrittäjyyteen. Toisaalta kaksi neljästä tuoteomistajasta koki, ettei voisi toimita jonkun toisen henkilön idean tai projektin tuoteomistajina.

5.8 Muita sivutoimen puutteita

Opiskelijoilla ei välttämättä ole kovin merkittävää ymmärrystä tuoteomistajan toiminnasta kokonaisuutena ennen sivutoimen valitsemista ohjelmistotuotannon analyysin perusteella. Täten tuoteomistajan sivutoimea voisi esitellä ennen kuin opiskelijat valitsevat sivutoimensa. Lisäksi sivutoimen valitsemisen jälkeen, sivutoimen rakenne, tavoitteet ja aikataulu voitaisiin esitellä ensimmäisellä viikkotapaamiskerralla.

Vision rakentamiseen ei ole mallia tai harjoitteita. Demo Labin aikana havaittiin useasti, että tuoteomistajilla ei aina ollut vahvaa visiota tuotteesta, joka johti ongelmiin projektityöskentelyssä ja tuotannossa. Osa tuoteomistajista ei aina osannut kertoa oman projektinsa kehittäjille, miten käyttäjäkertomus tulisi tarkalleen toteuttaa. Vision löytämiseen voisi käyttää jonkinlaista harjoitetta. Tähän tulisi kiinnittää huomiota ja pyrkiä löytämään ratkaisuja.

Lisäksi viikkotapaamisissa havaittiin, että Zenhub-sovelluksen ja Scrumin termistö menevät sekaisin. Vaikka käyttäjäkertomustyöpaja ratkaisi kyseisen ongelman, tulisi asiaa selventää aikaisemmassa vaiheessa havainnollistamalla, kuinka termistöt vastaavat toisiaan.

6 Kooste kehittämisehdotuksista

Analyysin perusteella voidaan todeta, että sivutoimen rakenteessa on paljon parannettavaa. Erityisesti käyttäjäkertomuksien käsittelyyn tulisi käyttää resursseja. Täten Demo Labin esituotantovaiheessa tai ennen tuotantovaiheen alkua tulisi järjestää käyttäjäkertomustyöpaja, jossa käyttäjäkertomuksia kirjoitetaan ohjatusti yhdessä. Zenhub-termistöä tulisi selventää suhteessa Scrumin termistöön. Manuaalin kirjoitusasu tulisi korjata ja visuaalista havainnollistamista parantaa.

Lisäksi esittäisin, että luentoja tulisi sivutoimen aikana olla kolme. Luento 1 esittelisi sivutoimen rakenteen, mahdolliset arvioitavat tuotokset, aikataulun ja tuoteomistajan vastuut yleisellä tasolla. Luento 2 keskittyisi tuoteomistajan vastuisiin, tuotteen kehitysjonoon ja käyttäjäkertomuksiin. Teemoina luennolla 3 olisi käyttäjäkertomuksiin liittyvät työkalut, menetelmät ja operaatiot.

Kaikkien materiaalien kääntäminen englanniksi tulisi aloittaa. Vision rakentamiseen tulisi olla harjoite tai menetelmä. Julkaisunhallinta tulisi ottaa paremmin osaksi sivutoimea tekemällä julkaisusuunnitelmia julkaisupalavereissa ja raportoimalla demotilaisuuksissa kehittämisen edistymisestä suhteessa julkaisutavoitteisiin. Peliteollisuuden puolelta tulisi pyrkiä löytämään asiantuntija, esimerkiksi pelituottaja, haastatella varten.

Portfolion suhteen tulisi tehdä päätös siitä, onko tuoteomistajan sivutoimesta mahdollista rakentaa kokonaisuutta, jota arvioida numeroasteikolla hylätyn ja hyväksytyin sijaan. Mikäli se on mahdollista, tulisi päättää, mitkä tuotokset arvioidaan.

7 Pohdinta

7.1 Tulokset

Opinnäytetyön tavoitteena oli kehittää tuoteomistaja-sivutoimen rakennetta ja saada konkreettisia, hyvin perusteltuja kehitysehdotuksia, jotka olisi helppo toteuttaa käytännössä. Opinnäytetyön tuloksena näitä konkreettisia ehdotuksia myös pystyttiin tuottamaan. Seuraava Ticorporate-toteutus alkaa 2021, joten ainakin osa kehitysehdotuksista olisi mahdollista viedä käytäntöön ennen sen alkamista.

Merkittävin rajoite opinnäytetyön tuloksille oli pedagogisen näkökulman puuttuminen, sillä tutkijalla ei ole pedagogista koulutustausta. Tulosten perusteella on voitu päätellä, mitä sivutoimen tulisi sisältää, tai miten sitä tulisi muuttaa teorian perusteella, mutta toteutustavan pedagogisuudesta ei ole näyttöä.

Tuloksia voidaan myös hyödyntää tietojenkäsittelyn tutkinto-ohjelman sisältöjen suunnittelun tukena soveltavin osin. Mikäli Ticorporate Demo Labin kaltaista menetelmää käytetään jossakin toisessa organisaatiossa, voidaan opinnäytetyön tuloksia tarkastella kriittisesti sen suhteen, mitä niistä voidaan kyseisessä organisaatiossa hyödyntää.

Materiaalien englanninkielinen kääntäminen ja uusiminen on jo osittain aloitettu vuoden 2020 syksyllä. Oppaasta pyritään julkaisemaan uusi versio ennen Ticorporate 8 toteutuksen alkamista 2021. Manuaalista saatuja palautteita ja opinnäytetyön tuloksia hyödynnetään uutta versiota kirjoitettaessa.

7.2 Tutkimuksen luotettavuustarkastelu

Tutkimuksen luotettavuuteen vaikuttaa suuri määrä erilaisia tekijöitä. Esimerkiksi lähdekirjallisuuden suhteen, Scrumin teoriaan perustuvaa tutkimusta on vaikeaa tehdä luotettavasti ilman Schwaberin ja Sutherlandin teosta *The Scrum Guide* (2017), sillä se on Scrum-viitekehyksen alkuperäiset ja ortodoksiset säännöt. Tältä osin

teoriapohja Opinnäytetyössä viitattu Clintonin teos Agile Game Production (2010) alkaa olla osittain ajastaan jäljessä. Pelituotanto on muuttunut 10 vuodessa valtavasti. Myös Cohnin User Stories Applied (2004) materiaali on myös 16 vuotta vanhaa. Toisaalta se on ensimmäisiä laajoja teoksia käyttäjäkertomusten käytön suhteen, ja siten laajasti referoitu klassikko.

Tutkimusmenetelmien suhteen, tutkijalla ei ole ollut aikaisempaa kokemusta kehittämistutkimuksesta tai teemahaastattelun järjestämisestä. Teemahaastattelun tuloksiin on myös voinut vaikuttaa se, että se on toteutettu virtuaalisesti. Virtuaalinen ympäristö ja ryhmähaastattelun muoto on voinut vaikuttaa informanttien tapaan kommunikoida. Toisaalta on vaikeaa todeta, onko se helpottanut vai vaikeuttanut haastateltavien halukkuutta tai kykyä informaation jakamiseen.

Aineiston luotettavuutta arvioidessa, tulee ottaa huomioon aineiston analyysi ja tutkijan havainnot. Kirjallisen aineiston analyysiin on voinut vaikuttaa se, että ohjelmistotuotannon ja Demo Labin aineiston tekijät ovat olleet tutkijan kollegoja. Lisäksi sivutoimen analysoidut dokumentit ovat olleet tutkijan tuottamia. Nämä seikat ovat voineet vinouttaa analyysia tiedostomattomasti. Tutkija on siten voinut analysoida tuloksia alitajuisesti pienemmällä tai suuremmalla kriittisyydellä, riippuen tutkijan suhteesta henkilöihin ja tuotettuihin sisältöihin.

Tutkijan tekemiä havaintoja Demo Labin aikana on kirjattu muistiin, mutta ei riittäväällä tieteellisellä tarkkuudella. Tästä syystä tutkimuksessa hyödynnettyjä havaintoja on pyritty käyttämään argumenttien tukena triangulaation tapaisesti, mutta suuntaa antavasti. On vaikeaa arvioida, tulisiko toinen tutkimuksen toteuttaja samoihin johtopäätöksiin ja tuloksiin. Ticorporate on ympäristönä ja toimintatapana melko harvinainen. Tästä syystä tulosten siirrettävyys on heikko.

7.3 Jatkotutkimus

Kirjoittamisprosessin ja Ticorporate 7-toteutuksen aikana tutkija on useasti pohtinut sitä, että mihin tietoon perustuen tuoteomistajan pitäisi päätöksiä tarkalleen tehdä? Scrum ei merkittävästi ota kantaa ns. ”Discovery”-prosessiin, eli mistä tieto tuotekehitykseen liittyvän ominaisuuden tarpeellisuudesta saadaan. Viitekehityksessä ei mainita sen tarkemmin, mistä kyseistä dataa voidaan kerätä. Scrum-viitekehityksessä puhutaan paljon lisäarvon maksimoimisen tärkeydestä, mutta ei ollenkaan siitä, miten tieto ominaisuuden ”arvosta” muodostetaan. Scrum vaikuttaisi perustuvan ajatukseen, että ensin rakennetaan tuotetta nopeasti, ja korjataan palautteen perusteella jälkeinpäin.

Tätä lähestymistapaa on kritisoinut erityisesti Jeff Patton, joka on kirjoittanut paljon käyttäjäsentrisestä tuotekehityksestä ja esittänyt aiheen suhteen mielenkiintoisia ajatuksia ja ratkaisuehdotuksia. Vaikka Mike Cohnin teos *User Stories Applied* (2004) on käyttäjäkertomusten suhteen klassikko, se on monelta osin vanhentunut. Patton tunnetaan kirjoittamastaan kirjasta *User Story Mapping*, joka käyttää erilaista ja hieman modernimpaa lähestymistapaa käyttäjäkertomuksien käytössä ketterässä ohjelmistokehityksessä. Tuoteomistajille järjestetystä asiantuntijahaastattelusta oli siis paljon hyötyä myös minulle, sillä hänen suositustensa perusteella tutustuin nimenomaan Pattonin ajatuksiin ja teoksiin. Mahdollisena jatkotutkimusaiheena voisi siis olla sivutoimen tarkasteleminen Pattonin ajatusten näkökulmasta.

Lähteet

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. 2001. Manifesto for Agile Software Development. Verkkojulkaisu. Viitattu 06.08.2020. <https://agilemanifesto.org/>.

Bister, T. 2019. Tietojenkäsittelyn opinnäytetyö: Viittoja ja karttoja tutkimisen ja kehittämisen teille. Jyväskylän ammattikorkeakoulun julkaisuja 272.

Bister, T., Immonen, J. & Tuikka, T. 2016. Ticorporateyrityssimulaatio JAMK:n tietojenkäsittelyn koulutuksessa. Julkaisussa Yrittäjyyskasvatuspäivät. Conference proceedings. Toim. M. Suoranta, P. Patja, I. Aaltio & M. Tunkkari-Eskelinen. Yrittäjyyskasvatuspäivät Jyväskylässä 13.-14.9.2016. Jyväskylä University. School of Business and Economics.

Clinton, K. 2010. Agile game development with Scrum. Yhdysvallat: Addison-Wesley signature series.

Cohn, M. 2004. User stories applied. For Agile software development. Yhdysvallat: Addison-Wesley signature series, p. 13.

Hirsjärvi, S. & Hurme, H. 2015. Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. E-kirja. Helsinki: Gaudeamus Helsinki University Press. Viitattu 06.08.2020. <https://www.ellibslibrary.com/book/9789524958868>.

Kananen, J. 2012. Kehittämistutkimus opinnäytetyönä: Kehittämistutkimuksen kirjoittamisen käytännön opas. Jyväskylä: Jyväskylän ammattikorkeakoulu.
Kananen, J. 2008. Kvali: Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Kiviaho, N. 2016. Ketteryys projekteissa - Scrumista vastaus? Powerpoint – luentomateriaali Jyväskylän ammattikorkeakoulun tietojenkäsittelyn ohjelmistotuotanto-opintojaksolla. Viitattu 28.10.2020. <https://optima.jamk.fi/learning/id2/bin/user?rand=9102>.

Kniberg, H., Skarin, M. 2010. Kanban and Scrum: making the most of both. Enterprise Software Development series. E-kirja agileinnovation.eu - sivustolla. Viitattu 06.08.2020. http://www.agileinnovation.eu/wordpress/wp-content/uploads/2010/09/KanbanAndScrum_MakingTheMostOfBoth.pdf.

Lee, J. 2020. Failed #SquadGoals. Spotify doesn't use the "the Spotify model" and neither should you. Verkkojulkaisu. Viitattu 06.08.2020. <https://www.jeremiahlee.com/posts/failed-squad-goals/>.

- Ohjelmistotuotanto HTT201101-3001. 2020. Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelman opetussuunnitelma Peppi -järjestelmässä. Viitattu 22.10.2020. https://opetussuunnitelmat.peppi.jamk.fi/HTK2018SPT/course_unit/HTT20101.
- Pernaa, J. 2013. Kehittämistutkimus tutkimusmenetelmänä. Luku Kehittämistutkimus opetuslalla -teoksesta. PDF-tiedosto Helsingin yliopiston tutkimusportaali -sivustolla. Jyväskylä: PS-kustannus. Viitattu 14.10.2020. <https://researchportal.helsinki.fi/en/publications/kehitt%C3%A4mistutkimus-tutkimusmenetelm%C3%A4n%C3%A4>.
- Product Ownership Ticorporate 7 Demolab edition. 2020. Tuoteomistaja -sivutoimen Powerpoint -luentomateriaali Ticorporate Demo Lab -opintojaksolla. Vain sisäiseen käyttöön. Viitattu 28.10.2020.
- Product Owner Weekly 2. 2020. Tuoteomistaja -sivutoimen Powerpoint -luentomateriaali Ticorporate Demo Lab -opintojaksolla. Vain sisäiseen käyttöön. Viitattu 28.10.2020
- Pölkki, T. 2020. Ticorporate – Demolab: Käsikirja opiskelijoille ja henkilökunnalle. Versio 7. Artikkelit Jyväskylän ammattikorkeakoulun tietojenkäsittelyn tutkinto-ohjelman tietovarastossa Github -verkkopalvelussa. Viitattu 28.10.2020. <https://github.com/jamktiko/tc-demolab-manuaali/blob/master/index.md>.
- Schwaber, K., Sutherland, J. 2018. The Scrum guide™. The definitive guide to Scrum: the rules of the game. Verkkojulkaisu. Viitattu 12.02.2020. <https://www.scrum.org/resources/scrum-guide>.
- Schwaber, K., Sutherland, J. 2017. Scrum-opas™: Scrumin määritelmä ja pelisäännöt. Verkkojulkaisu. Viitattu 12.03.2020. <https://scrumguides.org/download.html>.
- Singer, R. 2019. Shape Up: Stop Running in Circles and Ship Work that Matters. Lattava PDF-tiedosto Basecamp.com – verkkosivustolla. Versio 1.8. Viitattu 06.08.2020. <https://basecamp.com/shapeup/shape-up.pdf>.
- Tuomi, J. & Sarajärvi, A. 2018. Laadullinen tutkimus ja sisällön analyysi. Helsinki: Kustannusosakeyhtiö Tammi.

Liitteet

Liite 1. Teemahaastattelun teemat

Teemahaastattelurunko

Scrum ohjelmistotuotannossa ja Ticorporatessa

- Minkälaista kokemusta Scrumista ennen ohjelmistotuotantoa?
- Minkälaisia valmiuksia ohjelmistotuotanto antoi Scrum - viitekehukseen? Entäpä tuoteomistajuuteen?
- Miten kehittäisit Scrumin opetusta ohjelmistotuotannossa? Entäpä Ticorporatessa? Miksi?

Tuoteomistajuus

- Millainen on hyvä tuoteomistaja?
- Minkälaisia kokemuksia jäi tuoteomistajuudesta? Visio, Backlog, käyttäjäkertomukset, vastuut.
- Mikä osa-alue oli haastavin? Miksi?
- Koetteko tuoteomistajuusopintojen vaikuttavan työllistymiseen? Miten? Miksi?
- Mitkä ovat tuoteomistajan tärkeimpiä taitoja?

Sivutoimi Ticorporatessa

- **Manuaali.** Kuinka usein luitte manuaalia? Miksi? Manuaalin hyödyllisyys? Kuinka sitä voisi kehittää?
- **Luennot.** Minkälaisia kokemuksia luennoista jäi? Mitä opitte luennoista? Mitä aiheita olisitte halunneet käsitellä? Miksi?
- **Viikkotapaamiset.** Koitteko viikkotapaamiset hyödyllisiksi? Miksi? Kuinka viikkotapaamiset tulisi järjestää, jotta ne palvelisivat tarpeitanne?
- **Workshop.** Minkälaisia kokemuksia Käyttäjäkertomus-työpajasta jäi? Kuinka sitä voisi kehittää?
- **Asiantuntijahaastattelu.** Minkälaisia kokemuksia asiantuntijahaastattelusta jäi? Mitä asiantuntijoita haluaisit haastatella? Miksi?
- **Muuta.** Mitä olisi voinut olla? Miten? Miksi? Tarpeita, joihin ei vastattu?