Tampereen yliopisto

Ida Myller

TIETOTURVATESTAUSPROJEKTIEN KEHITTÄMINEN KETTERIEN MENETELMIEN ELEMENTTEJÄ HYÖDYNTÄEN

Diplomityö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Pasi Hellsten
Tarkastaja: Ilona Ilvonen
Huhtikuu 2020

TIIVISTELMÄ

Ida Myller: Tietoturvatestausprojektien kehittäminen ketterien menetelmien elementtejä hyödyntäen
Diplomityö
Tampereen yliopisto
Tietojohtamisen DI-tutkinto-ohjelma
Huhtikuu 2020

Ketterät menetelmät ovat syntyneet vastaamaan ohjelmistokehityksen nopeasti muuttuvan toimintaympäristön tarpeisiin, mutta kiinnostus menetelmien soveltamiseen muissakin konteksteissa on herännyt menetelmien yksinkertaisuuden ja käytännönläheisyyden vuoksi. Tämän tutkimuksen kohteena on tekniseen tietoturvatestaukseen keskittyvä suomalainen kyberturvallisuusalan yritys, jossa projektinhallintaa on viime vuosina kehitetty paljon, mutta jota halutaan kehittää edelleen itseohjautuvampaan ja ketterämpään suuntaan.

Tutkimuksen tavoitteena oli parantaa tietoturvatestausprojektien toteuttamista työntekijöiden näkökulmasta niin, että päivittäinen työnteko tuntuu vähemmän kiireiseltä ja stressaavalta. Tämä pyrittiin saavuttamaan implementoimalla kohdeyritykseen elementtejä ketteristä menetelmistä. Tutkimuksen tuloksena tehtiin konkreettisia muutoksia tietoturvatestausprojektien toteuttamiseen. Tutkimus rajattiin koskemaan vain kohdeyrityksen teknisiä tietoturvatestausprojekteja.

Tämä tutkimus toteutettiin toimintatutkimuksena. Tutkimuksessa haettiin kirjallisuudesta suosittuja ketteriä menetelmiä, joista tunnistettiin keskeisiä elementtejä, kuten itseohjautuvat tiimit, päivittäiset palaverit, retrospektiivit, kommunikointi kasvotusten, ja työn visualisointi. Näihin elementteihin liittyviä asenteita ja mielipiteitä sekä yrityksen nykytilaa kartoitettiin haastattelemalla kohdeyrityksen työntekijöitä. Haastattelujen löydösten perusteella suunniteltiin ensimmäisiä kehitystoimenpiteitä, kuten tiimien implementointi. Tiimien aloittaessa toimintansa toimintatavoista sovittiin yhdessä tiimien kesken ja kehitystä jatkettiin toimintatutkimuksen spiraalin mukaisesti suunnitellen, toteuttaen ja arvioiden kehitystoimenpiteitä yhdessä tiimien kanssa.

Tutkimuksen myötä kohdeyrityksen teknisten tietoturvatestausprojektien toteuttamiseen saatiin tehtyä merkittäviä muutoksia, jotka toteutettiin työntekijöiden toiveiden ja mielipiteiden perusteella. Yksi suurimmista muutoksista tämän tutkimuksen puitteissa oli tiimien implementointi. Tämän lisäksi tiimit päättivät itse keskenään tiimin toimintatavoista ja säännöllisesti järjestettävistä palavereista, kuten päivittäisistä palavereista ja retrospektiiveistä. Toimintatutkimuksen spiraalin mukainen kehityssykli tulee jatkumaan tämän tutkimuksen ulkopuolella, ja kehitystoimia tullaan tekemään edelleen tiimien mielipiteisiin ja kokemuksiin perustuen.

Avainsanat: Ketterät menetelmät, tekninen tietoturvatestaus, toimintatutkimus

ABSTRACT

Ida Myller: Improving security testing projects with elements of agile methods
Master's thesis
Tampere University
Degree Programme in Information and Knowledge Management, MSc(Tech)
April 2020

Agile methods have been created to answer to the rapidly changing environment's needs in software development, but an interest in applying agile to other contexts as well has arisen due to the simplicity and practicality of agile methods. This research is focusing on a Finnish cybersecurity company that specializes on technical security testing. The company has improved their project management a lot in the past years, but there is still a will to continue improving towards a more self-organizing and agile direction.

The goal of this research was to improve security testing projects from the employees' point of view so that daily work feels less taxing and stressful. This was tried to achieve by implementing elements of agile methods to the company's processes. The results of the research consisted of concrete changes in the execution of security testing projects. The research was defined to concern only the technical security testing projects in the target company.

This research was conducted as action research. Agile methods were researched, and their elements like self-organizing teams, daily meetings, retrospectives, and visualizing work, were identified from literature. The target company's employees' attitudes and opinions towards these agile elements and the present state of the company were surveyed through interviews. First improvements like implementing teams were planned out while considering the findings of the interview. When teams were implemented, their practices were agreed on together with all members of the teams. After this, the research continued following the action research spiral as the next steps were planning, implementing and assessing further improvements with the teams.

With the research, the processes related to technical security testing projects in the target company were changed significantly based on the employees' wishes and opinions. One of the most significant changes was implementing teams. In addition, teams decided about their own practices and regular meetings like daily meetings and retrospectives themselves. The development work will be continuing in the future following the action research spiral, and improvements will be planned based on the teams' experiences and opinions.

Keywords: Agile methods, technical security testing, action research

ALKUSANAT

Vaikka työelämä vei mennessään jo hyvän aikaa ennen opintojen loppuun saattamista, on tämä diplomityö viimeinen silaus opiskeluvuosiini. Haluankin kiittää kohdeyritystä siitä, että sain valita työni aiheen hyvin vapaasti itseäni kiinnostavasta aiheesta ja myös siitä joustavuudesta, jonka ansiosta työ oli helppo toteuttaa muun työnteon ohella. Kiitos tietysti myös tutkimukseen osallistuneille työkavereille ja muutenkin koko työyhteisölle, teidän kanssanne on todella mukava tehdä töitä!

Koulun suunnalta haluan kiittää työni tarkastajia ja ennen kaikkea työn ohjaajaa Pasi Hellsteniä ohjauksesta ja niin ikään joustavuudesta. Opiskeluajoista on jäänyt paljon hyviä muistoja ja tärkeitä ihmissuhteita, mutta näistä tärkeimmät liittyvät TEA-clubiin ja Tietojohtajakilta Man@gerin hallitukseen vuodelta 2014. Kiitos että teitte opiskeluajoistani mahtavaa aikaa! Lopuksi vielä erityiskiitos perheelleni, teiltä saatu tsemppaus on ollut korvaamaton tuki koko opintojen ajan.

Espoossa, 29.4.2020

Ida Myller

SISÄLLYSLUETTELO

1 JOHDANTO

Teknologian kehittyessä laitteista tulee entistä nopeampia ja ihmiset ovat laitteiden ja sovellusten kanssa tekemisissä entistä enemmän (Cohen ym., 2004). Sovellukset ja ohjelmistot ovat kietoutuneet vahvasti osaksi ihmisten päivittäistä elämää ja ne ovat kaikkialla, sillä esimerkiksi lentoyhtiöt, pankit, sairaalat, ruokakaupat ja monet muut tahot ovat niistä riippuvaisia (Merkow & Raghavan, 2010). Tämän myötä myös sovelluskehityksen tahti kiihtyy, kun se pyrkii vastaamaan nopeasti muuttuvan toimintaympäristön tarpeisiin (Cohen ym., 2004). Tämä onkin johtanut ketterien menetelmien yleistymiseen sovelluskehityksessä, sillä ne on kehitetty vastaamaan alati muuttuvan ja vaikeasti ennustettavan toimintaympäristön haasteisiin (Dingsøyr ym., 2010). Vaikka ketterät menetelmät ovat erityisen suosittuja sovelluskehityksessä, myös muilla aloilla on kiinnostusta niiden soveltamiseen muuttuvan toimintaympäristön tarpeisiin vastaamiseksi (Ashmore & Runyan, 2015).

Koska sovellukset ovat kaikkialla, niin ovat myös niissä esiintyvät virheet ja haavoittuvuudet, jotka voivat vaarantaa sovellusten turvallisuuden esimerkiksi jos ne antavat hyökkääjälle mahdollisuuden tehdä jotain vahingollista, kuten paljastaa tai muokata arkaluonteista tietoa tai häiritä järjestelmän toimintaa (Dowd ym., 2007, s. 4; Merkow & Raghavan, 2010). Sekä Whitakerin ja Newmanin (2005), että Weidmanin (2014, s. 2) mukaan paras tapa puolustautua on testaamalla sovellusta hyökkääjän näkökulmasta. Tällaisella tietoturvatestauksella hyökkääjälle hyödylliset haavoittuvuudet voidaan löytää ja korjata ennen kuin tämä ehtii löytää ja hyödyntää niitä.

Ketterien menetelmien myötä sovelluskehityksessä ei enää suunnitella asioita kattavasti etukäteen, minkä seurauksena myös tietoturvallisuuden suunnittelu kehityksen kohteena olevaan sovellukseen liittyen jää aiempaa vähemmälle (Burton-Spall ym., 2017). Ilman asianmukaisia tietoturvakäytäntöjä ketterien tiimien tuottamat sovellukset voivat olla tietoturvan näkökulmasta haavoittuvaisia (Ghani ym., 2014). Jos ketterissä tiimeissä ei ole tarpeeksi ymmärrystä kehitettävää sovellusta koskevista tietoturvauhista ja riskeistä, joita kehityksessä otetaan tietoturvan näkökulmasta, niitä ei voida kontrolloida riittävästi (Burton-Spall ym., 2017). Tämän seurauksena sovellusten asianmukainen tietoturvatestaus on entistä tärkeämpää.

Koska sovelluskehitystä tehdään monissa yrityksissä ketterästi, kehitysprosessin muutokset heijastuvat myös muun muassa ulkopuolisen yrityksen toteuttamaan tietoturvatestaukseen. Tämän vuoksi tässä tutkimuksessa selvitetään ketterien menetelmien elementtien sopivuutta tietoturvatestauksen kontekstiin ja erityisesti kohdeyrityksen tietoturvatestausprojektien toteuttamiseen, että yrityksen olisi helpompi tarjota palveluitaan ketterästi toimiville asiakkailleen oikea-aikaisesti toimintaympäristön muutoksista huolimatta.

1.1 Tutkimuksen tausta

Ketterät menetelmät ovat syntyneet vastaamaan nopeasti muuttuvan ja vaikeasti ennustettavan toimintaympäristön tarpeisiin (Dingsøyr ym., 2010). Useimmissa sovelluskehitystä tekevissä yrityksissä onkin siirrytty perinteisistä projektinhallintamenetelmistä ketteriin menetelmiin, joista suosituin on Scrum (Chóliz ym., 2015). Siinä missä perinteisiä projektinhallintamenetelmiä soveltavat projektit etenevät lineaarisesti projektin aluksi määritellyn suunnitelman mukaan, ketterissä menetelmissä työ suunnitellaan ja toteutetaan pala kerrallaan niin, että projektin kulkua voidaan ohjata projektin edetessä. Lisäksi ketterät menetelmät tunnistavat ihmisen projektin keskeisenä onnistumisen edellytyksenä ja mahdollistajana (Abrahamsson ym., 2002).

Vaikka ketterät menetelmät onkin kehitetty ohjelmistokehityksen tarpeisiin, ovat myös muut alat kiinnostuneet ketterien arvojen ja periaatteiden yksinkertaisuudesta ja käytännönläheisyydestä (Ashmore & Runyan, 2015). Koska sovelluskehitystä tehdään monissa yrityksissä ketterästi, kehitysprosessin vastatessa toimintaympäristön muutoksiin prosessin muutokset heijastuvat myös muun muassa ulkopuolisen yrityksen toteuttamaan tietoturvatestaukseen. Jotta tietoturvatestaus voidaan toteuttaa oikea-aikaisesti sitä tarvitseville sovelluksille, on testausta toteuttavan tahon voitava vastata nopeasti sovelluskehityksen muutoksiin.

Tekninen tietoturvatestaus on erityislaatuinen projektityyppi, joka poikkeaa tyypillisistä sovelluskehitysprojekteista usein eri tavoin. Testausprojektien erityispiirteisiin kuuluu muun muassa projektien verrattain lyhyt kesto, sillä esimerkiksi kohdeyrityksessä teknisen tietoturvatestausprojektin testaus- ja raportointivaiheet kestävät tyypillisesti vain yhdestä kahteen viikkoa. Koska projektit ovat hyvin lyhyitä, pienetkin viivästykset ja aikataulumuutokset vaikuttavat projektien toteuttamiseen merkittävästi ja tämän myötä projektijohtamisen merkitys korostuu. Esimerkiksi jos yksi projekti siirtyy, tilalle on saatava toinen projekti tai työntekijöillä ei ole laskutettavaa työtä. Vastaavasti siirtyvän projektin uuden toteutusajan kohdalla voi kalenterissa olla jo varaus toiselle projektille, ja tälle ajalle tarvitaan kaksinkertaisesti resursseja projektien toteuttamiseksi. Koska muutoksia

tapahtuu paljon ja niihin tulee voida reagoida nopeasti, ketterät menetelmät ovat potentiaalinen vaihtoehto muutoksiin vastaamiseksi.

Measey ym. (2015, s. 39) mukaan ei ole olemassa yhtä ketterää viitekehystä, joka sopisi kaikille organisaatioille, vaan sopiva ratkaisu riippuu liiketoimintakontekstista, johon ketterä ratkaisu halutaan implementoida. Usein sopiva kokonaisratkaisu löytyy yhdistelemällä useaa eri menetelmää, vaikka lähtökohtana olisikin yksi ketterä menetelmä (Project Management Institute, 2017, s. 31). Empiirinen tutkimus on myös osoittanut, että sopiva tapa hallita projektia riippuu projektista itsestään ja että erilaiset projektityypit ja kontekstit vaativat erilaisia lähestymistapoja projektin onnistumiseksi (Ahimbisibwe ym., 2017). Tämän myötä samankaltaisia projekteja voi olla tarpeellista hallita eri yrityksissä erilaisin menetelmin erilaisesta kontekstista johtuen. Vastaavasti kaksi erilaista projektia samassa yrityksessä voi vaatia erilaisen lähestymistavan projektien erityispiirteiden vuoksi.

1.2 Tutkimuksen kohde

Tämän tutkimuksen kohteena on pieni suomalainen kyberturvallisuusalan yritys, joka tarjoaa organisaatioille räätälöityjä tietoturvapalveluita kuten tietoturvatestauksia, henkilöstön tietoturvakoulutusta ja tietoturvakonsultointia. Yritys keskittyy tekniseen tietoturvatestaukseen, jossa asiakkaan järjestelmiin tehdään kontrolloituja koehyökkäyksiä ja niiden teknisiä turvallisuusmekanismeja arvioidaan hyökkääjän näkökulmasta.

Kohdeyrityksen sisällä on käyty paljon keskustelua projektien toteuttamisesta ja toiminnan kehittämisestä. Uuden johtoryhmän aloitettua keväällä 2018 projektinhallintaan on kiinnitetty paljon huomiota ja sitä on parannettu muun muassa projektinhallinnan kehittämiseksi teetetyn tutkimuksen tulosten perusteella. Tutkimuksen yhteydessä yritykseen implementoitiin uusi rooli, projektipäällikkö, ja tyypillisten testausprojektien vaiheita ja vastuita selvennettiin ja dokumentoitiin.

Tällä hetkellä yrityksessä toteutetaan projekteja vesiputousmallia mukaillen lineaarisesti. Projektien etenemistä visualisoidaan Kanban-taulua hyödyntäen, mutta se toimii pääasiassa visualisointitarkoituksiin eikä esimerkiksi Kanban-menetelmään kuuluvia keskeneräisen työn rajoja ole käytössä lainkaan. Aikataulut ovat tiukkoja ja heti edellisen projektin päätyttyä alkaa uuden projektin testaus. Tulevien projektien aikataulutuksesta vastaa yksin Chief Operative Officer, joka allokoi työntekijät projekteille myyjiltä saamiensa tietojen, kuten projektin arvioidun keston ja erityisosaamisvaatimusten mukaan.

Projektien lineaarisessa toteuttamisessa on havaittu ongelmia, sillä projektien viivästyminen teknisten ongelmien tai esimerkiksi työntekijän sairastumisen takia kertautuu nopeasti ja aiheuttaa päivittäisessä työskentelyssä stressin ja kiireen tunnetta. Tästä muodostuu tutkimusongelma, johon tässä tutkimuksessa pyritään löytämään ratkaisu. Testaajat ovat toivoneet, että työtä voitaisiin tehdä vahvemmin tiimeissä yksittäisten projekteihin allokoitujen testaajien sijaan. Ideaan usean projektin toteuttamisesta itseohjautuvasti tiimissä on suhtauduttu työntekijöiden keskuudessa positiivisesti.

1.3 Tutkimuksen tavoitteet ja rajaukset

Kohdeyrityksen projektien toteutusta halutaan muuttaa ketterämmäksi ja itseohjautuvammaksi. Yrityksessä halutaan uskoa, että testaajat ovat parhaita ammattilaisia oman työnsä jäsentelyyn ja hallitsemiseen. Kohdeyrityksen projektien erityispiirteiden vuoksi oletetaan, että mikään ketterä menetelmä kuten Scrum ei suoraan sovellu yrityksen käyttöön.

Project Management Instituten (2017) mukaan erityistä ketterää menetelmää ei tarvitse ottaa käyttöön, jos vain noudatetaan ketteriä arvoja ja käytössä oleva toimintatapa edistää kommunikaatiota asiakkaan kanssa. Yhden menetelmän valitsemisen sijaan halutaan selvittää, millaisia elementtejä ketterät menetelmät sisältävät ja mitkä elementit olisivat hyödyllisiä kohdeyrityksen kontekstissa. Näistä hyödylliseksi arvioiduista elementeistä halutaan muodostaa yritykselle räätälöity menetelmä, joka sopii sen projektien toteutukseen hyvin. Räätälöidyn menetelmän toivotaan sujuvoittavan yrityksen projektien toteutusta ja vähentävän stressin ja kiireen tunnetta työntekijöillä.

Abrahamsson ym. (2002) mukaan yksi menetelmä ei voi toimia kaikenlaisille projekteille, vaan kullekin projektityypille tulisi valita sopiva menetelmä. Tästä syystä tutkimus rajataan koskemaan vain yrityksessä toteutettavia teknisiä tietoturvatestausprojekteja. Nämä projektit on valittu siksi, että ne muodostavat valtaosan yrityksen liikevaihdosta ja henkilötyömäärästä. Näiden projektien parantamisella uskotaan olevan suurin vaikutus yrityksen maineeseen ja kannattavuuteen, sekä työntekijöiden hyvinvointiin ja tyytyväisyyteen. Tutkittavat menetelmät rajataan ketteriin menetelmiin, sillä ne ovat herättäneet kiinnostusta yrityksen henkilöstön keskuudessa ja yrityksessä koetaan, että projektien toteutuksen ketteryyden lisääminen voisi parantaa yrityksen työntekijöiden arkea.

Tutkimuksen tavoitteena on parantaa tietoturvatestausprojektien toteuttamista työntekijöiden näkökulmasta niin, että päivittäinen työnteko tuntuu vähemmän kiireiseltä ja stressaavalta. Näin pyritään ratkaisemaan tutkimusongelma, joka määriteltiin luvun 1.2 lo-

pussa. Tämä pyritään saavuttamaan implementoimalla kohdeyritykseen elementtejä ketteristä menetelmistä. Tutkimuksen tuloksena tehdään konkreettisia muutoksia tietoturvatestausprojektien toteuttamiseen. Tavoitteen saavuttamiseksi tutkimus pyrkii vastaamaan seuraavaan päätutkimuskysymykseen:

- Miten kohdeyrityksen tietoturvatestausprojektien toteuttamista voidaan parantaa työntekijöiden näkökulmasta ketterien menetelmien elementein?

Koska päätutkimuskysymys on laaja, siihen muodostettaan vastaus kahden alatutkimuskysymyksen avulla:

- Millaisia ketteriä menetelmiä on olemassa ja mitä elementtejä niissä on?
- Mitä mieltä kohdeyrityksen työntekijät ovat tietoturvatestausprojektien nykytilasta ja niiden kehittämisestä?

Ensimmäiseen alatutkimuskysymykseen etsitään vastausta kirjallisuudesta tutkimalla ketteriä menetelmiä ja selvittämällä, mitä elementtejä näissä on tunnistettu. Toiseen alatutkimuskysymykseen vastatessa selvitetään yksilöhaastattelujen kautta, mitä mieltä kohdeyrityksen työntekijät ovat tietoturvatestausprojekteista nykyään ja miten he toivoisivat niitä kehitettävän. Samalla selvitetään, millaisia asenteita työntekijöillä on erilaisten ketterien elementtien käyttöönottoon kohdeyrityksessä. Päätutkimuskysymykseen muodostetaan vastaus alatutkimuskysymysten vastausten ja ryhmähaastattelujen perusteella.

1.4 Tutkimuksen rakenne ja vaiheet

Johdantokappaleessa esitellään tutkimus, sen tausta ja kohdeyritys, sekä tutkimuksen tavoitteet. Seuraavassa kappaleessa esitellään tutkimuksen lähestymistapa, toimintatutkimus ja haastattelututkimus teoreettisesti, sekä kuvataan tutkimuksen empiirisen osuuden toteutus. Kolmannessa kappaleessa esitellään tarkemmin kohdeyrityksen kontekstia kuvaamalla teoriaan tukeutumalla kohdeyrityksen teknisten tietoturvatestausprojektien toteuttamista ja niiden projektinhallintaa. Seuraavat kaksi kappaletta käsittelevät tutkimuksen teoriaa, ja näissä kappaleissa esitellään ketteriä menetelmiä ja niissä tunnistettuja elementtejä teoreettisesti. Kappaleessa kuusi esitellään yksilöhaastatteluiden löydökset mukaillen samaa rakennetta kuin ketterien menetelmien elementtejä esitellessä ja kappaleessa seitsemän kuvataan, kuinka toimintaa kehitettiin ryhmähaastatteluissa keskustellen ja suunnitellen. Viimeisessä luvussa vedetään yhteen tutkimus ja vastataan johdannossa esiteltyihin tutkimuskysymyksiin. Lisäksi arvioidaan tutkimusta ja pohditaan, mitä tulevaisuuden tutkimuskohteita tutkimuksen aikana on noussut esiin.

2 TUTKIMUKSEN TOTEUTUS JA MENETELMÄT

Tässä kappaleessa esitetään tutkimuksen toteutus ja menetelmät, sekä niiden taustalla vaikuttava tieteenfilosofinen lähestymistapa, joka vaikuttaa tutkimukseen määrittelemällä tutkimuksessa tehtäviä oletuksia. Tutkimuksen lähestymistapana on interpretivismi, joka korostaa subjektiivisuutta ja olettaa, että jokainen tulkitsee omaa ja muiden ihmisten roolia sosiaalisessa toimintaympäristössä omien näkemystensä mukaan. Tutkimuksessa kerättävä aineisto on laadullista ja tutkimus on tyypiltään tapaustutkimus, joka toteutetaan toimintatutkimuksena. Tutkimuksen empiirinen osuus toteutetaan haastattelututkimuksen menetelmin, ja tutkimuksessa hyödynnetään sekä yksilö- että ryhmähaastatteluja.

2.1 Tutkimuksen lähestymistapa

Tutkimuksessa käytettävä tieteenfilosofinen lähestymistapa sisältää tärkeitä oletuksia siitä, millä tavalla maailma nähdään tutkimuksen näkökulmasta. Nämä oletukset heijastuvat tutkimusstrategiaan ja -menetelmiin. Tieteenfilosofisen lähestymistavan valintaan vaikuttaa osaltaan käytännön asiat, mutta suurin valintaan vaikuttava tekijä on tutkijan näkemys tiedon ja tiedonluontiprosessin välisestä suhteesta. (Saunders ym., 2009, s. 108.) Koska tutkija itse näkee kohdeyrityksen toiminnan erittäin henkilökeskeisenä ja uskoo työyhteisön huomioinnin ja hyvinvoinnin näyttelevän merkittävää roolia organisaatioiden menestyksessä, sosiaalisia elementtejä kuten tunteita ja asenteita huomioonottava tieteenfilosofinen lähestymistapa oli tässä tutkimuksessa luonnollinen valinta.

Tämän tutkimuksen tieteenfilosofinen lähestymistapa on interpretivismi, joka edustaa näkemystä, jossa tutkijan on ymmärrettävä ihmisten välisiä eroja rooleissaan sosiaalisina toimijoina. Sen mukaan ihmiset tulkitsevat niin omia kuin muidenkin ihmisten rooleja jokapäiväisessä elämässä omien näkemystensä mukaan. (Saunders ym., 2009, ss. 115–116.) Interpretivismi juontaa juurensa hermeneutiikasta ja fenomenologiasta (Eriksson & Kovalainen, 2008). Coghlanin ja Brydon-Millerin mukaan (2014) hermeneutiikassa pyritään kuvailemaan ja tulkitsemaan ihmisen toimintaa, ja tässäkin tutkimuksessa voidaan tunnistaa interpretivismin lisäksi hermeneuttisia piirteitä, sillä tulkintojen tekeminen nähdään välttämättömäksi osaksi tutkimusprosessia. Fenomenologiassa tutkimus taas perustuu Millsin ja Birksin (2014) mukaan omien tai muiden ihmisten kokemusten kautta tuotettavaan tietoon pohdiskelevalla otteella ja tutkimusta lähestytään avoimesti ilman ennakko-oletuksia tai teoreettista viitekehystä.

Interpretivistisessä lähestymistavassa on tärkeää, että tutkija suhtautuu empaattisesti ja pyrkii ymmärtämään tutkittavien henkilöiden maailmaa heidän näkökulmastaan. Interpretivistisen näkökulman voidaan nähdä sopivan erittäin hyvin esimerkiksi liiketoimintaympäristön tutkimukseen, sillä liiketoimintatilanteet ovat sekä monimutkaisia että ainutlaatuisia. (Saunders ym., 2009, ss. 115–116.) Interpretivismi on näiden väitteiden pohjalta luonnollinen valinta tutkimuksen lähestymistavaksi, sillä tutkijalla on empaattisia taipumuksia ja tutkimuksen kohde sijoittuu liiketoimintaympäristöön.

Benbasat ym. (1987) mukaan sekä tutkimusmenetelmät että tutkimuksessa kerättävä aineisto valitaan tutkimuskysymysten perusteella. Tutkimuksen kohdeyritykseen vahvasti kietoutuviin tutkimuskysymyksiin vastaamiseksi tämä tutkimus on tyypiltään tapaustutkimus, jossa kohdetta tutkitaan empiirisesti todellisessa elämässä (Benbasat ym., 1987; Saunders ym., 2009, ss. 145–146). Erikssonin ja Kovalaisen (2008) mukaan tapaustutkimuksen rajat ovat usein häilyvät, ja esimerkiksi tässä tutkimuksessa vaikutukset heijastuvat koko kohdeyritykseen, vaikka tutkimuksen kohteena on tietoturvatestausprojektien toteuttaminen. Interpretivististä lähestymistapaa noudattavissa tutkimuksissa tietoa kerätään usein laadullisesti, pienistä otoksista ja syvällisesti tutkien (Saunders ym., 2009, s. 119). Tämäkin tutkimus on laadullinen, sillä sekä kerättävä tutkimusaineisto että sen tulkinta on laadullista, eli ei-numeerista (Eskola & Suoranta, 1998).

Ontologia koskee todellisuuden luonnetta ja tutkijan tekemiä oletuksia siitä, millä tavalla maailma toimii (Saunders ym., 2009, s. 110). Useissa laadullisissa tutkimuksen lähestymistavoissa ontologiana on subjektivismi (Eriksson & Kovalainen, 2008). Subjektivismissa nähdään, että sosiaaliset ilmiöt luodaan sosiaalisten toimijoiden näkemysten ja niistä seuraavien tekojen kautta ja tämän prosessin toistuessa ilmiöt ovat jatkuvassa muutoksessa. Esimerkiksi organisaatiokulttuuri nähdään asiana, joka organisaatio itse on ja joka kehittyy jatkuvasti. (Saunders ym., 2009, ss. 110–111.) Subjektivismissa on olennaista myös se, että eri henkilöt voivat nähdä ja kokea samat asiat eri tavalla (Eriksson & Kovalainen, 2008). Tässä tutkimuksessa oletetaan, että jokainen työyhteisön jäsen tulkitsee työympäristöään omalla tavallaan, ja että työyhteisö rakentuu ja muuttuu sen jäsenten asenteiden ja toiminnan kautta.

Interpretivistisessä tutkimuksessa aineistoksi hyväksytään subjektiivisesti käsiteltävä tieto kuten tunteet ja asenteet, eli sosiaaliset ilmiöt, joita ei voida nähdä, mitata ja muokata kuten koneita tai muita fyysisiä laitteita. Interpretivismin mukaan liiketoimintaympäristö on aivan liian monimutkainen soveltuakseen ehdottomin laein teorioitavaksi luonnontieteiden kaltaisesti, ja teoreettisten lakien muodostaminen johtaisi monimutkaisen liiketoimintaympäristön liialliseen yleistämiseen ja syvän ymmärryksen menettämiseen.

(Saunders ym., 2009, ss. 112–116.) Tässä tutkimuksessa kerätään aineistoa kohdeyrityksen työntekijöiden mielipiteistä ja asenteista yrityksen nykyisiä käytäntöjä ja niiden kehittämistä koskien.

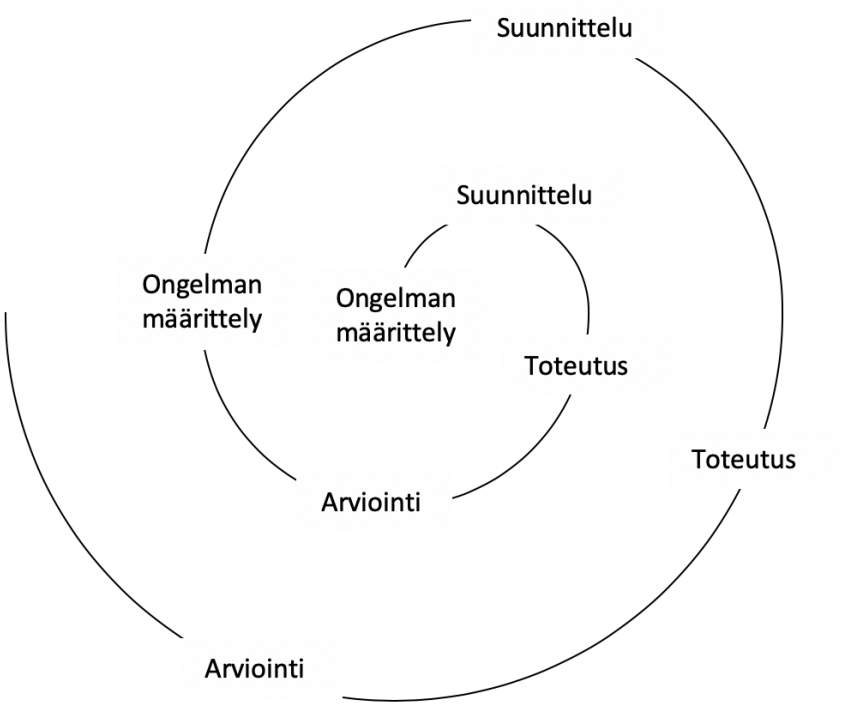
2.2 Toimintatutkimus

Tämä tutkimus toteutetaan toimintatutkimuksena, jossa pyritään vaikuttamaan tutkittavaan kohteeseen ja tukemaan muutosta kohdeyrityksessä parantamalla sosiaalisia käytäntöjä (Saunders ym., 2009, s. 147; Toikko & Rantanen, 2009, s. 30). Toimintatutkimuksen keskeisiä piirteitä ovat käytännönläheisyys ja ongelmakeskeisyys, ja tutkittava kohde on aina jokin tietty yhteisö (Davison ym., 2004; Eskola & Suoranta, 1998). Tässä tutkimuksessa tämä yhteisö on kohdeyrityksen työntekijät, jotka osallistuvat tietoturvatestausprojektien toteuttamiseen. Eskolan ja Suorannan (1998) mukaan toimintatutkimuksessa ei pyritä objektiivisuuteen, sillä tutkija on osa tutkimuskohdetta. Tässäkin tutkimuksessa tutkija on osa kohdeyrityksen henkilöstöä ja teknisten tietoturvatestausprojektien toteutusta, eli tutkittavan yhteisön jäsen.

Kohdeyrityksen henkilöstö synnyttää tutkimustuloksia tutkimuksen aiheesta, joka kiinnostaa heitä aidosti (Saunders ym., 2009, s. 147). Tämän tutkimuksen kohdalla tutkittavan yhteisön aito kiinnostus syntyy siitä, että tutkimuksessa pyritään parantamaan tietoturvatestausprojektien toteuttamista työntekijöiden näkökulmasta, eli muuttamaan työn tekemistä tutkittavalle yhteisölle mielekkäämmäksi. Saunders ym. (2009, s. 148) mukaan työntekijöiden osallistaminen tutkimukseen on tärkeää, koska muutosta tapahtuu todennäköisemmin silloin, kun sen luomiseen on itse voitu vaikuttaa. Tässä tutkimuksessa työntekijät osallistetaan tutkimukseen mahdollisimman monessa vaiheessa muun muassa selvittämällä heidän asenteensa ja mielipiteensä yrityksen nykytilasta, käyttämällä näitä löydöksiä pohjana muutosten suunnitteluun ja tekemällä päätökset työntekijöiden päivittäisen toiminnan kehittämisestä yhdessä heidän kanssaan. Tärkeimpänä muutosta edistävänä tekijänä nähdään se, että työntekijät pääsevät itse sopimaan heitä koskevista muutoksista.

Toimintatutkimukselle tyypillistä on sen spiraalimaisuus, eli prosessi jossa määritellään ongelma, suunnitellaan kehitystoimenpiteitä, toteutetaan suunnitellut toimenpiteet ja arvioidaan niitä pyörii tutkittavavan kohteen ympärillä ja tutkimuksen aikana saatavaa tietoa käytetään seuraavilla kierroksilla kehittämistoiminnan ohjaamisessa (Saunders ym., 2009, s. 147; Toikko & Rantanen, 2009, s. 115). Toimintatutkimuksen spiraalin ensimmäinen kehä muodostaa lähtökohdan, jonka pohjalta toimintaa voidaan lähteä kehittämään (Toikko & Rantanen, 2009, s. 67). Ongelman määrittelyssä tarkastellaan tutkimus-

kohteen tilaa tunnistaen siitä ratkaistava ongelma, jonka pohjalta suunnitellaan ja toteutetaan toimenpiteitä, joita sitten arvioidaan ja siirrytään uuteen kierrokseen tarkastelemaan tutkimuskohteen uutta tilaa uusien toimenpiteiden suunnittelemiseksi. Toimintatutkimuksen spiraali on esitetty kuvassa 1.



Kuva 1. *Toimintatutkimuksen spiraali, mukaillen (Saunders ym., 2009, s. 148)*

Tässä tutkimuksessa ensimmäinen ongelman määrittely tapahtui määrittelemällä tutkimusongelma johtoryhmän kanssa. Tämän perusteella tutustuttiin alan kirjallisuuteen ja suunniteltiin yksilöhaastattelut nykytilan ja työntekijöiden asenteiden selvittämiseksi. Haastattelujen toteutuksen jälkeen niiden löydöksiä arvioitiin johtoryhmän kanssa ja niiden perusteella tarkennettiin ratkaistavaa ongelmaa. Johtoryhmä päätti implementoida tiimit ja laati niiden toiminnalle reunaehdot, joiden puitteissa tiimit sopivat toiminnastaan tarkemmin toisella suunnittelukierroksella. Tiimien toiminnan toteutusta arvioitiin tiimien kanssa yhdessä noin kuukauden kuluttua tiimitoiminnan aloittamisesta ja määriteltiin uudelleen, mitä ongelmia halutaan tulevaisuudessa ratkaista. Näiden ongelmien ratkaisuun suunniteltiin jälleen toimenpiteitä, joiden toteutus aloitettiin suunnittelun päätteeksi.

2.3 Haastattelututkimus

Tämän tutkimuksen empiirinen osuus toteutetaan haastattelututkimuksena. Haastattelulla tarkoitetaan tarkoituksellista keskustelua kahden tai useamman henkilön välillä (Saunders ym., 2009, s. 318). Haastattelutyypit voidaan jakaa ja nimetä usein eri tavoin,

muun muassa haastattelukysymysten muotoilun kiinteyden mukaan (Eskola & Suoranta, 1998). Tällä jaolla haastattelun tyyppi asettuu välille, joka kulkee yhdenmukaista haastatteluru nkoa noudattavasta strukturoidusta haastattelusta strukturoimattomaan ja enemmän vapaata keskustelua muistuttavaan haastatteluun (Saunders ym., 2009, s. 320). Strukturoitu haastattelu soveltuu Johnsonin (2017, s. 81) mukaan tutkimukseen, jossa analysoidaan määrällistä aineistoa. Laadullisessa tutkimuksessa sopiva valinta on strukturoimattomampi haastattelu, jonka avulla kerättävä aineisto on laadullisempaa. Esimerkiksi Saundersin ym. (2009, s. 321) mukaan strukturoimaton haastattelu sopii tilanteisiin, joissa halutaan tutkia tiettyä aihealuetta syvällisesti. Johnsonin (2017, s. 82) mukaan näiden kahden ääripään välimuoto on laadullisessa tutkimuksessa kuitenkin suosituin haastattelutyyppi.

Strukturoidussa haastattelussa käytetään yhdenmukaista listaa kysymyksistä, jotka esitetään kaikille haastateltaville samalla tavoin ja joilla kerätään usein määrällistä dataa laadullisen sijaan (Johnson, 2017, s. 81; Saunders ym., 2009, s. 320). Eskolan ja Suorannan (1998) mukaan myös vastaukset valitaan valmiista vaihtoehdoista, ja haastattelu on toiselta nimeltään lomakehaastattelu. Strukturoimattomassa haastattelussa ennalta määriteltyä listaa haastattelukysymyksistä ei ole, vaan haastateltavalle annetaan mahdollisuus puhua käsiteltävästä aiheesta vapaasti.

Eskolan ja Suorannan (1998) mukaan edellä esiteltyjen kahden ääripään väliin asettuu kaksi haastattelutyyppiä, jotka ovat puolistrukturoitu haastattelu ja teemahaastattelu. Puolistrukturoitu haastattelu muistuttaa paljon strukturoitua haastattelua, mutta valmiita vastausvaihtoehtoja ei ole, vaan haastateltava saa vastata kysymyksiin vapaasti. Teemahaastattelussa taas tutkijalla on haastattelutyypin nimen mukaisesti etukäteen valmisteltu lista teemoista ja kysymyksistä, joita haastattelussa halutaan käydä läpi, mutta ne voivat muuttua haastattelusta toiseen (Saunders ym., 2009, s. 320). Haastattelun kulusta riippuen kysymyksiä voidaan käydä läpi eri järjestyksessä, eikä niitä ole muotoiltu tarkasti etukäteen (Eskola & Suoranta, 1998). Haastattelija voi esittää lisäkysymyksiä tarvittaessa, esimerkiksi kun halutaan haastateltavan selittävän vastaustaan tarkemmin tai kun haastateltavan vastauksen pohjalta halutaan kysyä aiheesta lisää. (Saunders ym., 2009, s. 320.)

Mahdollisuus lisäkysymyksiin sallii laajemman ja syvemmän ymmärryksen keräämisen haastatteluista, sillä niiden avulla voidaan esimerkiksi päätyä keskustelemaan aiheesta, jota haastateltava ei ole aiemmin ajatellut, mutta joka on silti tutkimukselle relevanttia (Johnson, 2017, s. 81). Lisäksi lisäkysymyksillä voidaan pyrkiä ymmärtämään paremmin haastateltavan sanoille ja ideoille asettamia merkityksiä. Tämä on erityisen tärkeää interpretivistisessä tutkimuksessa, jossa tulee pyrkiä ymmärtämään, miten haastateltava

kokee erilaisia ilmiöitä. Tällaisessa haastattelussa haastateltavan vuorovaikutus haastateltavan kanssa ja tapa kysyä kysymyksiä kuitenkin vaikuttavat haastatteluista saatavaan tutkimusaineistoon. (Saunders ym., 2009, s. 324.)

2.4 Empiirisen tutkimuksen toteuttaminen

Tutkimuksen empiirinen osuus aloitettiin selvittämällä kohdeyrityksen nykytilaa yrityksen työntekijöiden näkökulmasta sekä heidän asenteitaan yksilöhaastatteluilla, jotka asettuivat tyypiltään puolistrukturoidun ja teemahaastattelun väliin. Teemahaastattelun tapaan ennalta suunnitellut teemat käytiin läpi eri järjestyksessä haastattelun kulusta riippuen, mutta kuhunkin teemaan oli valmisteltu puolistrukturoidun haastattelutyypin mukaisesti kysymykset, joihin haastateltava sai vastata vapaasti. Lisäkysymyksiä esitettiin tarpeen vaatiessa. Yksilöhaastatteluihin suunniteltu haastattelurunko on esitetty liitteessä A.

Haastattelujen otanta tehtiin harkintaperusteisesti, eli haastateltavat valittiin itse asetettujen kriteerien mukaan. Haastatteluun valittiin ensisijaisesti ne yrityksen teknisten tietoturvatestausprojektien toteuttamiseen osallistuvat työntekijät, jotka ovat tehneet ohjelmistokehitystä. Tämä johtuu siitä, että ketterien menetelmien soveltaminen on ohjelmistokehityksessä suosittua ja näillä henkilöillä voi olla niistä kokemusta (Hoda ym., 2018). Yrityksen teknisistä tietoturvatestaajista haastateltiin puolet ja otantaa täydennettiin ohjelmistokehityskokemuksellisten testaajien lisäksi yrityksen kokeneimmilla testaajilla. Testaajien lisäksi haastateltiin Chief Operative Officer (COO), sillä hän on tähän asti vastannut yksin projektien aikataulutuksesta. Alun perin yksilöhaastatteluja oli tarkoitus pitää kuusi kappaletta, mutta haastattelujen edetessä mukaan kutsuttiin vielä yksi haastateltava, että aineisto saatiin saturoitumaan paremmin sen riittävyyden takaamiseksi (Eskola & Suoranta, 1998).

Haastattelut sovittiin lähettämällä kalenterikutsut saatetekstin kanssa haastateltaville. Tätä ennen haastatteluista tiedotettiin johtoryhmän toimesta kaikille työntekijöille. Haastattelun kysymyksiä ei toimitettu haastateltaville etukäteen, mutta johtoryhmän tiedotteessa ja kalenterikutsujen saatetekstissä kuvattiin haastattelun aihepiiriä niin, että haastateltavat tiesivät etukäteen, mitä aihepiiriä haastattelu koski. Haastattelut dokumentoitiin ensisijaisesti nauhoittamalla haastattelu haastateltavan luvalla.

Haastatteluista kerätty aineisto analysoitiin temaattisella analyysillä. Temaattisessa analyysissä tunnistetaan, analysoidaan ja raportoidaan säännönmukaisuuksia eli teemoja analysoitavasta datasta (Eskola & Suoranta, 1998). Haastatteluja analysoidessa haastattelujen nauhoitteet kuunneltiin ja niistä kirjoitettiin muistiinpanot. Muistiinpanojen pe-

rusteella haastattelun löydökset yhdisteltiin toisiinsa ja järjesteltiin teemoihin. Tämän jälkeen haastattelujen nauhoitteet kuunneltiin vielä uudelleen tarkentaen löydöksiä ja lisäten suoria lainauksia tukemaan niitä niin, että ne toimivat aineistoa kuvaavina esimerkkeinä.

Kohdeyrityksessä implementoitiin tiimijako vuoden 2020 alussa, ja tietoturvatestaukseen muodostettiin kaksi tiimiä, joista toinen keskittyy pelkästään projektimuotoiseen tietoturvatestaukseen ja toinen jatkuvaan testaukseen. Jatkuvan testauksen tiimin jäsenet auttavat toista testaustiimiä silloin kun heillä ei ole muuta tekemistä, sillä jatkuva testaus on palveluna vielä lanseerausvaiheessa eikä se vaadi tiimin jäsenten kaikkia resursseja. Tiimit ja niiden toiminnan reunaehdot määritteli yrityksen johtoryhmä. Johtoryhmä käytti reunaehtojen suunnittelussa päätöksenteon tukena tämän tutkimuksen yksilöhaastattelujen löydöksiä ja ketterien menetelmien teoria-aineiston tarjoamaa tietoa.

Tammikuun alussa tietoturvatestaustiimien jäsenet ja COO kokoontuivat sopimaan tiimien toiminnasta tarkemmin yhdessä. Tilaisuuteen kutsuttiin testaajien ja COO:n lisäksi tietoturvatestaustiimien projektipäälliköt sekä kolmas projektipäällikkö, joka on toiminut testausprojekteissa projektipäällikkönä ennen tiimijaon implementointia. Tilaisuus voidaan nähdä strukturoimattomana ryhmähaastatteluna, jossa aiheesta keskusteltiin vapaasti osallistujien kesken. Tilaisuudessa keskityttiin erityisesti projektimuotoisen testauksen toimintatavoista keskusteluun ja sopimiseen yksilöhaastatteluissa esiin nousseiden löydösten pohjalta.

Helmikuussa, noin kuukauden kuluttua edellisestä ryhmähaastattelusta, järjestettiin uusi tilaisuus, jossa keskusteltiin tiimin toiminnasta edellisen kuukauden aikana. Tämäkin tilaisuus oli strukturoimaton ryhmähaastattelu. Tiimi keskusteli siitä, mikä toiminnassa on sujunut ja mitä tulee vielä kehittää. Tilaisuudessa myös käytiin uudelleen läpi johtoryhmän asettamat reunaehdot ja keskusteltiin niiden täyttämisestä. Ryhmähaastatteluja ei nauhoitettu, vaan niistä kirjoitettiin pelkästään muistiinpanot.

3 TIETOTURVATESTAUSPROJEKTIT

Tässä kappaleessa esitellään kohdeyrityksen konteksti tutkimuksen rajauksen puitteissa. Ensin esitellään tietoturvatestausta yleisesti, sillä se tarjoaa kontekstin tutkimuksessa käsiteltäville projekteille. Seuraavaksi esitellään yrityksen teknisissä tietoturvatestausprojekteissa sovellettavat projektinhallintamenetelmät, jotka ovat vesiputousmalli ja Lean-filosofian mukainen Kanban.

3.1 Tietoturvatestaus

Smith ym. (2009) mukaan menossa on tietoturvan pronssikausi, sillä vaikka tietotekniikan saralla on tehty suuria läpimurtoja kuten maailmanlaajuinen Internet, tietoturvan työkalut ovat vielä alkukantaisella tasolla. Nämä työkalut, kuten palomuurit ja salausmekanismit, ovat yleisesti kömpelöitä eivätkä toimi hyvin yhdessä. Täten hyökkääjä on tietoturvan osalta edelleen etulyöntiasemassa. Tietoturvaa hallitaan suureksi osaksi puolustuskannalta suojelemalla tietoja tunnetuilta ja ymmärretyiltä hyökkäyksiltä, esimerkiksi käyttämällä vahvaa salasanasuojausta. Tietoturvassa on kuitenkin kyse kokonaisuudesta ja vahva salasana ei yksin tarjoa riittävää suojaa, jos järjestelmässä on muita aukkoja, joita hyödyntämällä hyökkääjä voi tunkeutua järjestelmään murtamatta salasanasuojausta.

Ajoittain uutisoidaan tunnettuihin yrityksiin kohdistuneista tietoturvahyökkäyksistä. Weidmanin (2014, s. 2) mukaan teknisiin järjestelmiin kohdistuvissa hyökkäyksissä on harvoin hyödynnetty nollapäivähaavoittuvuuksia, eli haavoittuvuuksia, joille ei ole vielä olemassa korjausta, ja yleensä hyökkäys on onnistunut korjattavissa olevaa haavoittuvuutta hyödyntäen. Tunnettuja tietoturvahyökkäyksiä ovat esimerkiksi Lahden kaupungin verkkoon ja työasemiin kohdistunut hyökkäys (Lahden kaupunki, 2019), jossa haittaohjelma saatiin ujutettua palomuurien läpi kuormittamaan verkkoa ja saastuttamaan työasemia, sekä Yhdysvaltalaiseen luottotietolaitos Equifaxiin kohdistunut tietomurto (Equifax, 2017), jossa tunnettua haavoittuvuutta hyväksikäyttämällä vuodettiin yli 140 miljoonan ihmisen luottotiedot järjestelmästä. Haavoittuvuuden korjaava päivitys oli julkaistu noin kaksi kuukautta ennen hyökkäyksen alkamista. Verizonin (2019) yli 40 tuhanteen tietoturvaloukkaukseen perustuvan raportin mukaan suurin osa hyökkäyksistä on toteutettu yrityksen ulkopuolisten henkilöiden toimesta rahallisella motivaatiolla ja hyökkäysten havaitsemiseen menee tyypillisesti kuukausia tai jopa pidempään.

Haavoittuvuudella tarkoitetaan kohteessa olevaa heikkoutta, jota hyökkääjä voi hyödyntää. Näitä ovat esimerkiksi verkkosovelluksen kontekstissa erityiset virheet tai huomiotta jätetyt kohdat, jotka antavat hyökkääjälle mahdollisuuden tehdä jotain vahingollista, kuten paljastaa tai muokata arkaluonteista tietoa, häiritä järjestelmän toimintaa, ottaa järjestelmän haltuunsa tai jopa tuhota sen. (Dowd ym., 2007, s. 4.) Tietoturvatestauksessa etsitään näitä haavoittuvuuksia ennen kuin hyökkääjä löytää ne (Weidman, 2014, s. 2; Whitaker & Newman, 2005). Tällöin haavoittuvuudet voidaan ehtiä korjaamaan ennen kuin niiden olemassaolo johtaa onnistuneeseen tietoturvahyökkäykseen, jonka seurauksena esimerkiksi järjestelmän käyttäjien arkaluontoista tietoa voi vuotaa ulos järjestelmästä (Engebretson, 2013, s. 1). Kohdeyrityksessä asiakkaiden järjestelmiä testataan usein jo ennen kuin ne siirretään tuotantokäyttöön, ettei hyökkääjällä olisi edes mahdollisuutta löytää haavoittuvuuksia ensin.

Tietoturvatestaus voi toimia monessa eri roolissa kohdeorganisaation tarpeista riippuen. Sillä voidaan esimerkiksi havaita yleisiä ohjelmointivirheitä tai komponentteja, joista puuttuu uusimmat tietoturvapäivitykset. Tietoturvatestaus voidaan toteuttaa myös viimeisimpien tietoturvatoimenpiteiden tehokkuuden arvioimiseksi. Testaus voi myös paljastaa odottamattomia heikkouksia kohdeorganisaation tietoturvassa. Tietoturvatestaus voi palvella muitakin rooleja, mutta lopulta se tähtää aina varmistamaan, ettei kohteen turvallisuus vaarannu, sillä pelkästään puolustusmekanismit eivät tätä varmistusta voi tehdä. (Smith ym., 2009.) Kohdeyrityksessä tietoturvatestausta lähestytään hyökkääjän näkökulmasta. Järjestelmästä ja sen puolustusmekanismeista testataan niitä osia, jotka ovat hyökkääjän saatavilla, jos tämä pääsee testauksen laajuuden piirissä oleviin käyttäjärooleihin käsiksi. Tietoturvatestauksessa painotetaan sovellusten ominaisuuksien turvallisuuden testausta ja arviointia, eikä sisäisten tietoturvakontrollien kuten lokien tarkastelu kuulu yleensä testauksen laajuuden piiriin. Esimerkiksi lokeja tarkastellaan kuitenkin tapauskohtaisesti, jos siitä sovitaan asiakkaan kanssa testausta suunniteltaessa erikseen.

Tietoturvaa voidaan testata useilla eri menetelmillä, joihin kuuluu muun muassa haavoittuvuusskannaus, penetraatiotestaus ja tietoturva-auditointi (Smith ym., 2009). Tietoturvaa voidaan testata sekä manuaalisesti että erilaisten työkalujen kuten skannereiden avulla. Työkalujen hyödyntäminen on huomattavasti manuaalista testausta halvempaa, mutta työkalut eivät välttämättä löydä kaikkia manuaalisessa testauksessa tunnistettavia haavoittuvuuksia ja tulokset saattavat sisältää myös virheellisesti positiivisia tuloksia. (Antunes & Vieira, 2014.) Kohdeyrityksessä suurin osa tietoturvatestauksista toteutetaan penetraatiotestauksen menetelmin.

Penetraatiotestauksella tarkoitetaan toimintaa, jossa luotettu kolmas osapuoli hyökkää kohdeorganisaation järjestelmään tarkoituksenaan arvioida kohteen tietoturvan tasoa (Whitaker & Newman, 2005). Penetraatiotestauksessa simuloidaan todellisia hyökkäyksiä, jotta voidaan arvioida potentiaalisiin tietoturvaloukkauksiin liittyviä riskejä. Toisin kuin haavoittuvuusskannauksessa, penetraatiotestauksessa ei vain etsitä haavoittuvuuksia, joita hyökkääjät voisivat hyödyntää, vaan myös hyödynnetään niitä mahdollisuuksien mukaan, että voidaan arvioida, mitä hyökkääjät voivat haavoittuvuuksia hyödyntämällä saavuttaa. (Engebretson, 2013, s. 2; Smith ym., 2009; Weidman, 2014, s. 1.) Teknisen penetraatiotestauksen lisäksi voidaan tehdä myös fyysistä testausta, jossa tunkeudutaan asiakkaan toimitiloihin (Allsopp, 2009). Suurin osa kohdeyrityksen toteuttamista penetraatiotestausprojekteista on kuitenkin teknisiä, ja tämä tutkimus keskittyy vain teknisiin testausprojekteihin rajaten fyysisen penetraatiotestauksen tutkimuksen ulkopuolelle.

Penetraatiotestaaja on eettisesti toimiva henkilö, joka on palkattu hyökkäämään kohdejärjestelmään järjestelmän tietoturvallisuuden analysoimiseksi. Penetraatiotestaajan toimintaa rajoittavat ennalta sovitut säännöt, ja yleensä testaaja ei saa esimerkiksi suorittaa palvelunestohyökkäyksiä tai asentaa kohteeseen haittaohjelmia. Testauksen laajuus voi kuitenkin vaihdella merkittävästi kohdeorganisaation tarpeista riippuen. (Whitaker & Newman, 2005.) Kohdeyrityksessä yhden projektin puitteissa testataan tyypillisesti esimerkiksi verkkosovellus, mobiilisovellus, API-rajapinta tai näiden yhdistelmä. API-rajapinnalla tarkoitetaan rajapintaa, jonka kautta sovellus voi kommunikoida esimerkiksi toisen sovelluksen tai tietokannan kanssa. Silloin tällöin testauksen kohteena voi myös olla esimerkiksi fyysinen laite. Palvelunestohyökkäykset ja haittakoodi on rajattu lähes aina testauksen laajuuden ulkopuolelle, sillä ne voivat aiheuttaa vakavaa haittaa kohdeorganisaation tietojärjestelmille ja liiketoiminnalle. Esimerkiksi koodikatselmoinnin tekeminen on myös harvinaista, sillä kohdeyritys on erikoistunut penetraatiotestaukseen, ja työntekijöiden osaamisprofiili vastaa paremmin penetraatiotestauksessa kuin koodikatselmoinnissa tarvittavaa osaamista. Koodikatselmoinnissa nimittäin tarvitaan syvällisempää osaamista testattavassa kohteessa käytetyistä ohjelmointikielistä, joita on paljon erilaisia, ja joihin liittyvä osaaminen vaihtelee testaajalta toiseen merkittävästi muun muassa aiemman ohjelmointikokemuksen perusteella.

Testausta valmistellessa asiakkaalta tiedustellaan mahdollisia testattavaan järjestelmään kohdistuvia tietoturvavaatimuksia, joiden toteutumista arvioidaan testauksen yhteydessä. Tyypillisesti erityisiä vaatimuksia ei kuitenkaan ole tai niitä ei osata kommunikoida, ja tällöin arvioinnissa hyödynnetään muun muassa Open Web Application Security Projectin OWASP Top 10 -haavoittuvuuslistaa (OWASP, 2020), Common Vulnerabi-

lity Scoring Systemiä (CVSS) (FIRST.org, 2020) ja testaajan omaa ammattitaitoa ja harkintaa, jota tuetaan kouluttamalla heitä esimerkiksi SANS Instituten ja Offensive Securityn kursseilla. Nämä ovat alalla tunnettuja tietoturvakoulutuksia ja sertifikaatteja tarjoavia tahoja. GIAC Web Application Penetration Tester (GWAPT) on kohdeyrityksen testaajien yleisin SANSin tarjoama sertifikaatti, kun taas Offensive Securityn sertifikaateista kohdeyrityksestä löytyy esimerkiksi Offensive Security Certified Professional (OSCP).

Haavoittuvuuksien paljastamisen ja tietoturvatason arvioimisen lisäksi penetraatiotestaus voi myös auttaa kouluttamaan testauksen kohdeorganisaation johtoa, IT-osastoa ja sovelluskehittäjiä todellisen hyökkääjän järjestelmiin murtautumisen potentiaalisesti aiheuttamista seurauksista. Monissa järjestelmissä on väistämättä haavoittuvuuksia, joita ei voida korjata liiketoiminnallisista tai teknisistä syistä. Kun näiden haavoittuvuuksien olemassaolo ja hyödyntämismahdollisuudet on selvitetty penetraatiotestauksen avulla, voidaan mahdollisesti hyödyntää muita tietoturvatoimia haavoittuvuuksien hyödyntämisen ehkäisemiseksi. (Smith ym., 2009.) Tällainen tietoturvatoimi voi olla esimerkiksi Web Application Firewallin (WAF) asentaminen, joka hankaloittaa verkkosovelluksessa olevien haavoittuvuuksien hyödyntämistä tarjoamalla ylimääräisen suojakerroksen sovellukselle. Tämä voi ehkäistä joitain hyökkäyksiä, sillä hyökkääjän on tehtävä ylimääräistä työtä WAFin kiertämiseksi.

Tietoturvatestaus jaetaan yleisesti white box ja black box -testaukseen (Antunes & Vieira, 2014). Black box -testauksessa testaajalla ei ole aiempaa tietämystä testattavasta kohteesta, yrityksen prosesseista tai sen tuottamista palveluista (Muniz & Lakhani, 2013). Testaajalle voidaan antaa esimerkiksi vain kohteen URL- tai IP-osoite (Whitaker & Newman, 2005). Tällaisen testauksen onnistuminen vaatii paljon tiedustelutyötä, ja se vastaa lähimmin todellisen hyökkääjän lähtökohtaa silloin, kun hyökkääjä on organisaation ulkopuolinen taho. Tiedustelua toteutetaan muun muassa avoimia lähteitä käyttämällä (OSINT) ja kalastelukampanjoilla puhelimitse tai sähköpostitse (Tietosuojavaltuutetun toimisto, 2020; Weidman, 2014, s. 2). Tällaisia testausprojekteja tehdään kohdeyrityksessäkin, mutta hyvin harvoin, sillä white box -testauksen koetaan tuottavan asiakkaalle black box -testaukseen verrattuna enemmän arvoa sen vaatimiin resursseihin nähden.

White box -testauksessa testaajalla on laaja tietämys testattavasta kohteesta yksityiskohtia myöten (Muniz & Lakhani, 2013). Vaikka white box -testaus ei annakaan kaikista todenmukaisinta kuvaa ulkopuolisista hyökkäyksistä, se antaa tarkimman kuvan kohteen turvallisuudesta simuloiden pahinta mahdollista tapausta, jossa hyökkääjällä on täydellinen tietämys kohteestaan (Whitaker & Newman, 2005). Black box ja white box -testauksen väliin asettuu gray box -testaus, jossa testaajalla on perustiedot kohteesta, mutta

osa yrityksen sisäisestä tiedosta pidetään tältä salassa. Tyypillisesti testauksessa on käytössä esimerkiksi käyttäjätunnukset ja kuvaukset järjestelmän kaikista käyttäjätasoista, mutta esimerkiksi lähdekoodi ei ole testaajien saatavilla (Najera-Gutierrez & Ansari, 2018). Gray box -lähestymistapa on suosittu ammattilaisten toteuttamissa tietoturvatestauksissa, sillä se jäljittelee todellisten hyökkääjien toimintaa ja keskittyy tiedustelun sijaan haavoittuvuuksiin (Muniz & Lakhani, 2013). Whitaker ja Newmanin (2005) mukaan gray box -testauksella voidaan simuloida yrityksen sisäistä uhkaa, sillä usein testaajalle annetaan yrityksen työntekijän oikeuksia vastaavat käyttöoikeudet kohdejärjestelmään.

Kohdeyrityksen projektit vastaavat useimmiten gray box -testausta, sillä kohdejärjestelmästä on vain harvoin saatavilla täydellisiä tietoja. Skaalalla white box -testauksesta black box -testaukseen pyritään kuitenkin sijoittumaan mahdollisimman lähelle white box -testausta pyytämällä asiakkaalta käyttöliittymäesittely eli niin sanottu demo testattavasta kohteesta, pääsy järjestelmään esimerkiksi palomuuriavauksin tai VPN-yhteyden yli, kaikkia eri tasoisia käyttäjätunnuksia ja olemassa oleva dokumentaatio, joka voi sisältää esimerkiksi rajapinta-, prosessi- ja arkkitehtuurikuvauksia. Kohdeyrityksessä nähdään, että tämä tuo asiakkaalle paljon enemmän arvoa, sillä hyökkääjällä on periaatteessa rajoittamattomasti aikaa tutkia kohdettaan, kun taas tietoturvatestaus toteutetaan rajatussa ajassa. Kun kohteesta on kaikki mahdollinen tieto saatavilla, ei arvokasta testausaikaa tarvitse käyttää tiedusteluun, vaan kaikki aika voidaan hyödyntää olemassa olevien haavoittuvuuksien etsimiseen ja hyödyntämiseen, sekä tietoturvan tason arviointiin.

Käytännössä täydelliseen white box -tilanteeseen päästään testauksessa vain harvoin, sillä esimerkiksi testattavan sovelluksen lähdekoodia ei usein katselmoida testauksen yhteydessä, minkä vuoksi testaajalla ei ole white box -testauksen määritelmää vastaavasti täydellistä tietämystä testattavasta kohteesta. Lisäksi järjestelmistä on joskus saatavilla niin valtava määrä dokumentaatiota, että sen täydellisen läpikäymisen sijaan koetaan arvokkaammaksi tutkia sovellusta suoraan ja tukeutua dokumentaatioon niissä kohdissa, joista testaaja kokee tarvitsevansa enemmän tietoa. Täten kohdeyrityksen toteuttamat tietoturvatestaukset ovat käytännössä lähes aina gray box -testauksia.

Weidmanin (2014, s. 2) mukaan penetraatiotestausprojekti alkaa valmisteluvaiheella, jossa asiakkaan kanssa sovitaan testauksen tavoitteista, laajuudesta ja muista tarvittavista asioista. Varsinainen testaus aloitetaan tiedustelulla, jossa testaaja kerää tietoa kohteesta ja tunnistaa potentiaalisia haavoittuvuuskohtia. Seuraavaksi suoritetaan uhkamallinnus, jossa testaaja käyttää keräämäänsä tietoa hyökkäysten suunnittelussa. En-

nen hyökkäystä suoritetaan haavoittuvuusanalyysi, jossa kohteesta etsitään hyökkäyksissä hyödynnettäviä haavoittuvuuksia. Vasta tämän jälkeen suoritetaan itse hyökkäykset, joiden jälkeen projekti viimeistellään raportoimalla testauksen tulokset asiakkaalle. Whitaker ja Newman (2005) taas jaottelevat hyökkäyksen hieman eri tavalla viiteen vaiheeseen, jotka ovat tiedustelu, skannaus, pääsyn järjestäminen, pääsyn ylläpitäminen ja todisteiden hävittäminen.

Kohdeyrityksessä testaus jaetaan kolmeen vaiheeseen, jotka ovat tiedustelu, skannaus ja hyväksikäyttö. Kohdeyrityksen penetraatiotestausprosessi on esitelty liitteessä B. Tiedustelu ja skannaus vastaavat pitkälti Whitakerin ja Newmanin (2005) esittämiä kahta ensimmäistä vaihetta. Hyväksikäyttövaiheessa aiempien vaiheiden aikana havaittuja haavoittuvuuksia hyödynnetään järjestelmään tunkeutumiseksi. Testauksen raportissa kuvataan löytyneet havainnot, niiden hyödyntämismenetelmät ja vaikutukset, sekä annetaan ehdotuksia haavoittuvuuksien korjaamiseksi. Korjausten suorittaminen ja mahdollisista tietoturvaloukkauksista toipuminen ei kuitenkaan kuulu kohdeyrityksen tietoturvatestausprojektien piiriin. Haavoittuvuuksien korjausten riittävyyttä voidaan testata erillisessä korjaustarkastuksessa, jossa testataan vain edellisessä tietoturvatestauksessa löytyneet haavoittuvuudet, tai uusintatarkastuksessa, jossa haavoittuvuudet testataan osana koko järjestelmän uudelleentestausta.

3.2 Tietoturvatestausprojektien hallinta kohdeyrityksessä

Tietoturvatestausta on alun perin tehty ilman erityistä projektinhallintamallia yrityksen ollessa hyvin pieni. Aikanaan projekteissa on kuitenkin tunnistettu eri vaiheita, kuten aloituspalaveri, testaus ja raportointi, ja tämän myötä projektien toteuttaminen on siirtynyt vesiputousmallia mukailevaan tapaan. Yrityksen kasvaessa johtoryhmä halusi lisätä työn tehokkuutta ja henkilökohtaisten kontaktien kautta päädyttiin ostamaan valmis Kanban-koulutus, jonka myötä Kanban-taulu luotiin ja valjastettiin käyttöön yrityksessä. Lean-ajattelun mukaisen menetelmän valinnan taustalla vaikutti myös käsitys siitä, että moni yrityksen asioita tekee sovelluskehitystä ketterin tai Lean-menetelmin, ja tällaisen filosofian implementointi yritykseen voisi auttaa ymmärtämään asiakasta paremmin.

Vesiputousmalli on yksi tunnetuimmista olemassa olevista projektinhallintamalleista. Siinä tehtävät aloitetaan ja lopetetaan lineaarisesti, eli edellinen vaihe on täysin valmis ennen kuin seuraava aloitetaan. (Pries & Quigley, 2010, ss. 95–96.) Vesiputousmalli perustuu osin uskomukseen, jonka mukaan merkittävä määrä analyysiä ja suunnittelua tuottaa syvän ymmärryksen projektista, mutta malli ei huomioi tietotyön epävarmuutta ja nopeaa muutosta (Measey ym., 2015, s. 21). Murrayn (2016) mukaan vesiputousmal-

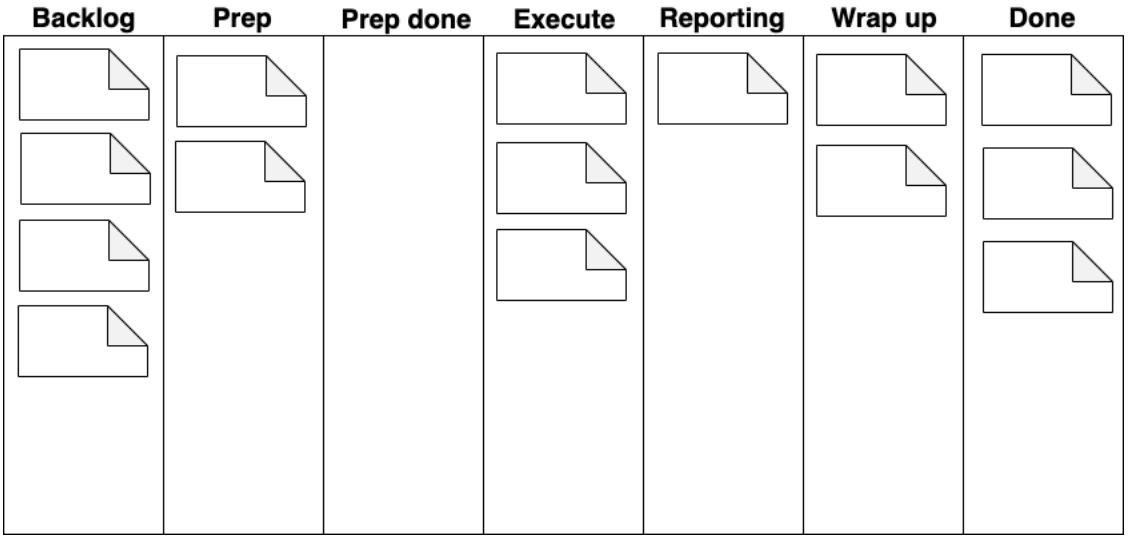
lissa on havaittu useita ongelmia, kuten joustamattomuus, täydellinen vaatimusmäärittely projektin alussa ja niiden mukaisen tuotteen toimitus vasta pitkän ajan päästä. Tässä ajassa vaatimukset ovat todennäköisesti ehtineet muuttua. Lisäksi vesiputouksen ongelmana on, että tuotteen arvo realisoituu vasta täysin valmiin tuotteen toimituksen yhteydessä.

Stoberin ja Hansmannin (2010) mukaan nykyäänkin on olemassa projekteja, jotka kannattaa toteuttaa vesiputousmallia hyödyntäen. Näitä ovat erityisesti pienet projektit, jotka ovat selkeitä ja toistavat suurimmaksi osaksi jotakin, joka on tehty jo aiemmin. Tätä väitettä tukee myös empiirinen tutkimus, sillä Ahimbisibwe ym. (2017) mukaan sopiva lähestymistapa riippuu projektityypistä ja projektin kontekstista.

Kohdeyrityksessä verkkosovelluksen tietoturvaa testataan vesiputousmallia mukaillen. Myyjä määrittelee projektin laajuuden asiakkaan kanssa teknisen asiantuntijan tukemana ja itse testausvaiheessa projektisuunnitelmaan ei tehdä muutoksia. Kesken testauksen asiakkaalle ilmoitetaan vain erityisen kriittisistä havainnoista, mutta muussa tapauksessa asiakas saa testauksen tulokset tietoonsa vasta testauksen päätteeksi raportin muodossa. Testausprojektit ovat kuitenkin hyvin lyhyitä, ja suurimmassa osassa testaus- ja raportointivaiheet ovat yhteensä vain noin viikon tai kahden pituisia.

Yrityksen käynnissä olevia projekteja seurataan Kanban-menetelmää soveltaen. Kanban on Lean-filosofiaan perustuva menetelmä, jossa työn näkyväksi tekeminen on keskiössä. Työtä visualisoidaan Kanban-taululla, jossa työ siirtyy työvaihetta kuvaavasta sarakkeesta toiseen vasemmalta oikealle kohti Valmis-saraketta. Yksinkertaisimmassa taulussa voi olla vain kolme saraketta; aloittamatta, työn alla ja valmis. Sarakkeita voi kuitenkin olla juuri niin paljon kuin Kanbania hyödyntävä tiimi kokee tarpeelliseksi. (Project Management Institute, 2017, ss. 103–105.) Kuvassa 2 on esitetty Kanban-taulu, jonka sarakkeet vastaavat kohdeyrityksessä tutkimuksen aloitushetkellä käytössä olevan Kanban-taulun sarakkeita.

Kanban-menetelmä perustuu Just-in-Time -filosofiaan, jossa resursseja vedetään oikealta kysynnän mukaan niin, että juuri oikea määrä resursseja on käytettävissä juuri oikeaan aikaan (Ríos-Mercado & Ríos-Solís, 2012, s. 263). Suuri ero Kanban-menetelmän ja muiden suosittujen tuotantomenetelmien välillä on se, että työtä ei työnnetä eteenpäin seuraavaan vaiheeseen heti sen valmistuttua, vaan sitä vedetään edellisestä vaiheesta silloin, kun tuotantokapasiteettia on vapaana (Leopold & Kaltenecker, 2015, ss. 18–19).



Kuva 2. *Esimerkki kanban-taulusta*

Kanban-menetelmässä työnkulku on jatkuvaa toisin kuin monessa ketterässä menetelmässä, joissa työtä tehdään ennalta määritetyissä iteraatioissa. Myös Kanbaniin voi soveltaa iteraatioita, kunhan säilytetään Kanbaniin kuuluvat keskeneräisen työn rajat ja tehtäviä vedetään jatkuvasti prosessin läpi työntämisen sijaan. (Project Management Institute, 2017, ss. 103–105.) Keskeneräisen työn rajoilla rajoitetaan sitä, kuinka monta keskeneräistä tehtävää kussakin taulun vaiheessa voi olla kerrallaan (Leopold & Kaltenecker, 2015, s. 19). Keskeneräisen työn rajojen tarkoituksena on edistää työn saattamista valmiiksi ennen uuden työn aloittamista ja parantaa sekä tuottavuutta että laatua (Project Management Institute, 2017, ss. 103–105).

Kohdeyrityksessä Kanban-taululla kutakin projektia kuvaa yksi post it -liimalappu. Alun perin oli tavoitteena jakaa projektit vielä pienempiin osiin taululle, mutta projektien pienen koon ja suuren määrän vuoksi koettiin, että yhden lapun edustaessa kokonaista projektia taulu tarjoaa riittävästi tietoa ja säilyy vielä helppolukuisena. Keskeneräisen työn rajoja ei ole käytössä, mutta käytännössä pyritään siihen, että kukin testaaja työstää vain yhtä projektia kerrallaan. COO lisää projekteja backlogille sitä mukaa kun myyjät saavat projekteja myytyä. Backlog sisältää listan tulevista projekteista, joiden valmistelua ei tehdä vielä aktiivisesti. Backlogilta projektit siirtyvät prep-vaiheeseen aloituspalaverin ja esitietojen hankkimisen ajaksi. Kun projektin kaikki esitiedot on saatu ja sekä testiympäristö että testauksessa käytettävät käyttäjätunnukset on testattu toimiviksi, projekti voidaan siirtää prep done -sarakkeeseen. Kun testaaja aloittaa projektin, se vedetään execute-vaiheeseen testattavaksi. Joskus käy kuitenkin niin, että seuraava projekti on jo alkamassa, kun edellistä vielä viimeistellään, ja tällöin Kanban-menetelmän vastaisesti projekteja työnnetään execute-vaiheeseen edellisten ollessa vielä kesken. Kun testaus on

valmis, projekti siirtyy suoraan reporting-vaiheeseen, jossa testauksesta kirjoitetaan raportti. Samat testaajat suorittavat sekä testauksen että raportoinnin. Kun raportti on lähetetty asiakkaalle, projekti siirtyy wrap-up vaiheeseen, jossa asiakkaan kanssa pidetään loppupalaveri ja projektia reflektoidaan sisäisesti. Kun nämä on suoritettu, projekti on valmis ja siirtyy sarakkeeseen done.

Jos työn etenemiselle on jokin este, se merkitään Kanban-taululle blockerina. Merkintä tapahtuu erottuvalla liimalapulla, joka liimataan esteellisen tehtävän päälle. Liimalappuun kirjoitetaan, mistä este johtuu, että tiimin olisi helpompi poistaa se. (Hammarberg & Sundén, 2014.) Blockereita esiintyy kohdeyrityksessä satunnaisesti, ja ne johtuvat usein asiakkaasta. Käytännössä blockereita on kolmea tyyppiä, eli joko asiakkaalta ei ole saatu tarvittavia esitietoja sovitussa aikataulussa, testiympäristö ei toimi tai testauksessa käytettävät käyttäjätunnukset eivät toimi. Nämä kaikki vaativat asiakkaalta toimenpiteitä ja kohdeyrityksessä voidaan vain kommunikoida asiakkaille, että ongelmat estävät testaamisen ja toivoa, että asiakas korjaa ne nopeasti. Jos blockereita ei saada poistettua nopealla aikataululla, projektia on siirrettävä.

4 KETTERÄT MENETELMÄT

Ketterät menetelmät ovat syntyneet vastareaktiona perinteisille sovelluskehitysmenetelmille, joissa painopiste on dokumentoinnissa ja perusteellisessa suunnittelussa ennen toteutuksen aloittamista (Dingsøyr ym., 2010). Sovelluskehitys on jatkuvassa muutoksessa, eivätkä kaikki asiakkaan tarpeet kehitettävälle tuotteelle ole tiedossa projektin aluksi, vaan tarpeet voivat muuttua tai paljastua vasta myöhemmin pitkien projektien aikana. Tämän tuloksena joukko konsultteja on kehittänyt menetelmiä ja käytäntöjä, jotka vastaavat sovelluskehityksen kohtaamaan väistämättömään muutokseen. (Cohen ym., 2004.) Nämä ketterät menetelmät jakavat yhteiset arvot ja perusajatukset, jotka on esitelty vuonna 2001 julkaistussa Agile manifestossa (Abrahamsson ym., 2002).

Agile manifestossa arvostetaan yksilöitä ja kanssakäymistä, toimivaa ohjelmistoa, asiakasyhteistyötä ja muutokseen vastaamista enemmän kuin menetelmiä ja työkaluja, kattavaa dokumentaatiota, sopimusneuvotteluja ja suunnitelmassa pitäytymistä. Julistuksen mukaan jälkimmäisetkään asiat eivät ole arvottomia, mutta edelliset ovat näitä merkittävämpiä. (Beck ym., 2001.) Kymmenen vuotta julistuksen luonnin jälkeen yksi sen kirjoittajista julkaisi päivitetyn version, jossa nostetaan tiimi yksilön edelle, todennettu oppiminen toimivan ohjelmiston edelle, asiakkaan tarpeiden ymmärtäminen asiakasyhteistyön edelle ja muutoksen aikaansaaminen muutokseen vastaamisen edelle (Hohl ym., 2018).

Ketterien menetelmien keskeinen eroavaisuus perinteisiin menetelmiin on se, että samalla kun ne keskittyvät intensiivisesti tuottavuuteen ja projektin ohjattavuuteen, ne tunnistavat ihmisen projektin keskeisenä onnistumisen edellytyksenä ja mahdollistajana (Abrahamsson ym., 2002). Ketterien menetelmien kolme tärkeintä onnistumisen edellytystä ovat kulttuuri, ihmiset ja kommunikaatio. Ilman kulttuurin tukea ei voida hyödyntää ketteriä menetelmiä, sillä tällöin ketterät arvot eivät toteudu. Menetelmissä tiimit ovat pieniä, mutta jäsenten korkea osaamisen taso on ensiarvoisen tärkeää. Kommunikaatio niin tiimin sisällä kuin toimittajan ja asiakkaan välillä ovat myös tärkeitä ketterien menetelmien menestykselle, ja kommunikaatiota edistetään muun muassa sillä, että kaikki tiimin jäsenet työskentelevät samassa tilassa. (Cohen ym., 2004.) Projektin ohjattavuus ja tuottavuus pyritään siis saavuttamaan osaavalla tiimillä, joka on omaksunut ketteriä menetelmiä tukevan kulttuurin, jossa kommunikoidaan aktiivisesti.

Tässä tutkimuksessa tarkasteltavien ketterien menetelmien valinnassa on käytetty julkaisuja, joissa on käsitelty ketteriä menetelmiä. Julkaisut on pyritty valitsemaan tasaisesti eri vuosilta, että myös tuoreemmat menetelmät olisivat edustettuna. Menetelmien valinnassa käytetyt julkaisut ja niiden luettelemat menetelmät on esitetty taulukossa 1. Julkaisut on järjestetty kronologiseen järjestykseen julkaisuvuoden mukaan, ja menetelmät ovat aakkosjärjestyksessä.

Taulukko 1. *Ketterien menetelmien esiintyminen kirjallisuudessa*

Ketterä menetelmä	**Abrahamsson ym. (2002)**	**Cohen ym. (2004)**	**Koch (2005)**	**Abrahamsson ym. (2010)**	**Measey (2015)**	**Project Management Institute (2017)**
Adaptive software development	x		x	x		
Agile modeling		x		x		
Agile project management					x	
Agile software process model				x		
Agile unified process						x
Crystal methods	x	x		x		x
Disciplined agile						x
Dynamic systems development method	x	x	x	x	x	x
Enterprise scrum						x
Extreme programming	x	x	x	x	x	x
Feature driven development	x	x	x	x		x
Internet-speed development				x		
Kanban					x	x
Large Scale Scrum						x
Lean development		x	x		x	
Lean start-up					x	
Open source software development	x					
Pragmatic programming				x		
Rational Unified Process	x					
Scaled agile framework					x	x
Scrum	x	x	x	x	x	x
Scrumban						x

Tähän tutkimukseen on valittu ne menetelmät, jotka esiintyivät suurimmassa osassa tarkastelluista julkaisuista. Kunkin julkaisun käsittelemät ketterät menetelmät on kuvattu merkillä ’x’, ja julkaisusta puuttuvien menetelmien kohdalla taulukossa on tyhjä solu. Jokaisessa menetelmien valintaan käytetyssä julkaisussa esiteltiin Dynamic systems development method, Extreme programming (XP) ja Scrum. Lisäksi Feature-driven development ja Crystal methods löytyivät suurimmasta osasta julkaisuja. Suosituista menetelmistä on myös kehittynyt itsenäisiä muunnelmia kuten Scrumban, joka soveltaa Scrum- ja Kanban-menetelmiä (Project Management Institute, 2017, s. 108).

4.1 Crystal methods

Crystal methods on menetelmäperhe, joka kehitettiin vuonna 1990 (Cohen ym., 2004). Menetelmiä on useita, koska Crystal tunnistaa, että kukin projekti on ainutlaatuinen ja ne voivat tarvita räätälöityjä politiikkoja, käytäntöjä ja prosesseja (Project Management Institute, 2017, s. 107). Crystal ei rajoita kehitysmenetelmiä tai työkaluja, ja se sallii esimerkiksi XP:n ja Scrumin elementtien hyödyntämisen samanaikaisesti projekteissa. Kaikkia Crystal-menetelmiä yhdistää kuitenkin tietyt säännöt, ominaisuudet ja arvot. (Abrahamsson ym., 2002.) Crystal-menetelmissä keskiössä on kommunikaatio ja sen tavoitteena on välttää ongelmat, jotka johtuvat sidosryhmien välisen kommunikaation puutteellisuudesta (Holcombe, 2008). Kommunikaatio on vahvasti osana muitakin ketteriä menetelmiä, ja se tunnistetaankin yhdeksi ketterien menetelmien keskeiseksi elementiksi.

Crystalin sisällä menetelmät on nimetty värien mukaan niin, että Crystal Clear on läpinäkyvimpänä värinä kaikista ketterin ja sitä seuraa Crystal Yellow, Crystal Orange, Crystal Red, jne. Käytettävä Crystal-menetelmä riippuu projektiin liittyvien ihmisten määrästä, eli menetelmä on väriltään sitä tummempi, mitä enemmän projektissa on ihmisiä. Lisäksi menetelmään vaikuttaa projektin kriittisyys, jonka mittarina on projektin epäonnistumisen aiheuttamien seurausten vakavuus. (Cohen ym., 2004.) Crystal Clear on suunniteltu pienille projekteille, joissa työskentelee enintään kuusi kehittäjää (Abrahamsson ym., 2002). Joissain tapauksissa myös kahdeksan tai kymmenen kehittäjää voi osallistua tällaiseen projektiin, mutta projektitiimin tulisi työskennellä jaetussa työtilassa kommunikaation helpottamiseksi (Dybå & Dingsøyr, 2008). Crystal Orange on suunniteltu keskikokoisille projekteille, joissa työskentelee kymmenestä neljäänkymmeneen henkilöä. Nämä projektit kestävät tyypillisesti noin vuoden tai kaksi. (Abrahamsson ym., 2002.) Crystal Yellow asettuu kooltaan näiden kahden värin välille ja Crystal Red on suunniteltu Crystal Orangea suuremmille projekteille.

Kaikissa Crystal-menetelmissä kehitystä tehdään inkrementaalisesti. Seuraava inkrementti suunnitellaan staging-vaiheessa, jossa tiimi valitsee tulevan inkrementin aikana toteutettavat vaatimukset sen mukaan, kuinka paljon he luulevat saavansa aikaan inkrementin aikana (Abrahamsson ym., 2002). Crystalissa on kaksi ehdotonta sääntöä, joiden mukaan inkrementin pituus ei saa ylittää neljää kuukautta ja menetelmässä käytetään reflektiopalavereja, joiden kautta menetelmä mukautuu ajan myötä (Highsmith & Highsmith, 2002, s. 262).

4.2 Dynamic Systems Development Method

Dynamic Systems Development Method (DSDM) on vuonna 1995 dynaamisia ja vaikeasti ennustettavia projekteja varten suunniteltu ketterä viitekehys, joka keskittyy aikaiseen toimitukseen ja iteratiiviseen kehitysprosessiin, jossa tehdään läheistä yhteistyötä asiakkaan kanssa (Hohl ym., 2018). Se keskittyy tarkkojen menetelmien määrittelyn sijaan johtamiseen ja soveltuu useiden erilaisten projektien toteuttamiseen (Measey ym., 2015, s. 140). Viitekehys on rajoitelähtöinen, sillä siinä aika, kustannukset ja laatu asetetaan muuttumattomiksi rajoitteiksi, joiden puitteissa toteutetaan kehitystä sen verran kuin on mahdollista (Project Management Institute, 2017, s. 110). Rajoitelähtöisyys on myös yksi ketteristä menetelmistä tunnistettu elementti, jota käsitellään tarkemmin seuraavassa luvussa.

DSDM-viitekehykseen kuuluu kahdeksan periaatetta, joiden mukaan DSDM:ssä keskitytään liiketoiminnan tarpeisiin, toimitetaan ratkaisuja ajallaan jakamalla tehtävät timeboxeihin, joihin kuluva aika on ennalta määritetty, tehdään yhteistyötä tiimin sisällä, pidetään kiinni sovitusta laatutasosta, työskennellään inkrementaalisesti ja iteratiivisesti, kommunikoidaan jatkuvasti ja osallistetaan tiimi suunnitteluun (Davis, 2012; Holcombe, 2008; Measey ym., 2015, ss. 140–141). DSDM ei viitekehyksenä rajoita tiimikokoja, tiimin jäsenten työskentelysijaintia tai iteraatioiden pituutta (Cohen ym., 2004). Tyypillinen time-boxin koko on kuitenkin parista päivästä muutamaan viikkoon (Abrahamsson ym., 2002).

DSDM:n elinkaaressa on viisi vaihetta, mutta ennen ensimmäistä vaihetta tulee varmistaa, että projekti on valmis alkamaan, rahoitus löytyy ja että kaikki edellytykset projektin onnistuneeseen toteuttamiseen ovat olemassa (Abrahamsson ym., 2002; Cohen ym., 2004). Ensimmäisessä vaiheessa varmistetaan projektin toteuttamiskelpoisuus ja määritetään, onko DSDM oikea lähestymistapa projektin toteutukseen. Toisessa vaiheessa muodostetaan ensimmäinen kuvaus projektin tuloksesta, esimerkiksi sovelluksen arkkitehtuurisuunnitelma, ja muodostetaan strategia projektin toteuttamiselle. (Coram &

Bohner, 2005.) Nämä kaksi vaihetta voidaan myös yhdistää omaksi pieneksi projektikseen (Davis, 2012).Tässä kohtaa projektin toteuttamiselle on luotu pohja ja iteratiivinen projektin toteuttaminen on valmis alkamaan (Measey ym., 2015, s. 144).

Kolmas vaihe on ensimmäinen mallin iteratiivisista vaiheista (Abrahamsson ym., 2002). Siinä tutkitaan ongelmaa ja etsitään sille mahdollisia ratkaisuja, joista voidaan rakentaa prototyyppejä (Davis, 2012). Tämän vaiheen prototyypit viimeistellään, yhdistellään ja testataan neljännessä vaiheessa, jonka tuloksena saadaan asetetut minimivaatimukset täyttävä tuote, jonka kehitystä voidaan jatkaa iteratiivisesti asiakkaan toiveiden perusteella (Coram & Bohner, 2005). Cohen ym. (2004) mukaan nämä kaksi vaihetta seuraavat samanlaista prosessia, jossa ensin tunnistetaan mitä halutaan tuottaa, sovitaan miten ja koska se tuotetaan, luodaan tuote ja lopuksi tarkastetaan, että tuote on tehty oikein.

Implementaatiovaiheessa järjestelmä toimitetaan asiakkaalle ja siirretään tuotantokäyttöön, ja sen käyttäjille järjestetään tarvittavat koulutukset järjestelmän käyttöä varten (Abrahamsson ym., 2002; Davis, 2012). Jos järjestelmää pitää vielä kehittää, mallin edellisiä vaiheita toistetaan, kunnes järjestelmä on valmis. (Cohen ym., 2004.) Mitä suurempia puutteita ja kehitystarpeita järjestelmässä havaitaan, sitä useampaa vaihetta toistetaan, ja hyvin suurien puutteiden löytyessä prosessi voidaan aloittaa jopa aivan ensimmäisestä vaiheesta (Abrahamsson ym., 2002). Projektin jälkeen suoritetaan normaalit siivoustoimenpiteet ja siirrytään ylläpitämään järjestelmää (Cohen ym., 2004). Vaikka DSDM:n on väitetty sopivan useiden erilaisten projektien toteuttamiseen, sen elinkaaren vaiheista näkyy vahvasti, että viitekehys on suunniteltu erityisesti sovellusten ja järjestelmien kehitykseen.

4.3 Extreme programming

Extreme programming (XP) kehitettiin 1990-luvulla ja se oli hallitseva ketterä menetelmä, kunnes Scrum syrjäytti sen 2000-luvun alkupuolella (Measey ym., 2015, s. 125). Menetelmä sisältää 12 käytäntöä, joista kuuluisin on pariohjelmointi, jossa kaksi kehittäjää työskentelee saman koneen ääressä ja kirjoittaa samaa koodia (Cohen ym., 2004; Hohl ym., 2018). Sittemmin menetelmä on kehittynyt ja siihen sisältyy useita muitakin käytäntöjä (Project Management Institute, 2017, s. 102). Extreme Programming -menetelmän arvot ovat kommunikaatio, yksinkertaisuus, palaute, rohkeus ja laadukas työ (Beck, 2004). Nimensä mukaisesti menetelmä on suunniteltu ohjelmistokehityksen tarpeisiin ja se näkyy vahvasti muun muassa projektin vaiheissa, jotka koskevat järjestelmän suunnittelua, kehitystä ja julkaisua.

Projekti alkaa tutkimisvaiheella, jossa asiakkaat ilmaisevat ensimmäiseen julkaisuun toivomansa ominaisuudet ja projektitiimi tutustuu työkaluihin, teknologioihin ja käytäntöihin, joita projektissa tullaan hyödyntämään (Coram & Bohner, 2005). Järjestelmä määritellään yhdessä asiakkaan ja kehittäjien kesken (Abrahamsson ym., 2002). Tutkimisvaihetta seuraa jokaisen iteraation alussa toistuva suunnitteluvaihe, jossa projektin sidosryhmät kokoontuvat arvioimaan ja priorisoimaan seuraavan julkaisun vaatimuksia (Cohen ym., 2004; Coram & Bohner, 2005).

Suunnitteluvaiheen jälkeen alkaa julkaisuversion iterointivaihe, johon kuuluu nimensä mukaisesti useita iteraatioita (Cohen ym., 2004). Kunkin iteraation kesto on yleensä viikosta neljään viikkoa ja iterointivaiheen viimeisen iteraation lopussa järjestelmästä on valmiina tuotantoon siirrettävä versio (Abrahamsson ym., 2002; Coram & Bohner, 2005). Kehitystä tehdään testaaminen edellä, eli ensin kirjoitetaan testit, jotka kehitetyn tuotteen tulee läpäistä iteraation lopuksi (Beck, 2004). Tuotteistamisvaiheessa järjestelmälle tehdään lisää testausta muun muassa suorituskyvyn näkökulmasta ennen kuin se luovutetaan asiakkaalle (Coram & Bohner, 2005). Kun ensimmäinen versio on tuotantokäytössä, siirrytään ylläpitovaiheeseen, jonka rinnalla toteutetaan uusia iteraatioita XP:n aiemmista vaiheista (Abrahamsson ym., 2002).

XP voi toimia minkä kokoisissa tiimeissä tahansa, mutta parhaat tulokset saavutetaan, kun tiimi työskentelee mahdollisimman paljon samassa tilassa (Beck, 2004). Kehittäjien kanssa samassa tilassa työskentelee jatkuvasti myös asiakkaan edustaja, joka voi vastata kysymyksiin, suorittaa hyväksymistestejä ja varmistaa, että kehitys etenee odotetulla tavalla (Koch, 2005, s. 248). Kun asiakkaalla ei ole enää uusia vaatimuksia järjestelmälle tai niiden toteuttaminen ei ole enää kannattavaa, saavuttaa se lopettamisvaiheen (Coram & Bohner, 2005). Vasta tässä vaiheessa järjestelmästä kirjoitetaan dokumentaatio, sillä se ei enää muutu (Abrahamsson ym., 2002).

4.4 Feature Driven Development

Feature Driven Development (FDD) kehitettiin 1990-luvun lopulla (Cohen ym., 2004). Se ei kata sovelluskehityksen koko elinkaarta, vaan keskittyy sovelluksen suunnitteluun ja rakentamiseen (Abrahamsson ym., 2002). FDD:ssä on kuusi eri roolia, mutta yksi henkilö voi edustaa useampaa kuin yhtä roolia. Roolit ovat projektipäällikkö, pääarkkitehti, kehityspäällikkö, pääohjelmoija, luokan omistaja ja sovellusasiantutija. (Project Management Institute, 2017, s. 108.) FDD-prosessissa on viisi eri vaihetta, joista kaksi viimeistä ovat iteratiivisia (Abrahamsson ym., 2002).

Muista ketteristä menetelmistä poiketen FDD:ssä kokonaisuuden suunnittelu toteutetaan projektin aluksi ja iteraatiot keskittyvät suunniteltujen ominaisuuksien toteuttamiseen, vaikka suunnitelmaa päivitetäänkin iteraatioiden yhteydessä (Koch, 2005, s. 249). Ensimmäisessä vaiheessa kehitetään yleinen malli, joka jaetaan pienempiin osakokonaisuuksiin (Cohen ym., 2004). Osia suunnitellaan tarkemmalla tasolla muun muassa hahmottelemalla niihin toteutettavia ominaisuuksia ja erilaisia vaihtoehtoja, ja näistä tarkemmin suunnitelluista osakokonaisuuksista muodostuu yhdessä ensimmäinen suunnitelma kokonaisuudelle (Abrahamsson ym., 2002).

FDD:n toisessa vaiheessa luodaan lista ominaisuuksista, jotka ovat pieniä, asiakkaalle hyödyllisiä osia tuotteesta (Cohen ym., 2004). Ominaisuuslistan pohjana toimii ensimmäisessä vaiheessa luotu suunnitelma ja siihen liittyvät alustavat ominaisuusehdotukset (Abrahamsson ym., 2002). Kolmannessa vaiheessa valmis ominaisuuslista priorisoidaan ryhmiin, jotka annetaan pääkehittäjälle ja tämä jakaa niiden omistajuuden ja vastuut edelleen muille kehittäjille (Cohen ym., 2004; Hunt, 2006). Priorisointi toteutetaan ominaisuuksien tärkeyden ja niiden välisten riippuvuuksien perusteella (Abrahamsson ym., 2002).

Kaksi viimeistä vaihetta muodostavat prosessin iteratiivisen osuuden. Iteraation aluksi pääkehittäjä valitsee pienen joukon ominaisuuksia, jotka toteutetaan seuraavaksi (Hunt, 2006). Ominaisuudet valitaan niin, että niiden toteutukseen kuluu korkeintaan kaksi viikkoa (Ashmore & Runyan, 2015). Tämän ryhmän sisältämät ominaisuudet suunnitellaan tarkemmin, toteutetaan, testataan, ja integroidaan kehitettävään järjestelmään (Hunt, 2006). Samaa järjestelmää voi kehittää yhtä aikaa useampi tiimi, joilla on oma joukko ominaisuuksia toteutettavana (Abrahamsson ym., 2002).

4.5 Scrum

Scrum on ketterän ohjelmistokehityksen malli, joka on osoittautunut tehokkaaksi fyysisesti samassa tilassa työskentelevien, pienikokoisten tiimien tapauksessa. Se on arvolähtöinen malli, joka tukee asiakkaiden muuttuviin tarpeisiin vastaamista asiakastyytyväisyyden takaamiseksi. (Ashraf & Aftab, 2017.) Scrum ei ota kantaa siihen, mitä tekniikoita sovelluskehityksessä tulisi hyödyntää, vaan keskittyy määrittelemään tiimin jäsenten toimintaa ketteryyden saavuttamiseksi alati muuttuvassa toimintaympäristössä (Abrahamsson ym., 2002). Menetelmä on lainannut nimensä rugby-lajista (Cohen ym., 2004).

Scrumissa on kolme avainroolia, jotka ovat tuoteomistaja, scrum master ja kehitystiimi. Kehitystiimi on itseohjautuvasti toimiva tiimi, jonka jäsenillä on yhdessä kaikki projektin

toteuttamiseen tarvittava osaaminen, eikä sitä tarvitse hakea tiimin ulkopuolelta (Project Management Institute, 2017, s. 101). Scrum master toimii tiimin palvelevana johtajana, jolla ei ole tiimin jäseniä korkeampaa auktoriteettia vaan jonka tehtävänä on fasilitoida ja johtaa scrum-prosessia, sekä mahdollistaa tiimin itseohjautuvuus (Measey ym., 2015, s. 132). Tuoteomistaja on vastuussa kehitettävästä tuotteesta ja sen arvon maksimoinnista (Project Management Institute, 2017, s. 101).

Tuotekehitystä tehdään itseohjautuvissa tiimeissä lyhyissä jaksoissa, joita kutsutaan sprinteiksi (Ashraf & Aftab, 2017). Scrumissa sprintin pituudeksi on alun perin ehdotettu yhdestä kuuteen viikkoa, mutta yleensä sprintit ovat neljän viikon mittaisia (Cohen ym., 2004). Jokainen sprintti alkaa suunnittelupalaverilla ja päättyy retrospektiiviseen arviointiin, jossa analysoidaan projektin edistymistä ja esitellään nykyistä järjestelmää (Ashraf & Aftab, 2017; Cohen ym., 2004).

Tuoteomistaja ylläpitää tuotebacklogia, joka sisältää priorisoidun listan ominaisuuksista, jotka halutaan toteuttaa (Ashraf & Aftab, 2017). Kullekin ominaisuudelle arvioidaan yhdessä scrum-tiimin kanssa työmääräarvio, jota päivitetään iteratiivisesti sitä mukaa kun ominaisuuksista on saatavilla tarkempaa tietoa (Abrahamsson ym., 2002). Tuotebacklogilta valitaan ominaisuudet, jotka halutaan toteuttaa tulevan sprintin aikana ja muodostetaan näistä priorisoitu sprintin backlog sprintin suunnittelupalaverissa (Ashraf & Aftab, 2017). Palaverissa myös määritetään tulevalle sprintille tavoite, jolla voidaan viestiä, miksi sprinttiin kuuluvat tehtävät suoritetaan ja kuinka yksityiskohtaisesti ne tulisi toteuttaa (Koch, 2005). Kun sprintti on suunniteltu, sprintin backlogin tehtävät jäädytetään niin, että niihin ei voida tehdä enää sprintin aikana muutoksia (Cohen ym., 2004).

Scrum-mallissa päivittäistä toimintaa ohjaa jokaisen työpäivän alussa järjestettävä Scrum-palaveri, joka kestää noin 15 minuuttia (Ashmore & Runyan, 2015). Tämä palaveri on kriittinen Scrum-menetelmän onnistumiselle ja sen tavoitteena on parantaa tiimin kommunikaatiota, tiedottaa kaikkia sidosryhmiä kuten asiakkaita ja tuoteomistajia projektin tilanteesta, tunnistaa kohdattuja ongelmia ja pitää koko tiimi keskittyneenä yhteistä tavoitetta kohti työskentelyyn (Cohen ym., 2004). Scrum-palaverin aikana kehitystiimi koordinoi suoritettavat aktiviteetit ja scrum master selvittää ongelmat, jotka voivat haitata tiimin toimintaa (Ashraf & Aftab, 2017). Kehitystiimin jäsenet valitsevat itse, minkä tehtävän parissa kukin jäsen haluaa työskennellä (Cohen ym., 2004).

Suosituista ketteristä menetelmistä on kehittynyt muunnelmia, jotka ovat itsenäistyneet omiksi menetelmikseen. Esimerkiksi Scrumban on Project Management Instituten (2017) mukaan ketterä menetelmä, joka on alun perin kehitetty tavaksi siirtyä Scrum-menetelmästä noudattamaan Kanban-menetelmää. Sittemmin se on kuitenkin kehittynyt

itsenäiseksi viitekehykseksi, jossa hyödynnetään elementtejä kummastakin menetelmästä.

Scrumban-menetelmässä työ jaetaan sprintteihin kuten Scrumissa ja työtä visualisoidaan Kanban-taululla (Ellis, 2016). Scrumban ei määrittele erityisiä rooleja, vaan roolit voidaan jakaa tiimin tarpeiden mukaan (Alqudah & Razali, 2018). Yksittäiset tehtävät eli tarinat sijoitellaan Kanban-taululle ja tiimi hallitsee työmääräänsä work-in-progress -rajoitusten avulla, eli tiimillä voi olla työn alla vain tämän rajoituksen mukainen määrä tehtäviä kerrallaan (Ellis, 2016). Project Management Instituten (2017) mukaan Scrumbanissa tiimi asettaa myös ”planning triggerin”, jonka kohdalla pidetään seuraava suunnittelupalaveri tulevan työn suunnittelemiseksi. Yleensä tämä planning trigger on esimerkiksi silloin, kun työtä on jäljellä work-in-progress -rajaa vähemmän. Tiimi pitää päivittäisiä tapaamisia yhteistyön edistämiseksi ja eteen tulevien esteiden poistamiseksi (Ellis, 2016).

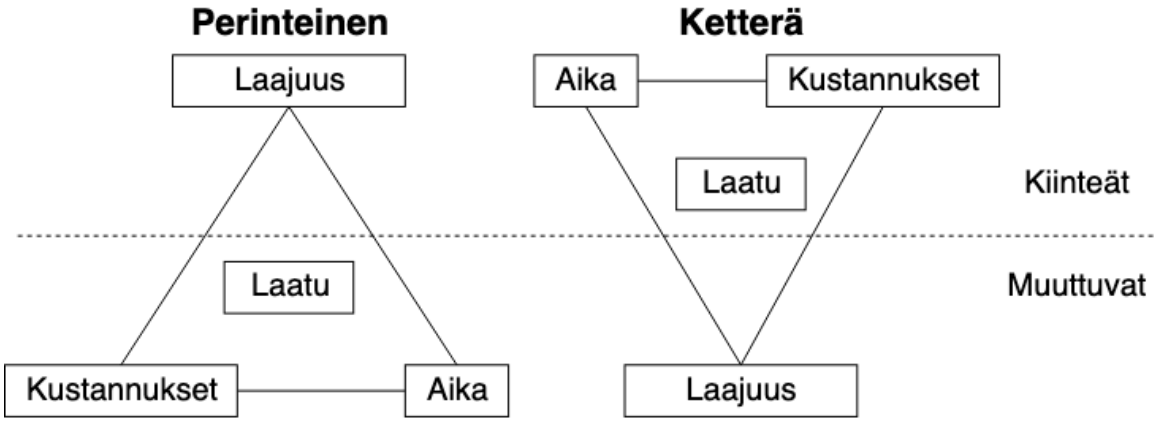
5 KETTERIEN MENETELMIEN ELEMENTIT

Tässä luvussa esitellään elementtejä, joita tarkastelluista ketteristä menetelmistä on tunnistettu. Siinä missä edellisen luvun tarkoituksena oli esitellä yksittäisiä menetelmiä, tässä luvussa tutustutaan tarkemmin elementteihin, joita esiintyy useissa ketterissä menetelmissä. Measey ym. (2015, s. 125) mukaan yleisimmät ketterien menetelmien elementit ovat lähtöisin Extreme Programming -menetelmästä.

Ketterissä menetelmissä projektin kustannukset ja aika määritellään etukäteen ja projektin laajuus muotoutuu projektin edetessä näiden rajoitteiden puitteissa. Ketterille menetelmille on ominaista myös itseohjautuvissa tiimeissä työskentely niin, että tiimin johtaja toimii palvelevana johtajana ja esteiden poistajana. Työtä suunnitellaan etukäteen vain sen verran kuin on tarpeen ja tarkka suunnitelma tehdään vain lähitulevaisuudessa toteutettavista tehtävistä. Työn etenemistä on ketterissä menetelmissä helppo seurata kaikille saatavilla olevien visuaalisten informaatiotaulujen avulla. Ketterät menetelmät ovat myös hyvin kommunikaatiokeskeisiä ja kommunikaatiossa suositaan kasvotusten kommunikointia. Kommunikaatiota pyritään edistämään säännöllisillä palavereilla kuten päivittäisellä palaverilla, jossa seurataan projektin etenemistä ja havaitaan esteitä. Tärkein palaveri ketterissä menetelmissä on kuitenkin retrospektiivi, joka fasilitoi oppimista ja sitä kautta toiminnan kehittämistä.

5.1 Projektin muuttujat

Kuvassa 3 esitellään projektinhallintakolmio, jota kutsutaan myös esimerkiksi rautakolmioksi (Pollack ym., 2018; Stober & Hansmann, 2010, ss. 28–29). Se on esitys projektin kolmesta päärajoitteesta, jotka ovat aika, kustannukset ja projektin laajuus. Yhden rajoitteen muuttuminen vaikuttaa kahteen muuhun rajoitteeseen, sillä esimerkiksi projektin laajuuden kasvaessa myös projektin toteuttamiseen vaadittu aika ja kustannukset kasvavat. Jos näistä kaksi lukitaan, kolmas määräytyy lukittujen rajoitteiden perusteella (Wysocki ym., 2013, s. 462). Projektipäällikön haasteena on hallita näitä rajoitteita ilman, että se vaikuttaa asiakkaalle toimitettavan tuotteen tai palvelun laatuun.



Kuva 3. *Kiinteät ja muuttuvat ominaisuudet perinteisissä ja ketterissä projekteissa, mukaillen (Measey ym., 2015, s. 19; Stober & Hansmann, 2010, s. 29)*

Perinteisessä vesiputousmallin mukaisessa projektinhallinnassa projektin laajuus on kiinteä elementti, sillä projektin sisältö määritellään ennen projektin alkua (Stober & Hansmann, 2010, s. 93). Esimerkiksi sovelluskehitysprojektissa kaikki toteutettavat ominaisuudet määritellään projektin suunnitteluvaiheessa. Measey ym. (2015, ss. 18–20) mukaan tämä toimii hyvin tapauksissa, joissa ominaisuudet ovat riittävän yksinkertaisia eivätkä vaatimukset muutu todennäköisesti projektin aikana. Muuttuvassa toimintaympäristössä voi kuitenkin esiintyä ongelmia, sillä jos alkuperäistä suunnitelmaa joutuu muuttamaan, muutokset voivat kasvattaa projektin kustannuksia ja projektin vaatimaa aikaa merkittävästi. Jos esimerkiksi projektiin lisätään työvoimaa aikataulun venymisen ehkäisemiseksi, projektin kustannukset kasvavat ja uuden työvoiman perehdyttämiseen tarvittava aika venyttää aikataulua edelleen, mikä johtaa uudelleen tarpeeseen lisätä työvoimaa projektiin.

Measey ym. (2015, ss. 18–20) mukaan toinen riski perinteisessä vesiputousmallissa on muutosten vaikutus projektin laatuun. Kun halutaan välttää kustannusten ja ajan kasvattamista, voidaan esimerkiksi vähentää toiminnallista testausta tai tehdä muita kompromisseja, joiden avulla saadaan projekti valmiiksi aiemmin, mutta joiden vaikutukset voivat realisoitua vasta myöhemmin. Myöskään Wysocki ym. (2013, s. 462) mukaan yritys kiinnittää kaikki projektin muuttujat ei luultavasti johda hyvään lopputulokseen. Koska muuttujat riippuvat toisistaan, yritys kiinnittää kaikki muuttujat johtaa todennäköisesti ristiriitoihin, ja jonkin muuttujan on annettava periksi. Jos esimerkiksi projektin kustannukset, aika ja laajuus yritetään kaikki lukita, joudutaan todennäköisesti tinkimään laadusta kaikkien muiden rajoitteiden noudattamiseksi, eikä työn tulos ole niin laadukasta kuin aluksi on suunniteltu.

Measey ym. (2015, s. 21) mukaan ketterässä projektissa tarvittavasta työstä on aikaisessa vaiheessa yleensä hyvin vähän tietoa. Tällöin kaikki arviot ovat epävarmoja ja niitä tarkennetaan projektin edetessä, kun tieto projektista lisääntyy. Lukitsemalla suurin osa projektin muuttujista saadaan kuitenkin luotua tasainen ja kestävä tuotantorytmi, jonka perusteella projektien etenemistä on helpompi ennustaa (Stober & Hansmann, 2010, s. 118). Ketterässä projektinhallintakolmiossa muuttujat on käännetty perinteiseen malliin nähden päinvastoin, eli perinteisen mallin muuttuvat ominaisuudet ovat ketterässä mallissa kiinteitä ja päinvastoin. Aika, kustannukset ja laatu määritellään etukäteen, ja projektin laajuus mukautuu näiden muuttujien asettamien rajojen mukaan (Stober & Hansmann, 2010, s. 93). Tämän konseptin nimi on ”time-boxing”. Sen idea on, että projekti on yksi time-box, johon kuluva aika on ennalta määritelty. Tämä voidaan jakaa useaan lyhyeen ja hallittavaan inkrementtiin, joita voidaan kutsua sprinteiksi tai iteraatioiksi. Nämä ovat omia time-boxejaan projektin sisällä. (Measey ym., 2015, ss. 19–20.)

Mitä pienempiä yksittäiset time-boxit ovat, sitä helpompi projektia ja sen etenemistä on hallita ja seurata (Stober & Hansmann, 2010, s. 102). Tällöin suunnittelua tarvitsee tehdä vain verrattain lyhyen ajan päähän, minkä jälkeen alkaa jo uusi time-box, jonka suunnittelun ollessa ajankohtaista projektista voi olla jo enemmän tietoa. Täten suunnittelu voidaan tehdä paremmin kuin silloin, jos tämä uusi time-box olisi ollut esimerkiksi suuremman time-boxin loppuosa, jolloin se olisi suunniteltu aiemmin ja siitä olisi voinut olla saatavilla vähemmän tietoa suunnittelun tueksi. Ketterissä menetelmissä projektin tehtävistä ylläpidetään priorisoitua listaa, josta tehtäviä otetaan työn alle prioriteettijärjestyksessä tärkeimmäksi priorisoidusta tehtävästä alkaen (Ashmore & Runyan, 2015). Kun aika ja kustannukset on lukittu, tehtäviä valmistuu se määrä, joka rajatussa ajassa ehditään toteuttamaan käytettävissä olevien resurssien avulla ja asetetut laatuvaatimukset täyttäen. Priorisoinnin myötä on varmistettu, että tärkeimmät tehtävät tulevat tehtyä ensin ja vähemmän tärkeät jäävät toteuttamatta, kun asetettu aikaraja täyttyy.

5.2 Tiimit

Ketterissä menetelmissä keskitytään vahvasti tiimityöhön (Stober & Hansmann, 2010, s. 13). Tiimit toimivat itseohjautuvasti, eli heillä on valta päättää työskentelystään itse (Project Management Institute, 2017, s. 39; Stober & Hansmann, 2010, s. 7). Tähän liittyy usein muun muassa tavoitteiden jakaminen yksittäisiin tehtäviin. Yksityiskohtaisen tehtävälistan myötä tiimi kykenee myös tuottamaan tarkemman työmääräarvion, sillä yksi tehtävä on tyypillisesti kooltaan niin pieni, että se voidaan tehdä alle kymmenessä tunnissa. (Measey ym., 2015, s. 45.) Tiimi myös jakaa tehtävät keskenään itse ja he

voivat päivittäisissä palavereissa päättää, kuinka heidän kannattaa järjestäytyä parhaan mahdollisen lopputuloksen saavuttamiseksi (Project Management Institute, 2017, s. 39).

Itseohjautuvien tiimien eduiksi on havaittu muun muassa nopeampi päätöksenteko, koska tiimit voivat tehdä päätöksiä itse, lisääntynyt motivaatio, koska työ tuntuu vapaammalta, lisääntynyt älyllinen suorituskyky, koska ongelmia voi ratkoa koko tiimi yhden henkilön sijasta, sekä lisääntynyt aloitteellisuus ja jatkuva kehittäminen, sillä itseohjautuvissa tiimeissä jäsenet alkavat pitää tiimin tavoitteita omaa henkilökohtaista etuaan tärkeämpinä (Measey ym., 2015, ss. 46–47; Stober & Hansmann, 2010, s. 8). Lisäksi tiimin jäsenet ymmärtävät luultavasti parhaiten omat ja toistensa vahvuudet ja erityisosaamisen, minkä vuoksi heidän on mahdollista jakaa tehtävät keskenään hyödyntäen niitä parhaalla mahdollisella tavalla. Tiimin itseohjautuvuus ei kuitenkaan tarkoita, että tiimi saa tehdä mitä ikinä haluaa, vaan tiimin toimintaa ohjaavat korkeamman tason ohjeet ja rajoitukset. Näiden tehtävänä on varmistaa, että toiminta on johdonmukaista ja ne pyritään asettamaan niin, että ne toimivat organisaation ja tiimin hyväksi häiritsemättä itseohjautuvuutta merkittävästi. (Measey ym., 2015, s. 46.) Esimerkiksi monikaan tiimi ei voi valita jäseniään, budjettiaan tai ratkaistavaa liiketoimintaongelmaansa.

Ketterän tiimin johtajaan viitataan eri nimillä eri menetelmissä. Kaikissa menetelmissä johtajan päävastuu on kuitenkin mahdollistaa tiimin itseohjautuvuus ja jatkuva kehittyminen (Project Management Institute, 2017, s. 33). Joissain ketterissä menetelmissä johtaja nähdään muutoksen aloittajana, kun taas toisissa rooliin kuvataan sisältyvän enemmän tiimin johtajuutta (Measey ym., 2015, s. 48.). Ketterän tiimin johtajan tulee toimia palvelevana johtajana, joka ei käske ja kontrolloi tiimiä vaan palvelee tiimin tarpeita ja auttaa tiimiä itseohjautumaan kohti tiimin yhteisiä tavoitteita (Project Management Institute, 2017, s. 33; Stober & Hansmann, 2010, s. 8). Measey ym. (2015, ss. 48–49) mukaan käskevä ja kontrolloiva johtaja tuo tiimille tyypillisesti valmiin suunnitelman, jonka luontiin tiimi ei ole osallistunut. Jos tiimi epäonnistuu suunnitelman toteuttamisessa, syytetään johtajaa ja syytöstä perustellaan huonolla suunnitelmalla. Tiimi ei ohjaudu itse, koska jäsenet eivät koe sitä turvalliseksi eikä heillä ole siihen motivaatiota.

Ketterissä menetelmissä johtajan tulee ymmärtää, että menetelmä keskittyy yksilöihin ja näiden väliseen tehokkaaseen kommunikaatioon kasvotusten (Measey ym., 2015, ss. 48–49). Lisäksi johtajan tulee voida luoda kulttuuri, jossa ihmiset eivät pelkää uusien asioiden kokeilua ja ymmärtävät, että kaikki tekevät virheitä ja virheet ovat hyväksyttäviä niin kauan kuin niistä opitaan (Stober & Hansmann, 2010, s. 8). Johtajalla tulee myös olla kokemusta muutoksesta ja ketteryydestä (Measey ym., 2015, s. 48). Ketterän tiimin johtajalta vaaditaan siis paljon osaamista, joka painottuu erityisesti ihmisten johtamiseen projektin johtamisen sijaan.

5.3 Työn visualisointi

Ketterille menetelmille on tyypillistä, että työtä visualisoidaan niin, että tieto sen etenemisestä on kaikille saatavilla (Koch, 2005, s. 80). Visualisoinnin tarkoituksena on tarjota kaikille ajantasaista tietoa ilman, että heidän tarvitsee keskeyttää tiimin jäseniä kysyäkseen asiasta (Project Management Institute, 2017, s. 152). Työtä voi visualisoida tiimin sopivaksi kokemalla tavalla, mutta visualisoinnista olisi hyvä käydä ilmi vähintään tehtävien tila ja kuinka kaukana ne ovat valmistumisesta. Visualisointi voi sisältää myös muuta informaatiota, kuten kuka työskentelee minkäkin tehtävän parissa. (Measey ym., 2015, ss. 83–84.) Tärkeintä on, että visualisoinnin sisältämä tieto muuttuu ja päivittyy, koska silloin sen jatkuva jakaminen on perusteltua (Cockburn, 2006). Lisäksi on tärkeää, että visualisointia todella päivitetään tiedon muuttuessa ja päivittyessä, sillä jos visualisointia ei käytetä ja päivitetä, se voi myös sisältää harhaanjohtavaa informaatiota, jonka perusteella tehdyt toimenpiteet ja päätökset voivat olla turhia tai jopa haitallisia.

Visualisoinnissa suositaan yleensä fyysistä ratkaisua kuten seinälle tai tussitaululle rakennettua tilannekuvausta, mutta myös digitaalisia työkaluja työn visualisointiin on olemassa (Measey ym., 2015, ss. 83–84). Digitaalisen taulun suurimpana ongelmana voidaan nähdä se, että sitä pitää mennä erikseen katsomaan, kun taas fyysisen taulun informaatiolle on helppo altistua vain kävelemällä taulun ohi (Cockburn, 2006). Toinen digitaalisten taulujen ongelma on se, että ne on helpompi unohtaa ja jättää päivittämättä, jolloin ne eivät tarjoa ajantasaista ja oikeaa tietoa työn etenemisestä. (Measey ym., 2015, ss. 83–84.) Fyysisessäkin ratkaisussa on kuitenkin ongelmia, kuten se, että maantieteellisesti hajautunut tiimi ei pysty näkemään ja päivittämään fyysistä taulua yhtä lailla kuin samassa sijainnissa työskentelevä tiimi, jonka kaikki jäsenet pääsevät tauluun käsiksi.

5.4 Työn suunnittelu

Ketterät menetelmät on luotu vastaamaan nopeasti muuttuvan toimintaympäristön tarpeisiin, ja siksi Stoberin ja Hansmannin (2010, s. 97) mukaan kullakin hetkellä suunnittelua kannattaa tehdä vain niin paljon kuin on välttämätöntä ja sillä tasolla, mikä tuntuu järkevältä. Heidän mukaansa ei kannata tuhlata aikaa tekemällä epätarkkoihin tietoihin perustuvia suunnitelmia, jos tiedot tulevat tarkentumaan tulevaisuudessa merkittävästi ja riski suunnitelmien muutostarpeeseen on suuri.

Ketterissä menetelmissä työtä suunnitellaan pääasiassa kahdella eri tasolla. Stoberin ja Hansmannin (2010, s. 95) mukaan korkean tason suunnitelma määrittelee karkeasti projektin prioriteetit, tavoitteet, projektitiimin ja niin edelleen. Toimintaympäristössä, jossa projektit ovat hyvin lyhyitä voi olla perusteltua, että korkean tason suunnitelma koskee

useampaa projektia kerrallaan ottaen kantaa esimerkiksi tulevien projektien keskinäiseen prioriteettijärjestykseen (Project Management Institute, 2014). Tulevaa työtä suunnitellaan projektin backlogille, jossa ylläpidetään priorisoitua listaa tulevista tehtävistä tai käyttäjätarinoista (Ashmore & Runyan, 2015). Lista muuttuu jatkuvasti sitä mukaa kun uusia tehtäviä tunnistetaan ja priorisoidaan, ja kun tärkeimmät prioriteettitehtävät siirtyvät työn alle.

Jotta projektinhallinta olisi vaikeasti ennustettavassa ympäristössä helpompaa, projekti voidaan jakaa useaan yhtä pitkään osaan, joita kutsutaan iteraatioiksi tai sprinteiksi (Measey ym., 2015, ss. 19–20). Sopivasta iteraation pituudesta on olemassa useita näkemyksiä, mutta yleisin mielipide puoltaa kahden viikon pituutta (Stober & Hansmann, 2010, s. 118). Kaikki iteraatiot ovat keskenään yhtä pitkiä (Davis, 2012, s. 59). Työtä suunnitellaan tarkasti vain kuluvan iteraation tai viikon osalta, ja tässä suunnitelmassa voidaan muun muassa eritellä, kuka tiimin jäsenistä työstää mitäkin tehtävää (Stober & Hansmann, 2010, s. 95). Project Management Instituten (2017, s. 25) mukaan iteraatioiden sijaan tehtäviä voidaan toteuttaa myös jatkuvana virtana, jolloin uutta työtä otetaan backlogilta sitä mukaa kun tiimin kapasiteettia vapautuu ja tällöin suunnittelu- ja retrospektiiviaikataulut määritellään erikseen. Tällöin on tärkeää pitää keskeneräisen työn määrä pienenä. Keskeneräisen työn rajoittamista voidaan pitää muutenkin kannattavana tiimityöskentelyn parantamiseksi, sillä mitä vähemmän työtä on kerrallaan kesken, sitä todennäköisemmin tiimin jäsenet tekevät yhteistyötä ja voivat näin saada tehtäviä valmiiksi tehokkaammin.

Iteraation suunnittelussa hyödynnetään muun muassa tietoa käyttäjätarinoiden toteutuksen arvioidusta kestosta, tiimin työskentelytahdista ja iteraation aikaisista tapahtumista, kuten mahdollisista lomista, jotka vaikuttavat käytettävissä olevaan työaikaan. Iteraation tarkempi suunnittelu aloitetaan päättämällä, kuinka monta käyttäjätarinaa halutaan saada valmiiksi iteraation aikana, ja tästä muodostetaan iteraation tavoite. Tämän jälkeen käyttäjätarinat jaetaan yksittäisiin tehtäviin, joista tiimin jäsenet valitsevat ne tehtävät, jotka kukin haluaa suorittaa. Lopuksi tiimin jäsenet antavat yksittäisille tehtäville työmääräarviot. (Ashmore & Runyan, 2015.) Erittäin kokeneet tiimit voivat jättää käyttäjätarinat jakamatta yksittäisten tehtävien tasolle, sillä heillä voi olla tarpeeksi taitoa ja kokemusta toimittaa projekteja ilman niin tarkkaa suunnittelua (Measey ym., 2015, s. 45). Tällöin työn suunnittelu ei ole niin yksityiskohtaista, sillä työtä ei jaeta pieniin osiin. Measey ym. (2015, s. 45) mukaan tarinoiden jakamatta jättäminen yksittäisiin tehtäviin voidaan kuitenkin yleisesti nähdä virheeksi, sillä se on yksi hyvin yleinen syy sprintin tai iteraation tavoitteiden täyttymättömyyteen.

5.5 Kommunikaatio ja palaverit

Monissa ketterissä menetelmissä suositellaan vahvasti tiimien työskentelyä samassa tilassa kommunikoinnin edistämiseksi, sillä kommunikointia pidetään yhtenä ketterien menetelmien keskeisimmistä elementeistä (Ashraf & Aftab, 2017; Beck, 2004; Holcombe, 2008). Kochin (2005, s. 23) mukaan myös fyysisesti hajautetut tiimit voivat hyödyntää ketteriä menetelmiä, mutta tiedossa on kommunikaatiohaasteita, jotka tulee huomioida ja pyrkiä ratkaisemaan. Kommunikaatio toimii parhaiten kasvotusten, ja tätä kommunikaatiomenetelmää tulisikin suosia väärinkäsitysten välttämiseksi ja kommunikaatiotiheyden kasvattamiseksi (Measey ym., 2015, s. 8). Fyysisesti samassa tilassa työskennellessä ihmisten ympärille muodostuu niin sanottu kommunikaatiokupla, jonka sisällä kommunikaatiota tapahtuu jatkuvasti niin verbaalisin kuin nonverbaalisinkin keinoin. Kahden sijainnin välille voidaan avata hetkellisesti kommunikaatioväylä puhelun tai telekonferenssin muodossa, mutta ne ovat silti vain väliaikaisia ja kasvotusten kommunikointia köyhempiä kommunikaatioväyliä. (Koch, 2005, s. 23.) Puhelujen yhteydessä kehonkieltä ei voi tulkita ja tekstimuotoisessa viestinnässä myös äänenpainot jäävät pois lisäten väärinkäsitysten mahdollisuuksia.

Ketterissä projekteissa päivittäiset palaverit järjestetään nimensä mukaan päivittäin ja niissä käydään läpi edellisen palaverin jälkeen tapahtunut edistys ja suunnitellaan seuraavaan palaveriin mennessä tehtävä työ (Project Management Institute, 2017, s. 53). Measey ym. (2015, s. 75) mukaan paras aika päivittäisen palaverin pitämiselle on noin puoli tuntia työpäivän alkamisen jälkeen, sillä tällöin työntekijät ehtivät piipahtaa työpisteillään ja muistella edellisen päivän asioita ennen palaveriin osallistumista, mutta päivittäisessä palaverissa ehditään sopimaan jokaiselle päivän prioriteettitehtävät. Päivittäisen palaverin kesto tulisi olla korkeintaan 15 minuuttia ja palaverin pitämiseksi lyhyenä on tavallista, että kaikki osallistujat seisovat koko palaverin ajan (Koch, 2005, s. 77). Palaverin aikana havaitut ongelmat tulisi selvittää palaverin ulkopuolella erikseen ongelmaan liittyvien henkilöiden kesken heti palaverin jälkeen (Measey ym., 2015, s. 75). Myös Koch (2005, s. 77) ja Project Management Institute (2017, s. 54) painottavat, että vaikka palaverissa kerrotaan esiintyneistä ongelmista, niitä ei ratkaista palaverin aikana vaan niistä keskustellaan tiimin johtajan ja kiinnostuneiden tiimin jäsenten kesken palaverin jälkeen, jolloin muut voivat jatkaa töitä.

Project Management Instituten (2017, s. 50) mukaan retrospektiivi on ketterien menetelmien tärkein elementti, sillä se mahdollistaa tiimin oppimisen, mukautumisen ja kehittymisen. Se on tilaisuus, jossa projektitiimi analysoi omaa työskentelyään ja tunnistaa niin hyviä asioita kuin kehityskohteitakin (Measey ym., 2015, s. 77). Tarkoituksena on tunnistaa ongelmia ja niiden juurisyitä, suunnitella näihin vaikuttavia parannustoimenpiteitä

ja kehittää toimintasuunnitelmia, joiden avulla toimintaa kehitetään parempaan suuntaan (Project Management Institute, 2017, s. 51). Tilaisuuksien tavoitteena on tekemisen jatkuva parantaminen ja ne fasilitoi tyypillisesti ketterän tiimin johtaja (Measey ym., 2015, s. 77). Ashmoren ja Runyanin (2015) mukaan fasilitoinnissa voi esimerkiksi käyttää kortteja, joille osallistujat listaavat havaintonsa ja mielipiteensä siitä, mitä heidän mielestään tulisi jatkaa, lopettaa tai alkaa tekemään. Retrospektiivi pidetään yleensä sprintin päätteeksi varsinkin jos sprintin pituus on noin kaksi viikkoa, mutta tärkeintä on pitää retrospektiivi säännöllisesti (Project Management Institute, 2017, ss. 50–51).

Sprinttien ja julkaisujen jälkeen järjestetään demotilaisuus, jossa projektitiimi esittelee valmiit tuotoksensa projektin sidosryhmille (Ashmore & Runyan, 2015). Tässä tilaisuudessa tiimi saa palautetta ja tunnustusta tekemästään työstä ja tilaisuuteen tulisi osallistua kaikki tiimin jäsenet sekä kaikki saatavilla olevat sidosryhmät (Measey ym., 2015, ss. 76–77). Demoesityksen valmisteluun ei ole tarkoitus käyttää valtavasti aikaa, vaan pääpainon tulee olla hienojen PowerPoint-kalvojen sijaan työn laadukkaissa tuloksissa (Ashmore & Runyan, 2015). Measey ym. (2015, ss. 76–77) mukaan palaveri tuottaa enemmän arvoa, jos tuotoksia esitellään käytännössä kuvankaappauksien sijaan.

6 YKSILÖHAASTATTELUJEN LÖYDÖKSET

Kaikki haastateltavat olivat yhtä mieltä siitä, että yleisellä tasolla kohdeyrityksen tekniset tietoturvatestausprojektit toimivat hyvin. Useampi haastateltavista kehui näiden projektien prosessin selkeyttä, ja etenkin prosessin alkupuolella on koettu tapahtuneen selviä parannuksia viime aikoina. Kohdeyrityksen toiminnasta ja projektien toteuttamisesta tunnistettiin kuitenkin muutamia ongelmia, jotka liittyvät pääasiassa kommunikaatioon ja projektin laajuuden määrittelyyn. Muutaman testaajan mukaan ongelmat projektin laajuudessa koskevat koko alaa, eivät vain kohdeyritystä.

Suuren haasteen kohdeyrityksen projekteihin näyttää tuovan projektien lyhyt kesto. Koska projektit ovat hyvin lyhyitä, viivästykset ovat projektin kokoon nähden helposti merkittävän suuria. Viivästyksiä aiheuttavat usein tekniset ongelmat, joita kohdeyritys voi harvoin itse ratkaista täysin. Yleensä ongelmista voidaan vain kommunikoida asiakkaalle ja toivoa, että tämä saa ongelman ratkaistua mahdollisimman nopeasti. Yksi testaajista on myös sitä mieltä, että tekniset tietoturvatestausprojektit ovat liian lyhyitä ketterien menetelmien soveltamiseen. Hänen mielestään ketteristä menetelmistä soveltuvat elementit on jo otettu käyttöön, sillä työtä suunnitellaan etukäteen ja projektin lopuksi mietitään mitä on saatu aikaiseksi. Hänen mielestään ketterät menetelmät soveltuvat paremmin pidempiin projekteihin, joiden suunnittelua voidaan tehdä iteratiivisesti.

6.1 Projektin muuttujat

Tällä alalla projektin laajuuden arviointi on vaikeaa. Kohdeyrityksen myyjät käyttävät laajuuden arvioinnissa apuna teknistä asiantuntijaa, jonka tehtävä on testattavan kohteen käyttöliittymäesittelyn eli niin sanotun demon ja muiden saamiensa tietojen perusteella arvioida, kuinka monta henkilötyöpäivää tarvitaan siihen, että kohde ehditään testaamaan riittävän kattavasti. Teknisen asiantuntijan antama arvio on kuitenkin parhaimmillaan valistunut arvaus, jos arvio tehdään vain asiakkaan tekemän järjestelmäesittelyn perusteella. Niin kauan kuin asiantuntija ei pääse itse käymään järjestelmää läpi, testauksen vaatiman työmäärän arviointi on hyvin vaikeaa. Järjestelmään tutustuminen veisi aikaa, ja se ei ole välttämättä kannattavaa ottaen huomioon, että tutustumiseen kuluva aika olisi pois testaajan tekemästä laskutettavasta työstä. Yksi testaajista huomauttaa, että vaikka testaaja voisikin antaa arvion projektin laajuudesta päästyään tutustumaan testattavaan järjestelmään päiväksi tai kahdeksi, tämäkään arvio ei välttämättä ole tarkka.

> ”Sovelluksia ei oikein pysty skouppaan järkevästi. Koko alalla se on sama ongelma.” (Haastateltava B)

Laajuuden arviointi epäonnistuu testaajien mukaan siksi, että asiakas ei osaa esitellä järjestelmästään relevantteja asioita. Yksi haastateltava kiteyttää asian niin, että ongelmana on saada sekä myyjä, asiakas, että testaaja ymmärtämään samalla tavalla, mitä ollaan testaamassa. Testaajien mukaan asiakasta voitaisiin auttaa luomalla ohje siihen, mitä testaajat katsovat järjestelmästä saadakseen käsityksen sen laajuudesta, että asiakas voi tarjota vastaavan tiedon työmääräarviota varten.

Projekteja myydään kiinteällä projektihinnalla, mikä johtaa siihen, että laajuuden aliarviointi johtaa taloudellisiin tappioihin. Haastateltavista tuntuu myös siltä, että projekteja myydään helposti liian pieneen hintaan, mikä johtaa siihen, että testaukselle jää liian vähän aikaa, jos se halutaan tehdä kannattavasti. Jos testausta myytäisiin tuntilaskutuksella antamalla alustava arvio ja kattohinta, paine siirtyisi laajuuden arvioinnista raportointiin. Tällöin raportoidessa olisi eriteltävä, mitä järjestelmästä on ehditty testaamaan ja mitkä asiat jäivät testaamatta.

> ”Jos asiakas ostaa vähemmän aikaa projektiin niin voidaanhan se aika myös käyttää ja sillon voidaan löytää osa [haavoittuvuuksista] mut voidaan myös sanoo et ei löydetty kaikkii varmasti koska ei ollu aikaa, ja sillon niinkun semmosissa caseissa monesti must tuntuu et välillä penaajat [testaajat] on silleen et tää on skoupattu päin v****u koska tää on niin iso, vaikka se idea ei oo se, että se kolutaan samanlail ku joku toinen projekti. Et mun mielestä se skouppaus toimii paljon paremmin, kun mitä ehkä välillä tulee niinku palautetta.” (Haastateltava D)

Laajuuden arvioinnin lisäksi sovitun laajuuden toteuttamisessa on haastateltavien mukaan ongelmia. Testaajat haluavat kovasti testata kohdejärjestelmän läpikotaisin huolimatta siitä, kuinka suppeaksi projektin laajuus on myyjän ja asiakkaan kesken määritetty. Tämä johtaa siihen, että testaajat käyttävät projekteihin sovittua enemmän aikaa ja asiakas saa tarjoukseen ja projektin hintaan verrattuna enemmän arvoa.

> ”Kommunikaatio on ehkä sisäisesti puutteellista, että niinkun testaajalle ei oo aina selvää, että kuinka monta päivää se voi käyttää siihen työhön.” (Haastateltava G)

> ”Erityisesti jos ei tunnu mitään löytyvän niin koetaan, että mä oon tehny jotenkin huonoo työtä.” (Haastateltava G)

Yksi haastateltavista totesi, että usein testaajista vain tuntuu, ettei aikaa ole riittävästi, ja ongelma on sisäisen kommunikaation puutteellisuudessa. Hänen mukaansa testaajille ei ole aina selvää, kuinka paljon aikaa he voivat testauksen käyttää. Lisähaasteen tuo se, että testaukselle ei ole selvästi määritelty, koska on testattu tarpeeksi, vaan testaajan tulee itse päättää, koska testaus on valmis. Kokeneet testaajat ovat vähemmällä testaamisella varmempia siitä, että he ovat löytäneet kaiken tarvittavan testattavasta kohteesta, mutta vähemmän kokeneet testaajat haluavat testata enemmän varmistuakseen

siitä, että kaikki löydettävissä olevat haavoittuvuudet on varmasti paikannettu. Tämä korostuu tilanteissa, joissa testattava kohde on toteutettu tietoturvan näkökulmasta hyvin ja havaintoja ei ole tai ne ovat vaikutukseltaan vähäisiä.

> ”Mun mielestä sen [epäonnistuneen laajuuden arvioinnin selvittämisen] pitäis olla jotenkin sen myyjän ja asiakkaan välinen dialogi … myyjä hommaa lisää aikaa.” (Haastateltava G)

Suurin osa haastateltavista on sitä mieltä, että jos huomataan projektin laajuuden arvioinnin menneen pieleen, siitä pitää ilmoittaa asiakkaalle välittömästi. Useimpien mielestä tällaisessa tilanteessa tulee testata vain sen verran kuin ehditään. Yksi testaajista ehdottaa, että asiakkaan kanssa tulisi tällaisessa tilanteessa pitää palaveri, jossa sovitaan yhdessä mitä kaikkea testataan ja mitä jätetään pois. Toisen testaajan mielestä asiakkailta tulisi kuitenkin pyytää lisää rahaa, että testaukseen voitaisiin käyttää enemmän aikaa ja koko järjestelmä ehdittäisiin testata. Kolmas haastateltava alleviivaa, että tällaisessa tilanteessa myyjän on otettava vastuu kommunikaatiosta, sillä hän on sopinut projektin laajuudesta asiakkaan kanssa alun perinkin.

6.2 Tiimit

> ”Pitäis olla tiimijako, pitäis tehdä tiimissä ihan ehdottomasti.” (Haastateltava D)

> ”Toi on niinku jotenki aika virhealtista et porukka tekee yksin, … se vaatii … liikaa osaamista yhdeltä ihmiseltä silleen et sun pitää samaan aikaan iteroiden tehä sitä penaa ja samaan aikaan sun pitäis syventyy johonkin tiettyihin asioihin ni se ei oikein mun mielestä toimi niinku yhen ihmisen projektina.” (Haastateltava D)

Kohdeyritys on päättänyt ottaa käyttöön tiimijaon, ja tähän suhtaudutaan pääosin myönteisesti, vaikka kaikki eivät uskokaan sen muuttavan työskentelyä tulevaisuudessa merkittävästi. Tiimityöskentelyn toivotaan vaikuttavan yrityksen ilmapiiriin positiivisesti ja sitouttavan henkilöstöä paremmin organisaatioon, kun yksittäiset työntekijät tuntevat kuuluvansa porukkaan aiempaa vahvemmin. Nyt moni kokee toimivansa pitkälti yksilönä.

> ”Mä ehdotan, että se [työn jakaminen tiimin sisällä] tapahtuis hyvin semmosella tota liberaalilla tavalla … tiimi saa itte päättää kuka tekee ja mitä tekee.” (Haastateltava E)

> ”Duunii pystytään jakaa järkevästi niille ihmisille, jotka osaa tehdä tiettyjä asioita ja sit tiimi saa ite päättää et kuka tekee ja mitä tekee.” (Haastateltava B)

Suurin osa haastateltavista on ehdottomasti sitä mieltä, että projektit tulee jakaa tulevaisuudessa tiimille ja tiimien tulee itse saada jakaa työ haluamallaan tavalla. Yksi haastateltavista tuo esiin, että testaajilla on metatieto siitä, mitä kukin testaaja osaa tehdä. Hänen mukaansa tätä tietoa ei ole muualla yrityksessä. Tämä puoltaa sitä mielipidettä, että tiimin jäsenten tulisi saada jakaa tehtävät keskenään itse. Toisen haastateltavan mie-

lestä jonkun pitää kuitenkin valvoa tehtävien jakamista, sillä hänen mielestään tiimin saadessa jakaa projektit keskenään kiinnostaviin projekteihin tulee luultavasti olemaan paljon halukkaita, kun taas toisia projekteja kukaan ei tahdo tehdä. Hän on sitä mieltä, että testaajat eivät voi jakaa työtä keskenään ilman johtajaa. Yksi haastateltavista arvelee, että etenkin tiimitoiminnan alkuvaiheessa tiimin jäsenille on vahvemmin kerrottava, kuka tekee mitäkin, mutta ajan myötä tehtäviä opitaan jakamaan tehokkaammin ja työnjakoon voidaan suhtautua kevyemmin.

> ”Mä en tiedä sit et, miten sen [projektien jakamisen tiimin sisällä] saa tehtyä niinku sillee reilusti ilman et siin on joku vähän määräämässä koska … ei kukaan haluu tehä jotain p***aprojektii jos sulla on joku paljon mielenkiintosempi projekti siin vieressä.” (Haastateltava F)

> ”Kyllä se [projektien jakaminen tiimin sisällä] vaatii vähän semmosta karkkien jakamista, että saa niinku kaikki saa tehdä jotain mielenkiintoista välillä ja kaikki joutuu tekee vähän tylsempii projektei toisinaan.” (Haastateltava F)

Tiimijaon suurimpana hyötynä odotetaan olevan työkuorman tasaantuminen tiimin jäsenten välillä. Nykyisin kun projektit on allokoitu yksittäisille henkilöille, tieto työmäärästä ja käynnissä olevista projekteista ei ole riittävän hyvin saatavilla. Tämän vuoksi toimettoman työntekijän on vaikea hypätä avuksi toiseen projektiin, jossa lisäkäsille voisi olla tarvetta. Usein tilanne on se, että uuden henkilön tutustuminen testattavaan järjestelmään veisi liian kauan aikaa. Kun koko tiimi on tietoinen tiimin projekteista, on helpompi kohdistaa resursseja tarpeen mukaan ja esimerkiksi sairastapauksissa tiimin muut jäsenet voivat viedä projekteja eteenpäin sairastuneen henkilön puolesta.

> ”Tiimien käyttö vois mahdollistaa sit sen, että tavallaan eri tyyppien vahvuudet pääsis eri tavalla niinku käyttöön että se tyyppi joka tykkää enemmän testaa ja vähemmän esiintyy niin vois keskittyy siihen mutta kumminkin kun porukalla tehdään niin kaikki oppis vähän siit toisestakin domainista.” (Haastateltava G)

> ”No se [työskentely tiimissä] ois vähemmän stressaavaa, sit se ois varmasti niinku pidemmän päälle tehokkaampaa.” (Haastateltava D)

Työkuorman tasaamisen lisäksi tiimien odotetaan kasvattavan yrityksen tuottavuutta mahdollistamalla kunkin työntekijän erityisosaamisen tehokkaampi hyödyntäminen. Erityisosaamista toivotaan myös siirrettävän työntekijältä toiselle yrityksen sisäisen hiljaisen tiedon jakamisen lisääntymisen myötä. Käytännössä tiimin jäsenten toivotaan oppivan erilaisia testaustekniikoita ja muita ammatillisesti hyödyllisiä asioita toisiltaan. Koska testaajat ovat kaikki melko nuoria, heidän toivotaan saavan kokemuksen tuoman varmuuden puuttuessa varmuutta toimintaansa siitä, että ammatillisten mielipiteiden muodostamiseen osallistuu useita henkilöitä.

> ”Semmonen niinku kysymys itsellä on, että miten sitä tiimiä johdetaan. Että onks joku team leader ja jos on, niin mikä sen niinku homma on.” (Haastateltava G)

> ”Tiimihän tarttee jonkinnäkösen niinku tiiminjohtajan joka tapauksessa, muutenhan toi homma hajoo käsiin koska et sä voi laittaa neljää penaajaa [testaajaa] vaan yhdessä et jonku pitää mun mielest niinku pitää vähä niitä niinku ohjaksia siin käsissä.” (Haastateltava F)

Yksi testaajista arvaa, että tiimien implementoinnin alkuvaiheessa tiimin johtamisessa voi olla haasteita. Hänen mielestään tiimi vaatii sellaisen johtajan, joka on osaamiseltaan tarpeeksi tekninen ymmärtääkseen koko testauksen kontekstin. Toinenkin haastateltava nostaa esiin epäilyksensä tiimin johtamisesta. Hän pohtii sitä, tuleeko tiimillä olemaan erillistä johtajaa ja jos tulee, mikä tämän rooli ja auktoriteetti muihin tiimin jäseniin on. Hän ottaa esimerkiksi scrum masterin, jolla ei ole auktoriteettia siihen, kuka tekee mitäkin vaan keskittyy varmistamaan, että kaikilla on tarpeeksi tehtävää. Scrum masterin yhteydessä nostetaan esiin yrityksessä vallitseva johtamisosaamisen puute, sillä kukaan yrityksestä ei ole kouluttautunut scrum masteriksi ja muutakin johtamisosaamista nähdään olevan yrityksessä puutteellisesti.

> ”Jos ei kukaan osaa hallinnoida semmosta tiimiä ni sit se on niinku riski että hommat ei etene yhtä nopeesti ku nytten kun ihmisillä on henkilökohtainen vastuu.” (Haastateltava D)

Kaksi haastateltavista nosti esiin, että tiimeistä hyötyminen edellyttää muutosta työskentelytavoissa yrityksen sisällä. Toinen korosti organisatorisen muutoksen johtamisen tärkeyttä ja totesi, että tiimien toimintaa tulee valvoa ja mitata, ettei toiminta lipsahda takaisin vanhaan malliin, jossa työtä tehdään yksinäisemmin. Tiimijaosta huolimatta yksittäisillä henkilöillä tulee edelleen olemaan myös henkilökohtaisia vastuita, kuten tietoturvakoulutuksia, jotka vaativat vain yhden henkilön työpanosta.

Yksi testaajista pohtii sitä, kuinka tiimit saadaan pitkällä tähtäimellä työskentelemään kohti suurta yhteistä visiota. Hän kuitenkin lisää tähän, että tämä on helppoa, jos kullakin tiimillä on oma selkeä funktionsa. Useampi haastateltava toi esiin huolen siitä, että tiimeissä vaarana on jäädä jumiin yhteen rooliin. Heidän mielestään olisi hyvä, että tehtäviä voisi edelleen tehdä riittävän monipuolisesti, vaikka sitten vaihtamalla välillä tiimiä.

6.3 Työn visualisointi

Suurin osa testaajista kokee, että yrityksessä nyt käytössä oleva Kanban -taulu ei auta heitä konkreettisesti visualisoimalla työn etenemistä. Heidän mielestään taulu on suunnattu COO:lle ja projektipäälliköille, jotka voivat katsoa taululta missä vaiheessa projekti on. Jos tiimillä olisi oma taulu, sekin toimisi yhden testaajan mukaan tiedotusvälineenä projektin statuksesta projektipäällikölle.

> ”Kyllähän mä ite tiän missä kohtaa mun omat projektit menee, ku mä niitä teen koko ajan.” (Haastateltava B)

> ”Mä en nää sitä et yksittäinen testaaja hyötyy tosta Kanbanista [Kanban-taulusta] kauheesti.” (Haastateltava B)

Suurin osa haastateltavista on sitä mieltä, että kaikki projektit on hyvä pitää yhdessä Kanban-taulussa, eivätkä tiimikohtaiset Kanban-taulut saa kannatusta. Haastateltavien mielestä kaikkien yhteinen Kanban-taulu olisi hyvä jakaa riveihin niin, että kullakin tiimillä olisi oma rivi. Tiimin omalta backlogilta halutaan nähdä, mitä projekteja tiimille on tulossa ja taulun halutaan myös kertovan, mitkä projektit ovat tulossa työn alle seuraavaksi. Haastateltavien mielestä ei ole tarpeellista pilkkoa projekteja Kanban-taululle pienemmiksi paloiksi, vaan valtaosan mielestä lappu per projekti on hyvä resoluutio.

> ”Pidetään se [Kanban-taulu] vaan mahdollisimman simppelinä ja et sä niinku nopeesti näät siinä et mikä projekti sul on tulossa ja mikä sul on menossa ja missä vaiheessa se on.” (Haastateltava F)

> ”Musta se on ihan hyvä, ei ei niinku Kanban-tauluu kannata ton tarkemmaks laittaa.” (Haastateltava D)

Keskeneräisen työn rajoittaminen Kanban -taululla ei saa haastateltavilta merkittävää kannatusta. Yksi haastateltava on sitä mieltä, että ne eivät toimi kohdeyrityksen kontekstissa, koska projektit ovat keskenään hyvin eri kokoisia, mikä tekee työn rajoittamisesta projektien määrän mukaan kannattamatonta. Toisen haastateltavan mukaan tiimikohtaiset rajat voivat toimia, mutta liian tiukat rajat muodostuvat ongelmaksi poikkeustilanteissa, esimerkiksi kun yksi projekti ei pääse etenemään asiakkaasta johtuvasta syystä ja testaajat haluaisivat tehdä odotellessa uutta projektia. Käytännössä testaajat testaavat tai raportoivat aina yhtä projektia kerrallaan, mutta valmistelevat ja päättävät samaan aikaan muita projekteja osallistumalla palavereihin ja testaamalla, että pääsyt järjestelmiin toimivat.

> ”Intuitiivisesti tuntuu siltä et noi hyökkäyspuut mitä ollaan pohdittu niin se vois olla paras tapa pilkkoo se [projekti] … usean ihmisen samaan aikaan työstettäväksi.” (Haastateltava G)

Yksi haastateltava tuo esiin, että tiimityössä korostuu työtehtävien näkyväksi tekeminen. Hänen mukaansa on tärkeää voida nähdä kullakin hetkellä, mitkä tehtävät ovat työn alla ja mitkä aloittamatta, että kaksi testaaja ei vahingossa tee päällekkäistä työtä. Testaajat ovat kokeilleet viimeisen puolen vuoden aikana hyökkäyspuiden luomista verkkosovellusten tietoturvatestausprojekteissa, ja useampi haastateltava pitää tätä potentiaalisena apuvälineenä yksittäisen projektin tehtävien visualisoinnissa. Hyökkäyspuulla tarkoitetaan kaaviokuvaa, jota käytetään kohdetta koskevien uhkien dokumentoimiseen (Ince, 2013).

Erään testaajan mielestä hyökkäyspuulla voi hyvin kommunikoida asiakkaalle testauksen laajuutta, mutta itse testauksen aikana niistä ei ole testaajille merkittävää hyötyä

niiden tekemiseen kuluvaan aikaan suhteutettuna. Hyökkäyspuun tekemisestä on hyötyä testauksessa vain, jos se tehdään projektin alkuvaiheessa. Jos testattava kohde on hyvin yksinkertainen, hyökkäyspuun luominen alkuvaiheessa vie yhden haastateltavan mukaan turhaan aikaa testaukselta, mutta monimutkaisemmassa järjestelmässä puun rakentaminen voi auttaa hahmottamaan järjestelmää kokonaisuutena. Toisen testaajan mukaan testausta voi auttaa myös testattavan kohteen visualisointi miellekartan avulla. Eräs haastateltavista on sitä mieltä, että hyökkäyspuu on työmäärän visualisoinnissa huomattavasti parempi keino kuin lista asioista tarjouksessa.

6.4 Työn suunnittelu

Suurin osa haastateltavista on sitä mieltä, että projektit tulisi tulevaisuudessa allokoida suoraan tiimeille sen sijaan, että COO suunnittelisi yksittäisten henkilöiden aikataulut päivän tarkkuudella. Suosittu ehdotus tiimityön aikataulutukseen on se, että tietyssä ajassa pitää saada valmiiksi ennalta määritellyt projektit, mutta tiimi voi itse päättää kuka tekee mitäkin projektia ja milloin. Tällöin tiimissä tiedetään heti, mihin projektiin voi siirtyä avuksi, jos oma projekti viivästyy tiimistä riippumattomasta syystä. Testaajat kuitenkin toivovat, että he voisivat lähtökohtaisesti välttää projektista toiseen siirtymistä kesken kaiken mahdollisimman paljon, ja että he saisivat olla kussakin projektissa sen alusta loppuun saakka.

> ”Se on vähän ongelmallista, että sä lennät projektista toiseen pienellä aikavälillä.” (Haastateltava F)

> ”Ei oo järkee heittää niinku lyhyeks ajaks johonkin projektiin tyyppiä, ku sul menee vaikka sul ois tosi hyvin dokumentoitu se asia ja kaikki ni kyl sun pitää vähän aikaa sitä kliksutella kumminkin sitä järjestelmää.” (Haastateltava F)

Yksi haastateltava on ehdottomasti sitä mieltä, että projektit tulisi pilkkoa pienempiin osiin. Hänen mukaansa aiemmin mainittu hyökkäyspuu voisi toimia tässä apuvälineenä niin, että puun haaroille tehdään työmääräarviot ja ne jaetaan tehtävinä tiimin jäsenille. Toisenkin haastateltavan mielestä projekteja voisi yrittää jakaa pienempiin tehtäviin, ja hän pohtii, että näiden koko voisi olla esimerkiksi noin yksi henkilötyöpäivä. Hän ei kuitenkaan osaa sanoa, olisiko tällainen mahdollista teknisissä tietoturvatestausprojekteissa ja jos on, millaisia tehtävien tulisi olla.

Työnjakoon testausprojektien sisällä ehdotetaan myös rooliperusteista menetelmää, jossa olisi kerääjiä ja analyytikoita. Kerääjän tehtävä olisi käydä testattavaa kohdetta läpi iteratiivisesti ensin pinnallisesti ja joka iteraatiolla aina hieman syvällisemmin. Aina kun kerääjälle tulee vastaan kohteessa jokin ominaisuus, jota tulisi testata perusteellisemmin, tämä antaa perusteellisemman testauksen tehtäväksi analyytikolle. Analyytikon ei

tarvitse ymmärtää testattavaa palvelua kokonaisuutena vaan riittää, että tämä testaa yksittäisiä ominaisuuksia. Tämä mahdollistaisi sen, että jos jollain testaajalla on ylimääräistä aikaa, hän voi siirtyä lyhyeksikin ajaksi toiseen projektiin analyytikon rooliin, eikä aikaa tarvitse kuluttaa testattavaan järjestelmään tutustumiseen.

> ”Se [projektien toteuttaminen neljän viikon sprinteissä] vois ehkä toimia, se pitää kokeilla”. (Haastateltava B)

> ”Se [projektien toteuttaminen sprinteissä] on mun mielestä noissa penaprojekteissa [tietoturvatestausprojekteissa] kokeilemisen arvonen koska tietyllä tapaa meillä on kuitenkin suurin osa penaprojekteista [tietoturvatestausprojekteista] ni se on max 2 viikkoo mitä sitä oikeesti tehdään. Jos meillä olis kahden viikon sprintit, niin se osuis siihen aika hyvin et se on mun mielestä kokeilemisen arvonen mut ei ehkä semmonen niinku mihin pitäis niinku olla silleen väkisin tiätsä et nyt meil on niinku nää kahden viikon sprintit.” (Haastateltava D)

Haastattelevat kokevat vaikeaksi arvioida, voisiko sprinttien hyödyntäminen toimia kohdeyrityksen teknisissä tietoturvatestausprojekteissa. Kukaan haastateltavista ei kuitenkaan tyrmää ajatusta ja moni on sitä mieltä, että sprintit ovat kokeilemisen arvoinen asia. Mahdollisesta sprinttien pituudesta on kahta erilaista mielipidettä, joista toinen on kaksi viikkoa ja toinen neljä viikkoa. Pidemmän sprintin arvellaan antavan enemmän vapautta projektien tekemisessä, mutta toisaalta muutama testaaja pohtii myös sitä, kuinka asiakkaat suhtautuvat siihen, että heidän järjestelmänsä voi olla testausvuorossa vasta kuukauden päästä alkavan sprintin lopussa, eli lähes kahden kuukauden päästä. Monet ovat sitä meiltä, että sprintin sopivaa pituutta on mahdotonta tietää etukäteen, ja että kohdeyritykselle sopiva pituus selviää vain erilaisia vaihtoehtoja kokeilemalla. Sprinteistä pidetään ajatuksena siksi, että vaikka yksi projekti viivästyisi, samaan sprinttiin on suunniteltu muitakin projekteja, joita voidaan tehdä sillä välin, kun yhtä asiakasta odotetaan. Tällöin kenenkään ei tarvitsisi olla toimettomana.

> ”Näkisin että sillä [sprinteillä] pystyis pelaamaan sitä aikataulutusta niinku joustavammaks [kuin projektien ottaminen työn alle backlogilta sitä mukaa kun edelliset valmistuvat].” (Haastateltava E)

> ”Se sprinttipohjanen aikataulutus niinku sille penatiimin [tietoturvatestaustiimin] vetäjälle olis kyllä kätevä koska, koska on helpompi aikatauluttaa sillee kahen viikon palikoihin niinku juttuja kun silleen et sä merkkaat vaikka jokaisen päivän et mitä tehään. Et vaikka se olis sit kanban-tyylinen et meil ei olis mitään erityisii kahden viikon välein joku retro ja taas intro palaveri seuraavaan sprinttiin niin meillä vois olla niinku sillee kuitenki että kahen viikon sloteissa aikataulutetaan asioita et ilmotetaan kahen viikon tarkkuudella asia tai silleen että et asiakkaalle sanotaan vaikka nytte joku kolmen viikon päässä oleva aika et tällön tää on valmis.” (Haastateltava D)

Useampi haastateltava pohtii sprinttien kohdalla sitä, kuinka asiakkaalle voidaan kommunikoida testausajankohta riittävällä tarkkuudella. Asiakkaalle tarkalla testausajankohdalla on merkitystä muun muassa siksi, että usein asiakas haluaa korjata löydetyt haavoittuvuudet ennen tietyn sovellusversion julkaisua. Yksi haastateltavista kuitenkin oli

sitä mieltä, että projektien aikataulutuksessa sprinttipohjainen lähestymistapa voisi toimia tarkkaa päiväkohtaista allokointia paremmin. Toinen haastateltavista pohtii, että projekteja myydään monesti liian tiukalla kalenteriaikataululla ja monesti asiakkaalle voisi riittää vain se, että testaus suoritetaan tietyn kalenterikuukauden aikana sen sijaan, että luvataan tälle tietty testausviikko. Toinen konkreettinen idea aikataulutuksen toteuttamiseen on, että projekti aikataulutetaan kahden viikon aikaikkunaan ja asiakkaalle luvattava valmistumisajankohta asetetaan tämän aikaikkunan lopusta viikon päähän, että projektin mahdollinen viivästyminen ei aiheuttaisi ongelmia asiakkaalle.

> ”Meillähän ei oo mitään syytä tarjota asiakkaalle sitä, että meillä on se viisi päivää [viisi kalenteripäivää testauksen suorittamiseen].” (Haastateltava E)

> ”Must toi Kanban on niinku tässä se järkevä malli, ku ei periaattees ei pysty tehä ku yhtä asiaa kerrallaan.” (Haastateltava C)

Sprinteissä kohdeyrityksen kohdalla tulee ottaa huomioon se, että sprintin lopuksi ei välttämättä voida aina julkaista kuten ketterässä ohjelmistokehityksessä tehdään. Yksi haastateltavista huomauttaa, että ei ole järkevää toimittaa asiakkaalle vain osaa löydöksistä, kun kokonainen raportti on tulossa testauksen lopuksi. Yksi testaajista näkee Kanban-menetelmän sprinttien soveltamista parempana vaihtoehtona siksi, että testaustyö on luovaa ja testaaja voi keskittyä vain yhteen projektiin kerrallaan. Sprinttien tapauksessa hän pelkää sitä, että menetelmä vaatii usean projektin työstämisen samaan aikaan.

> ”Jos miettii jotain ketterii menetelmii niinku vaik jos, jos meidän niinku projektit kestää ylipäätään vaan viikon, … ni siihen on vähän vaikee ympätä jotain sellasii niinku et must täs on jo niinku tehty ne hommat niinku selkeesti alotukses mietitään mitä ollaan tekemässä ja miks tehään ja miten tehään ja mihin tehään, sit se tehdään ja sit sen jälkeen niinku katotaan niinku tulokset ja käydään läpi mikä meni hyvin ja mikä meni huonosti.” (Haastateltava C)

> ”Siinähän [jatkuvassa testauksessa] se vois ehkä toimiikki se niinku sprinttijuttu.” (Haastateltava C)

Kahden testaajan mielestä sprintit voisivat toimia paremmin jatkuvassa testauksessa, jossa samaa kohdetta testataan säännöllisesti ja löydöksiä raportoidaan asiakkaalle sovitun aikataulun mukaisesti. Yhden testaajan mielestä taas tähän toimii luultavasti paremmin Kanban-tyylinen jatkuva lähestymistapa, ja yksittäisiä projekteja, joissa kohde testataan projektiluontoisesti kerran ilman sopimusta tulevasta testauksesta, olisi parempi toteuttaa puolestaan sprintteinä.

6.5 Kommunikaatio ja palaverit

> ”Siis kommunikaatio toimii ihan ookoo, mutta mä uskon et se toimis paremmin semmosella tiimijaottelulla.” (Haastateltava D)

> ”Se on ainakin hyvä tääl … että voi sanoo sillai että tää ei oo mun mielest hyvä idea niin sit sua ainakin kuunnellaan.” (Haastateltava F)

Haastateltavien mielestä yrityksen kommunikaatiovälineet ovat kunnossa ja etenkin Microsoft Teamsin käyttöönoton on koettu lisänneen sisäistä viestintää. Testaajat kokevat, että heidän välillään kommunikaatio toimii hyvin ja ilmapiiri on hyvä. Yksi testaajista uskoo tiimien parantavan kommunikaatiota edelleen, koska tiimeissä on helpompi järjestää statuspalavereja kuin nykyään, sillä kun testaajat on ripoteltu eri projekteihin, palaverien sovittaminen kaikkien kalenteriin on vaivalloista ja usean lyhyen palaverin järjestämiseen kuluu enemmän aikaa kuin yhden tiimikohtaisen palaverin järjestämiseen.

> ”Meillä käytetään hirveesti tekstimuotoista viestintää … hyvä puoli on siinä, että niihin tekstillä tuleviin asioihin ei tarvitse reagoida välttämättä niinku heti … mutta siinä häviää huomattava osa siitä kommunikaatiosta, joka mahdollistaa taas väärinkäsitysten tekemisen tai tapahtumisen.” (Haastateltava E)

> ”Kommunikaatiovälineiden oikeanlainen käyttö ois ehkä yks millä sitä [sisäistä kommunikaatiota] pystyis parantamaan … Jos sulla on sellanen tilanne et sä pystyt puhuu puhelimessa niin se on aina parempi puhuu puhelimessa.” (Haastateltava E)

Kaikki ovat yhtä mieltä siitä, että samassa tilassa työskentely on tärkein hyvän kommunikaation edistäjä, sillä samassa tilassa kynnys tiedon jakamiseen on matalampi. Eräs testaajista kokee, että yrityksessä käytetään liikaa tekstimuotoista ja asynkronista viestintää, ja kokee että kommunikaatio parantuisi ja väärinkäsitykset vähenisivät, jos kommunikointi suoritettaisiin ensisijaisesti kasvotusten, toissijaisesti puhelimitse ja vasta viimeisenä viestillä.

> ”Olis erikseen vaikka joka päivä semmonen pieni lyhyt [palaveri] et puhutaan niistä [projekteista] mitä nyt tehään, mut sit ois vaikka kerran viikossa semmonen yhteinen briiffi et mietitään mitä on tulossa.” (Haastateltava D)

Muusta kommunikaatiosta mielipiteet eriävät jonkin verran. Eräs haastateltavista on havainnut, ettei asiakaspalavereista tehdä aina muistiinpanoja. Tämä johtaa hänen mukaansa siihen, että sovitut asiat unohtuvat testaajilta ja ne kommunikoidaan muille testaajille puutteellisesti. Haastateltavan mukaan ei pitäisi tuudittautua siihen, että asiakas täyttää testauksen esitietolomakkeen, vaan palavereista tulisi aina tehdä myös omat muistiinpanot. Toisen haastateltavan mukaan tulevista projekteista on epätietoisuutta, ja hän ehdottaa tämän ratkaisemiseksi viikoittaista palaveria, jossa käydään tiimin kesken tulevat projektit läpi. Kolmas pohtii, että tulevia projekteja voisi käsitellä kuukausittain.

> ”Mä olin alkuun vähän silleen että mitä … näillä tyypeillä [projektipäälliköillä] tekee ihan rehellisesti ku mä tulin koska mä olin tottunu siihen edellisissä duuneissa et mä joudun tekee sen kaiken mitä PM:t [projektipäälliköt] tekee ni ite plus sen mitä mun duuni on, mutta sit mä niinku vähän ajan päästä oon ruvennu niinku arvostaan sitä et … mun ei oikeesti tarvi miettii mitään et joku niinku järjestää kaiken tiäksä siin ympärillä et mun täytyy vaan testata tunnukset ja tälleen.” (Haastateltava D)

> ”Meillä on allokoitu projektimanagerit niille projekteille jotka hoitaa sen aikataulutuksen se on mun mielestä hyvin selkee hyvä konsepti. … Mä koen että mä voin käyttää aikani johonkin järkevämpään.” (Haastateltava E)

> ”Me [testaajat] voidaan keskittyy siihen meidän omaan asiaan eli penaamiseen [testaamiseen] ja muut keskittyy sit siihen hallinnolliseen et meidän ei tartte välittää siitä et mitä enemmän mennään siihen suuntaan ni sitä parempi koska se tarkottaa sitä että me voidaan keskittyy työntekoon ja ne keskittyy sit taas hallinnointiin jotka osaa hallinnoida.” (Haastateltava F)

Lähes kaikki haastateltavat ovat sitä mieltä, että projektipäälliköiden huolehtiessa asiakaskommunikaatiosta ja aikataulutusasioista testaajille jää enemmän aikaa heidän ydinosaamiseensa liittyviin tehtäviin. Yksi testaajista on kuitenkin sitä mieltä, että projektipäälliköt ovat turhia ja hankaloittavat kommunikaatiota. Hänen mielestään jonkun testaajista tulisi hoitaa myös projektipäällikön tehtäviä, ettei testaajien tarvitsisi ensin kommunikoida projektipäällikölle, joka välittää viestin asiakkaalle. Toinen testaaja mainitsi, että projektipäälliköt olisi hyvä sulauttaa nykyistä tiiviimmin osaksi tiimiä, että nämä pysyisivät jatkuvasti ajan tasalla testauksesta.

> ”Palaverit ja tällaset ni ne haittaa työtä jonkun verran. Ne katkasee sun ajatuksen ja jos sä teet varsinki niinku penaa [tietoturvatestausta] … se häiritsee.” (Haastateltava F)

> ”Suurin ongelma alotuspalsuissa oli aikasemmin se, että me oltiin hyvin hanakkaita menemään asiakkaalle paikan päälle ja siihen kulu sitä aikaa sit ihan hirveen paljon.” (Haastateltava E)

> ”Semmonen ideaalitilanne olis se et niis [palavereissa] ei tarttis hirveesti penaajien [testaajien] käydä … se että niit ei olis ihan hirveesti viikossa … kaikki [palaverit] samana päivänä … sä voisit keskittyä työn tekoon niinä muina päivinä.” (Haastateltava F)

Testaajien mukaan palaverit haittaavat testaustyötä jonkin verran, sillä ne katkaisevat työpäivän ja testaus on luovaa työtä, joka sietää keskeytyksiä huonosti. Raportointivaiheessa keskeytysten ei koeta olevan yhtä häiritseviä ja yksi testaajista erittelee, että yksinkertaisempaa järjestelmää testatessa on helpompi sietää työn keskeytymistä kuin monimutkaisemmissa projekteissa. Testaajat ovat mielissään siitä, että nykyään suuri osa asiakaspalavereista järjestetään etäpalavereina, sillä tämä vähentää palavereihin kuluvaa aikaa poistamalla matkustusajat, jotka usein jopa kaksinkertaistavat palaveriin käytettävän ajan. Testaajat toivovat, että palaverit sijoiteltaisiin mahdollisuuksien mukaan samoille päiville niin, että työviikkoon jäisi mahdollisimman paljon keskeytyksetöntä aikaa testaukselle, joka on testaajien pääasiallinen työtehtävä.

> ”Ehottomasti semmonen [henkilö asiakaspalaveriin] joka niinkun kysyy oikeet kysymykset, joka sit jaksaa viestii ne eteenpäin sille tiimille.” (Haastateltava C)

Siitä, millä perusteella asiakaspalavereihin tulisi valita tekninen asiantuntija, on hyvin yhteneviä mielipiteitä. Vallitseva mielipide on, että palavereihin tulisi valita ensisijaisesti

sellainen henkilö, jolla ei ole jo valmiiksi kiire. Yhden testaajan mukaan palaveriin olisi parempi irrottaa testaaja, jolla on parhaillaan työn alla melko yksinkertainen projekti eikä kovin laajan järjestelmän testaus, sillä yksinkertainen testaus kestää paremmin keskeytyksiä. Toinen testaaja lähestyy asiantuntijan valintaa sitä kautta, että palaveriin osallistujan on oltava sellainen henkilö, joka osaa kysyä asiakkaalta tarvittavat kysymykset kaiken tarvittavan tiedon keräämiseksi ja joka myös jaksaa kommunikoida nämä asiat eteenpäin muille tiimin jäsenille. Tämän ei välttämättä esimerkiksi projektin aloituspalaverissa tarvitse olla tekninen asiantuntija.

> ”Kiertääkö vuoro vai onko se aina sama tyyppi [joka käy asiakaspalavereissa] ni mun mielestä se tiimi voi päättää sen.” (Haastateltava G)

> ”Joidenkin ihmisten meilläkin niin ei ehkä tarvi olla niin paljon palavereissa kun toisten jos ei ne haluu ja ne ei oo siinä ihan elementissään ni ei niinku väkisin, mut mun mielestä kaikkien tulis aina välillä osallistua niihin.” (Haastateltava D)

Palavereissa käyvän testaajan on hyvä toimia haastateltavien mielestä vastaavana testaajana kyseisessä projektissa, jolloin tämä voi olla kontakti asiakkaan suuntaan teknisissä asioissa. Kiertävässä palaverivuorossa tunnistetaan se hyvä puoli, että kaikkien käydessä palavereissa säännöllisesti taito asiakaspalaverissa toimimisesta pysyy yllä ja esimerkiksi sairastapauksessa korvaava tekninen asiantuntija on helpompi löytää. Jos palavereissa käy aina sama henkilö, on riskinä että kun tämä henkilö ei ole saatavissa, yrityksessä ei ole saatavissa sellaista henkilöä, joka kykenee hoitamaan asiakaspalaverit niin, että kaikki tarvittavat asiat käsitellään varmasti ja saadaan kaikki tarvittava tieto testausta ja laajuuden arviointia varten.

> ”Se [päivittäinen statuspalaveri] vois olla vaan silleen että, et kertoo et mitä projektia vaikka tällä hetkellä tekee tai juurikin että jos jotain vaikka raportoi tai tarvii jonkun että … kuka vois oikolukee tämän et siin vois myös käydä niitä jo etukäteen niinkun et missä vaik tarvii jeesii.” (Haastateltava A)

> ”Mun ei tarvii kuulla projekteista, joissa mä en oo mukana.” (Haastateltava B)

> ”Jos haluu niinku tämmöstä yleiskuvaa kaikesta ni PM:t [projektipäälliköt] voi pitää jonkun oman päivittäisen Huddlensa mut jättää siit penaajat [testaajat] tekemään duuniansa.” (Haastateltava F)

Kuten Kanban-taulun, suurin osa näkee kohdeyrityksessä päivittäin pidettävän kaikille yhteisen Huddle-palaverin projektipäälliköille ja COO:lle suunnatuksi statuspalaveriksi. Huddle on päivittäinen statuspalaveri, jossa kaikki kokoontuvat yhteen juuri ennen lounasta ja jokainen kertoo vuorotellen, mitä on päivän aikana tekemässä. Päivittäisten työtehtävien sijaan Huddlessa pitäisi haastateltavien mukaan keskittyä enemmän siihen, mitä ongelmia kukin on kohdannut. Lisäksi Huddleissa tulisi haastateltavien mielestä pyytää apua esimerkiksi, kun tarvitsee jonkun muun mielipidettä tai tuntuu että on liian kiire ja kaipaa apua. Tiimikohtaiset Huddlet ovat haastateltavien mielestä pääosin hyvä

idea, ja niiden etuna nähdään etenkin se, että pienemmän porukan kesken on aikaa pureutua tarkemmin kunkin tiimin jäsenen tehtäviin ja ongelmiin eikä tarvitse kuunnella asioita, jotka eivät ole omalle tiimille relevantteja. Haittapuolena esiin nostettiin se, että ilman kaikille yhteistä Huddlea kuva koko yrityksen tilanteesta voi hämärtyä. Toinen haastateltava ehdotti, että jos yleiskuva kaikista projekteista halutaan, se voidaan koota projektipäälliköiden kesken tiimien omien Huddle-palaverien perusteella niin, että testaajien ei tarvitse tähän osallistua.

> ”Ne [internal exit -palaverit] pitää pitää heti … muuten ne vie turhaa aikaa jossain kalenterislotissa … sä unohdat koko projektin sen jälkeen, kun se on tehty ja sulla alkaa joku toinen projekti … ne pitäis mun mielestä yhdistää loppupalsuun.” (Haastateltava F)

Projektin lopuksi pidettävä internal exit -palaveri koetaan myös palaveriksi, joka tuottaa johtajille tietoa, jota nämä voivat käyttää päätöksenteon tukena. Testaajat eivät kuitenkaan ole kokeneet palavereissa täytettyjen lomakkeiden johtaneen konkreettisiin toimiin, eikä ajankäyttöä näihin palavereihin nähdä järkevänä, jos palaverien tulokset eivät johda konkreettisiin toimiin. Vaikka projektin laajuuden arvioinnissa on ollut ongelmia, internal exitissä asian toteamisesta ei koeta syntyvän arvoa, sillä kukin projekti on ainutlaatuinen. Yhdessä haastattelussa pohdittiin, onko palaveri tarpeellinen jokaisen projektin yhteydessä, vai voisiko lomakkeen täyttää vain silloin kun projektissa on tapahtunut jotain merkittävää, joka halutaan dokumentoida. Suurin osa on sitä mieltä, että internal exitit on kuitenkin hyvä olla olemassa ja ne kannattaa pitää nimenomaan projektikohtaisesti.

7 TOIMINNAN KEHITTÄMINEN

Tässä kappaleessa kerrotaan, kuinka toimintaa on kehitetty kohdeyrityksessä tutkimuksen puitteissa. Kehityksessä on otettu huomioon tutkimuksen aiempien vaiheiden löydökset ja kehityksen ensimmäinen askel oli päätös tiimijaon implementoinnista ja tähän liittyvistä reunaehdoista, jotka johtoryhmä määritteli syksyllä 2019. Reunaehtojen määrittelyssä on käytetty tämän tutkimuksen aiempien vaiheiden tuottamaa tietoa päätöksenteon tukena.

Kohdeyrityksessä siirryttiin toteuttamaan tietoturvatestausta tiimeinä tammikuussa 2020. Tässä tutkimuksessa tarkastellaan pääasiassa PEPE-tiimiksi nimettyä tiimiä, joka vastaa projektimuotoisten tietoturvatestausprojektien toteuttamisesta. Tiimi on osittain päällekkäinen JAPE-tiimin kanssa, jonka vastuulla on jatkuva tietoturvatestaus. JAPE-tiimi tekee ensisijaisesti jatkuvaa testausta, mutta sen jäsenet työskentelevät PEPE-tiimissä silloin kun jatkuvaa testausta ei tarvitse tehdä.

7.1 Tiimien implementointi

PEPE- ja JAPE-tiimit kokoontuivat tammikuussa 2020 ja sopivat yhdessä PEPE-tiimin toiminnan yksityiskohdista. Tilaisuudessa oli tarkoitus sopia yhdessä työskentelytavoista, jotka ovat tiimille sopivia ja jotka täyttävät johtoryhmän asettamat reunaehdot. Tutkija fasilitoi keskustelua tuoden esiin yksilöhaastatteluissa esiin nousseita näkemyksiä ja varmisti, että kaikista reunaehdoista keskustellaan ja niiden täyttämisestä sovitaan yhdessä tiimin kanssa.

Ketterille menetelmille tyypillisesti projektien kustannukset ja aika pyritään lukitsemaan, kuten Measey (2015, ss. 18–20) sekä Stober ja Hansmann (2010, s. 118) ovat esittäneet. Kullekin projektille annetaan tietty määrä henkilötyöpäiviä, jonka testaamiseen voi käyttää. Tiimin jäsenet pitävät näistä kirjaa ja projektipäällikön vastuulla on seurata kokonaisuutta. Koska projektiin käytetään vain tietty aika, aivan kaikkea ei välttämättä ehditä testaamaan. Testaajat arvioivat oman asiantuntemuksensa ja asiakkaan esittämien uhkaskenaarioiden ja toiveiden perusteella, mitä osia järjestelmän testauksessa priorisoidaan.

Projektit jaetaan tiimeille ja tiimit päättävät itse, kuka tekee mitäkin projektia. Tällä pyritään kannustamaan tiimiä itseohjautuvuuteen, joka on yksi ketterien tiimien toiminnan kulmakivistä (Project Management Institute, 2017, s. 39). Tiimeissä on sekä testaajia

että projektipäällikkö, joka vastaa projektien koordinoinnista. Projektipäällikkö ei ole esimiesasemassa vaan yksi tasavertaisista tiimin jäsenistä. Tiimin sisällä valitaan jokaiseen projektiin lead tester, joka vastaa testauksesta ja osallistuu ensisijaisesti projektiin liittyviin asiakaspalavereihin. Yhdelle asiakkaalle toteutettaviin testausprojekteihin pyritään valitsemaan aina sama lead tester, mutta kaikkien projektien välillä lead testerin vastuita pyritään jakamaan tasaisesti kaikille testaajille. Asiakaspalavereihin osallistuu lähtökohtaisesti yksi testaaja tai tekninen asiantuntija, ei kaikki projektissa työskentelevät henkilöt, että mahdollisimman moni saa keskittyä työn tekemiseen palavereissa istumisen sijaan.

Työtä visualisoidaan korkealla tasolla vanhaan tapaan Kanban-taululla muun muassa Kochin (2005, s. 80) suositusten mukaan niin, että kaikki voivat seurata työn etenemistä. Taulua kuitenkin muutetaan niin, että laput jaotellaan tiimikohtaisiin riveihin. Lappuihin merkitään käytettävissä oleva kokonaistyömäärä henkilötyöpäivinä. Työn jaksotusta ei muuteta ainakaan vielä tässä vaiheessa inkrementaaliseksi, vaan projektien toteuttamista jatketaan aiempaan tapaan jatkuvana toteutuksena niin, että uutta työtä otetaan työn alle edellisen valmistuessa. Tarkemmalla tasolla työtä visualisoidaan hyökkäyspuun avulla. Jokaisesta projektista piirretään puu, johon merkitään testatut ja testattavat kohteen osat ja mahdolliset löydökset. Tämän puun luomisesta vastaa lead tester. Puun avulla jaetaan työtä testaajien kesken.

Aiemmin käytössä ollut päivittäinen Huddle-palaveri korvataan tiimin omalla Daily-palaverilla. PEPE-tiimin Daily järjestetään kello 10.45, juuri ennen lounasaikaa. Tämä ajankohta valittiin Measey ym. (2015, s. 75) neuvoja mukaillen siksi, että monet tulevat töihin vasta kello kymmenen aikaan ja hieman ennen yhtätoista järjestettävä palaveri mahdollistaa sen, että kaikki ehtivät kerrata edellisen päivän asiat, mutta työpäivää on vielä reilusti jäljellä niin, että Daily-palaverissa voidaan sopia päivän prioriteettitehtävistä.

Daily-palaverin kestoksi on määritelty viisitoista minuuttia ja sen aikana käydään tiimin asiat läpi projekti kerrallaan. Jos tiimin jäsen ei pääse fyysisesti paikan päälle, hän voi osallistua etäyhteydellä Microsoft Teamsin välityksellä. Jos etänä osallistuminenkaan ei onnistu esimerkiksi päällekkäisen palaverin vuoksi, tiimin Daily-kanavalle tulee laittaa viesti, jossa kertoo mitä on tekemässä kyseisenä päivänä. Jokaisesta Daily-palaverista kirjoitetaan Daily-kanavalle lyhyt kooste, josta voi tarkistaa mitä palaverissa puhuttiin. Daily-palaverin lisäksi maanantaisin järjestetään viikoittainen palaveri, jossa puhutaan tulevista projekteista.

Projektin päättyessä järjestetään retrospektiiviä vastaava internal exit -palaveri, jossa keskustellaan projektin sujumisesta ja täytetään lomake, johon keskustellut asiat dokumentoidaan eksplisiittiseen muotoon. Tämä on Project Management Instituten (2017, s. 50) mukaan ketterien menetelmien tärkein elementti, mutta projektikohtaisen internal exitin järjestäminen otettiin käyttöön jo edellisen projektinhallintatutkimuksen yhteydessä ja tämä tutkimus on antanut vahvoja suosituksia käytännön jatkamiselle. Palaveri pyritään pitämään heti asiakkaan kanssa pidettävän purkupalaverin jälkeen, mutta jos tämä ei onnistu, perjantaisin on varattu aika Daily-palaverin jälkeen, jolloin internal exit -palavereja voi pitää. Tämän ajan lisäksi internal exit -palaverin voi pitää silloin kun se projektin osallistujille sopii.

7.2 Tiimin toiminnan kehittäminen

Helmikuussa, kun tiimi oli toiminut noin kuukauden, kokoonnuttiin uudestaan keskustelemaan tiimin toiminnasta ja reflektoimaan kuluneen kuukauden tapahtumia. Tiimi kävi yhdessä läpi mikä toiminnassa oli sujunut hyvin ja mitä pitäisi kehittää. Lisäksi käytiin läpi johtoryhmän asettamat reunaehdot ja keskusteltiin niiden täyttämisestä. Tämäkin tilaisuus toteutettiin tutkijan fasilitoimana.

PEPE-tiimi on toiminut pääosin hyvin, ja testaajat ovat kokeneet, että tiimi on mahdollistanut resurssien kohdistamisen sinne, missä niitä kulloinkin tarvitaan. JAPE-tiimillä ei ole kuitenkaan ollut tarpeeksi projekteja tehtävänä ja osa heistä on kokenut sekavaksi sen, että he auttavat välillä PEPE-tiimiä ja välillä työskentelevät JAPE-tiimin projekteissa. JAPE-tiimin jäsenet ovat siis käytännössä puoliksi PEPE-tiimin jäseniä. Tämän toivotaan korjaantuvan, kun jatkuvaan testaukseen saadaan enemmän asiakkaita, sillä tuote on toistaiseksi vasta lanseerausvaiheessa.

Huddle-palaverin tilalle tullut tiimikohtainen Daily-palaveri on saanut tiimin jäseniltä positiivista palautetta, ja se on koettu Huddlea paremmaksi palaveriksi. Dailysta tekee paremman erityisesti se, että asiat käydään läpi projektikohtaisesti toisin kuin Huddlessa, jossa kukin kertoi yksitellen mitä on itse tekemässä päivän aikana. Nyt tiimi on kokenut päivittäisen palaverin helpommaksi seurata, sillä se etenee johdonmukaisesti. Aiemmin saattoi esimerkiksi käydä niin, että yhdestä projektista keskusteltiin vuorossa ensimmäisen henkilön puheenvuoron aikana ja uudestaan viimeisen henkilön kohdalla, sillä nämä työskentelivät kumpikin saman projektin parissa. Daily on kuitenkin välillä venynyt liian pitkäksi ja tiimi päätti, että he keskittyvät tulevaisuudessa löytämään niin sanotun kultaisen keskitien sille, että palaveri ei kestä liian kauan, mutta kaikki tiimin jäsenet saavat tarvittavan tiedon seuraavan vuorokauden työtehtäviään varten joko palaverissa tai heti sen jälkeen.

Tiimi on sopinut yhdessä, että jos ei pääse Daily-palaveriin mukaan, tulee tiimin Daily-kanavalle kirjoittaa viesti, jossa kertoo samat asiat kuin Dailyssa kertoisi. Tämä ei ole kuitenkaan aina toteutunut ja muu tiimi on jäänyt välillä epätietoiseksi muutamien tiimin jäsenten tekemisistä ja työn etenemisestä. Tähän liittyen tiimi on keskustellut siitä, että Daily-viesteissä tulee tulevaisuudessa muistaa kertoa paremmin muun muassa se, millä aikataululla oma työ on etenemässä.

Projektin päättyessä järjestettävän internal exitin palautelomaketta on lyhennetty ja kevennetty niin, että täytettäviä kenttiä on vähemmän. Tämän toivotaan keventävän palavereja ja tuottavan parempia vastauksia, kun ei tarvitse vastata moneen samankaltaiseen kysymykseen. Internal exitille oli aiemmin varattu aika perjantaisin Dailyn jälkeen, mutta koska tiimin jäsenillä on usein kova tarve päästä lounaalle heti Dailyn jälkeen, internal exiteille varattu aika siirrettiin ennen Dailya. Internal exitit pyritään kuitenkin edelleen pitämään heti asiakkaan kanssa pidettävän loppupalaverin jälkeen.

Turhan palavereissa istumisen vähentämiseksi tiimissä on kokeiltu sitä, että asiakaspalavereihin osallistuisi testaajista vain yksi koko projektitiimin sijaan. Tiimin mielestä tämä on toiminut muuten hyvin, mutta laajojen ja epätavallisten järjestelmien testauksen purkupalavereissa on koettu tarvittavan useampaa teknistä asiantuntijaa, sillä monimutkaisia havaintoja on vaikea selittää asiakkaalle, saati vastata heidän tarkentaviin kysymyksiinsä, jos havaintoa ei ole löytänyt tai testannut itse. Eräs tekninen asiantuntija toi esiin myös sen, että usein testauksen purkupalaveri on se hetki, kun projektin arvo realisoituu asiakkaalle, joten siksi palaveriin on hyvä panostaa eikä aina ole järkevää pyrkiä minimoimaan palaveriin käytettyjä työtunteja. Tiimi totesi yhdessä, että lähtökohtaisesti pyritään edelleen hoitamaan purkupalaverit yhden teknisen asiantuntijan voimin, mutta tarpeen vaatiessa otetaan mukaan muita asiantuntijoita tapauskohtaisen harkinnan perusteella.

Projektien aikatauluja on hallittu pitämällä kirjaa projektiin käytetyistä henkilötyöpäivistä. Testaajille käytettävissä olevat työpäivät on kommunikoitu merkitsemällä ne kanban-taulun projektilappuihin, tiimin yhteisen Teams-kanavan Planner-sovelluksen projektikortteihin ja muistuttelemalla näistä Dailyissa. Tiimin jäsenten mielestä projektin aikataulujen kommunikointi henkilötyöpäivinä on helpottanut ymmärtämään, kuinka paljon aikaa testaamiseen voi käyttää. Tiimi on kokenut, että näistä on hyvä pitää tarkempaa kirjaa, joten tiimi sopi, että työpäivät merkitään projektin hyökkäyspuuhun. Tiimi huomasi, että projektinhallintaan käytettyjä työpäiviä ei ole laskettu tiimille annettuun työmäärään, ja niistä on pidettävä tulevaisuudessa kirjaa tarkemmin. Koska henkilötyöpäivien laskennasta on ollut hieman epäselvyyksiä, todettiin että työmääriä koskevat asiat dokumentoidaan eksplisiittisesti kaikkien saataville väärinymmärrysten minimoimiseksi.

Projekteista on ryhdytty piirtämään hyökkäyspuita, jotka visualisoivat testattavaa kohdetta tarjouksessa määritellyn projektin laajuuden puitteissa. Tiimi on sopinut, että kullakin projektilla on lead tester, joka piirtää ensimmäisen version hyökkäyspuusta ensimmäisenä testauspäivänä ja merkitsee puuhun tehtäviä, joita voidaan jakaa tiimin jäsenten kesken. Tiimi on kokenut, että hyökkäyspuun käytöstä on ollut apua tehtävien jakamisessa. Yhtä mieltä oltiin kuitenkin siitä, että tehtävien jakaminen on hankalaa ja tulee tehdä aina projektikohtaisesti, eikä yhtä oikeaa tapaa jakaa kaikkia projekteja tehtäviin samalla tavalla välttämättä ole edes olemassa. Tehtävien jakamisen on tarkoitus vähentää päällekkäistä työtä. Tiimi kuitenkin totesi, että koska testaajien osaamisessa on eroavaisuuksia, on perusteltua tehdä tiettyjä päällekkäisiä asioita. Tästä esimerkkinä nostettiin cross site scripting -haavoittuvuuksien etsintä testattavista kohteista. Tehtävien jakamisen tarkoitusta muotoiltiinkin tiimin kesken uudestaan niin, että sen tarkoituksena on välttää tahatonta päällekkäisen työn tekemistä.

Hyökkäyspuun muodosta oli myös keskustelua. Nyt hyökkäyspuita on piirretty niin, että ne kuvaavat asiakkaan järjestelmää ja sen komponentteja, eli vaikka niitä on kutsuttu hyökkäyspuiksi, ne ovat oikeammin työkaluja kohteen analysointiin, eivätkä suoraan hyökkäysten suunnitteluun. Tiimi päätti kokeilla rakentaa hyökkäyspuita myös siltä kannalta, että edetään hyökkäys edellä ja puussa kuvataan eri reittejä järjestelmään murtautumiseksi järjestelmän sisältämien osien sijaan. Sopivan puun valinta perustuu asiakkaan tarpeeseen, sillä ensin mainittu puu on testaajien mielestä hyödyllisempi, kun on tarkoituksena skannata koko järjestelmä ja arvioida sen tietoturvaa kokonaisuutena, kun taas määritelmän mukainen hyökkäyspuu on oiva työkalu, jos tavoitteena on vain päästä murtautumaan järjestelmään.

8 YHTEENVETO

Tässä luvussa esitetään yhteenveto tehdystä tutkimuksesta, sekä tarkastellaan tutkimuksen tavoitteiden saavuttamista ja tutkimuskysymyksiin vastaamista. Tutkimusta arvioidaan kriittisesti ja lopuksi nostetaan esiin tutkimuksen aikana ilmenneitä mahdollisia jatkotutkimuskohteita.

8.1 Tulosten yhteenveto

Tämän tutkimuksen tavoitteena oli parantaa tietoturvatestausprojektien toteuttamista työntekijöiden näkökulmasta niin, että päivittäinen työnteko tuntuu vähemmän kiireiseltä ja stressaavalta. Tutkimus rajattiin koskemaan vain yrityksessä toteutettavia teknisiä tietoturvatestausprojekteja, sillä niiden osuus yrityksen toteuttamista projekteista on suurin sekä lukumäärässä, liikevaihdossa että henkilötyömäärässä mitattuna.

Tutkimuksessa haettiin kirjallisuudesta suosittuja ketteriä menetelmiä, joista tunnistettiin elementtejä kuten itseohjautuvat tiimit, päivittäiset palaverit, retrospektiivit, kommunikointi kasvotusten, työn suunnittelu tarkasti vain lähitulevaisuuteen, työn visualisointi ja projektin laajuuden määräytyminen ennalta lukittujen aika- ja kustannusrajoitteiden mukaan. Tällä saatiin vastaus ensimmäiseen alatutkimuskysymykseen, eli millaisia ketteriä menetelmiä on olemassa ja mitä elementtejä niissä on. Näihin elementteihin liittyviä asenteita ja mielipiteitä sekä yrityksen nykytilaa kartoitettiin yksilöhaastatteluilla, joiden löydökset vastasivat toiseen alatutkimuskysymykseen, eli mitä mieltä kohdeyrityksen työntekijät ovat tietoturvatestausprojektien nykytilasta ja niiden kehittämisestä.

Yksilöhaastattelujen löydösten perusteella suunniteltiin ensimmäisiä kehitystoimenpiteitä, kuten tiimit. Tiimien aloittaessa toimintansa toimintatavoista sovittiin yhdessä tiimin kesken ja kehitystä jatkettiin toimintatutkimuksen spiraalin mukaisesti suunnitellen, toteuttaen ja arvioiden kehitystoimenpiteitä yhdessä tiimien kanssa. Tämän kokonaisuuden avulla saatiin vastaus päätutkimuskysymykseen, eli miten kohdeyrityksen tietoturvatestausprojektien toteuttamista voidaan parantaa työntekijöiden näkökulmasta ketterien menetelmien elementein. Yksi suurimmista parannuksista tämän tutkimuksen puitteissa oli tiimien implementointi, sillä niitä oli työntekijöiden puolesta toivottu kovasti ja ne nähtiin ketterissä menetelmissä yhtenä keskeisimmistä elementeistä. Tämän lisäksi tiimit päättivät itse keskenään tiimin toimintatavoista ja säännöllisesti järjestettävistä palavereista, kuten päivittäisistä palavereista ja retrospektiiveistä.

Tutkimuksen myötä kohdeyrityksen teknisten tietoturvaprojektien toteuttamiseen saatiin tehtyä merkittäviä muutoksia, jotka tehtiin työntekijöiden toiveiden ja mielipiteiden perusteella. Työntekijöiden päätäntävaltaa päivittäisessä työskentelyssä pyrittiin lisäämään, jotta he voisivat vaikuttaa enemmän siihen, millaista arki yrityksessä on. Työntekijöiden stressitasoja ja kiireen kokemusta ei ole erikseen mitattu, mutta voidaan olettaa, että työntekijät ovat kannattaneet sellaisia muutoksia, jotka vähentävät stressin ja kiireellisyyden tunnetta työympäristössä. Tällöin voidaan sanoa, että tutkimuksen tavoitteet on saavutettu. Muutosten vaikutukset tullaan kuitenkin näkemään paremmin vasta tulevaisuudessa, kun ne juurtuvat syvemmälle osaksi yrityksen arkea. Toimintatutkimuksen spiraalin mukainen kehityssykli tulee jatkumaan tämän tutkimuksen ulkopuolella, ja kehitystoimia tullaan tekemään edelleen tiimien mielipiteisiin ja kokemuksiin perustuen. Tutkimuksen yhtenä tuloksena voidaankin nähdä myös se, että yrityksessä on ymmärretty jatkuvan kehityksen tarpeellisuus ja siihen on sitouduttu tämän tutkimuksen aikana aiempaa paremmin.

8.2 Tutkimuksen arviointi

Tutkimuksen uskottavuutta ja luotettavuutta tukee se, että tutkimuksen metodologiset valinnat ovat liiketoimintatutkimukselle ominaisia ja tukevat muutosta kohdeyrityksessä. Myös kohdeyrityksen ja tutkijan välisellä työsuhteella on vaikutusta, ja se näkyi varsinkin haastattelutilanteissa, sillä työyhteisön jäsenenä tutkijan ja haastateltavien välisillä sosiaalisilla suhteilla voidaan olettaa olevan vaikutusta haastattelutilanteiden kulkuun. Tutkijan asema osana työyhteisöä kuitenkin auttoi tutkijaa tunnistamaan yrityksessä vallitsevia sosiaalisia rakenteita. Tutkimuksen luotettavuutta olisi voinut parantaa useamman empiirisen menetelmän hyödyntäminen, mutta tässä tutkimuksessa menetelmät rajoittuivat yksilö- ja ryhmähaastatteluihin.

Eskolan ja Suorannan (1998) mukaan laadullinen tutkimus on avoimesti subjektiivinen, minkä vuoksi luotettavuuden arviointi koskee koko tutkimusprosessia. Koska tämä tutkimus on toimintatutkimus ja tutkija on osa tutkittavaa kohdetta, tulkintojen subjektiivisuus tiedostetaan ja hyväksytään. Tutkija pyrki kuitenkin välttämään puolueellisuutta kaikissa tulkinnoissa ja tuomaan esiin tasapuolisesti niin tutkijan oman näkemyksen mukaisia kuin siitä poikkeavia löydöksiä haastatteluissa. Haastattelujen nauhoitteita kuunneltiin useaan kertaan tarkasti pyrkien varmistamaan, että tutkimukseen voitiin tuoda haastateltavan esittämät asiat mahdollisimman tarkasti ja että tutkijan tulkinta heijastuisi nauhoitettujen haastattelujen löydöksiin mahdollisimman vähän.

Gubrium ym. (2012) mukaan haastattelututkimuksen aineiston laatuun vaikuttaa merkittävästi vastausten puolueellisuus. Haastattelija voi omilla kommenteillaan, äänenpainoillaan tai käytöksellään vaikuttaa siihen, miten haastateltavat vastaavat kysymyksiin. Tämä otettiin haastattelutilanteissa huomioon niin, että haastattelija kiinnitti erityistä huomiota käytökseensä ja pyrki toimimaan tilanteessa mahdollisimman neutraalisti. Kaikki haastattelut järjestettiin kohdeyrityksen tiloissa, että paikka olisi kaikille osapuolille käytännöllinen ja mahdollisimman neutraali (Saunders ym., 2009, s. 329).

Haastateltava voi antaa vajaita vastauksia, jos tämä ei pidä haastattelijaa luotettavana, ja tutkija voi myös tulkita vastauksia puolueellisesti oman näkemyksensä kautta (Gubrium ym., 2012). Luottamuksen puutetta ei nähdä tämän tutkimuksen haastatteluissa todennäköisenä, sillä haastattelija ja haastateltavat ovat saman työyhteisön jäseniä ja täten toisilleen tuttuja. Lisäksi haastateltavia rohkaistiin haastatteluissa rehellisyyteen kertomalla, että kiinnostuksen kohteena ovat haastateltavan omat mielipiteet ja että kysymyksiin ei ole olemassa oikeita tai vääriä vastauksia.

Tämän tutkimuksen otoskokoa voidaan pitää tutkimuksen laajuuteen nähden riittävänä, sillä haastattelujen aikana löydökset saatiin saturoitumaan ja valtaosa tutkimuksen piiriin kuuluvien projektien toteutukseen osallistuvista henkilöistä haastateltiin. Laajemmalla otoskoolla olisi ollut mahdollista saada vielä laajempaa näkemystä tutkimuskohteeseen, mutta koska kaikki tutkimuksen piiriin kuuluvia projekteja toteuttavat työntekijät olivat osallisina vähintään ryhmähaastatteluissa, voidaan todeta, että jokaisella on ollut mahdollisuus tuoda oma näkemyksensä esiin ja vaikuttaa omalta osaltaan tutkimuksen tuloksiin.

Tämän tutkimuksen kohteena oli vain yksi yritys, minkä vuoksi tutkimuksessa esitetyt kehityskohteet ja -toimet eivät ole yleispäteviä. Teoreettinen viitekehys on kuitenkin yleistettävissä ketterien menetelmien elementtien soveltamiseksi muissakin yrityksissä. Lisäksi tutkimuksen toteutus on pyritty kuvaamaan mahdollisimman tarkasti, että tutkimus olisi mahdollista siirtää toiseen organisaatioon ja siten toistettavissa. Tutkimuksen tulokset ja toiminnan kehittäminen kohdeyrityksessä ovat kuitenkin vahvasti sidoksissa kohdeyrityksen kontekstiin, eikä niistä voida tehdä yleispäteviä johtopäätöksiä. Tämän tutkimuksen toistaminen toisessa organisaatiossa tuottaa erilaisesta kontekstista ja subjektiivisista tekijöistä johtuen todennäköisesti ainakin osittain erilaisia tuloksia.

8.3 Tulevaisuuden tutkimuskohteet

Tulevaisuudessa kohdeyrityksessä suositellaan jatkettavan toiminnan jatkuvaa kehittämistä työntekijöiden näkemykset huomioon ottaen. Tämä voidaan toteuttaa esimerkiksi

jatkamalla tässä tutkimuksessa esitellyn toimintatutkimuksen spiraalin seuraamista edelleen. Tällöin jo tehtyjä muutoksia ja niiden seurauksia voidaan tarkastella pidemmällä aikavälillä ja tehdä kehitystyötä edelleen tästä tarkastelusta saadun tiedon pohjalta. Muutosten suunnitteluun ja toteuttamiseen suositellaan systemaattisuutta ja pitkäjänteisyyttä, sillä etenkin suuret muutokset vaativat aikaa ja niiden seuraukset voivat olla nähtävissä selväsi vasta pitkän ajan kuluttua.

Tietoturvatestausprojektien lisäksi ketterien menetelmien elementtien laajentamista sovellettaviksi muissakin yrityksen toteuttamissa asiakasprojekteissa ja sisäisissä projekteissa voidaan tarkastella tulevaisuudessa. Erityisesti elementtien soveltaminen jatkuvan testauksen projekteihin voi olla mielenkiintoinen tulevaisuuden tutkimuskohde kohdeyrityksessä, sillä tämän tutkimuksen puitteissa sopivasta lähestymistavasta näissä projekteissa oli useita erilaisia mielipiteitä ja vaikka näissä projekteissa on paljon yhtäläisyyksiä kertaluontoisten tietoturvatestausprojektien kanssa, niiden pituus on huomattavasti suurempi ja toteutuksessakin on huomattavia eroavaisuuksia.

LÄHTEET

Abrahamsson, P., Oza, N., & Siponen, M. T. (2010). Agile Software Development Methods: A Comparative Review. Teoksessa T. Dingsøyr, T. Dybå, & N. B. Moe (Toim.), *Agile Software Development: Current Research and Future Directions* (ss. 31–59). Springer Berlin Heidelberg.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods: Review and analysis*.

Ahimbisibwe, A., Daellenbach, U., & Cavana, R. Y. (2017). Empirical comparison of traditional plan-based and agile methodologies. *Journal of Enterprise Information Management*, *30*(3), 400–453.

Allsopp, W. (2009). *Unauthorised Access: Physical Penetration Testing for IT Security Teams* (1. p.). John Wiley & Sons, Incorporated.

Alqudah, M., & Razali, R. (2018). An Empirical Study of Scrumban Formation based on the Selection of Scrum and Kanban Practices. *International Journal on Advanced Science, Engineering and Information Technology*, *8*(6), 2315–2322.

Antunes, N., & Vieira, M. (2014). Penetration Testing for Web Services. *Computer*, *47*(2), 30–36.

Ashmore, S., & Runyan, K. (2015). *Introduction to agile methods*. Pearson Education.

Ashraf, S., & Aftab, S. (2017). IScrum: An Improved Scrum Process Model. *International Journal of Modern Education and Computer Science; Hong Kong*, *9*(8), 16–24.

Beck, K. (2004). *Extreme Programming Explained: Embrace Change* (2. p.). Addison-Wesley.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., & others. (2001). *Manifesto for agile software development*.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, *11*(3), 369–386.

Burton-Spall, M., Smith, R., Bird, J., & Bell, L. (2017). *Agile Application Security*. O'Reilly Media, Inc.

Chóliz, J., Vilas, J., & Moreira, J. (2015). Independent Security Testing on Agile Software Development: A Case Study in a Software Company. *2015 10th International Conference on Availability, Reliability and Security*, 522–531.

Cockburn, A. (2006). *Agile Software Development: The Cooperative Game* (2. p.). Addison-Wesley Professional.

Coghlan, D., & Brydon-Miller, M. (2014). *The SAGE Encyclopedia of Action Research*. SAGE Publications Ltd.

Cohen, D., Lindvall, M., & Costa, P. (2004). An Introduction to Agile Methods. Teoksessa *Advances in Computers* (Vsk. 62, ss. 1–66). Elsevier.

Coram, M., & Bohner, S. (2005). The impact of agile methods on software project management. *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, 363–370.

Davis, B. (2012). *Agile Practices for Waterfall Projects: Shifting Processes for Competitive Advantage* (1. p.). J. Ross Publishing.

Davison, R., Martinsons, M. G., & Kock, N. (2004). Principles of canonical action research. *Information Systems Journal*, *14*(1), 65–86.

Dingsøyr, T., Dybå, T., & Moe, N. B. (2010). Agile Software Development: An Introduction and Overview. Teoksessa T. Dingsøyr, T. Dybå, & N. B. Moe (Toim.), *Agile Software Development: Current Research and Future Directions* (ss. 1–13). Springer Berlin Heidelberg.

Dowd, M., McDonald, J., & Schuh, J. (2007). *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities* (1. p.). Addison-Wesley Professional.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859.

Ellis, G. (2016). Chapter 8 - Agile Project Management: Scrum, eXtreme Programming, and Scrumban. Teoksessa G. Ellis (Toim.), *Project Management in Product Development* (ss. 223–260). Butterworth-Heinemann.

Engebretson, P. (2013). *Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy* (2. p.). Elsevier Science & Technology Books.

Equifax. (2017). *Equifax Releases Details on Cybersecurity Incident, Announces Personnel Changes*. https://investor.equifax.com/news-and-events/news/2017/09-15-2017-224018832

Eriksson, P., & Kovalainen, A. (2008). *Qualitative Methods in Business Research*. SAGE Publications Ltd.

Eskola, J., & Suoranta, J. (1998). *Johdatus laadulliseen tutkimukseen* (1. p.). Osuuskunta Vastapaino.

FIRST.org. (2020). *Common Vulnerability Scoring System SIG*. FIRST — Forum of Incident Response and Security Teams. https://www.first.org/cvss

Ghani, I., Azham, Z., & Seung Ryul Jeong. (2014). Integrating Software Security into Agile-Scrum Method. *KSII Transactions on Internet & Information Systems*, *8*(2), 646–663.

Gubrium, J., Holstein, J., Marvasti, A., & McKinney, K. (2012). *The SAGE Handbook of Interview Research: The Complexity of the Craft*. SAGE Publications, Inc.

Hammarberg, M., & Sundén, J. (2014). *Kanban in action*. Manning Publications.

Highsmith, J. A., & Highsmith, J. (2002). *Agile Software Development Ecosystems*. Addison-Wesley Professional.

Hoda, R., Salleh, N., & Grundy, J. (2018). The Rise and Evolution of Agile Software Development. *IEEE Software*, *35*(5), 58–63.

Hohl, P., Klünder, J., van Bennekum, A., Lockard, R., Gifford, J., Münch, J., Stupperich, M., & Schneider, K. (2018). Back to the future: Origins and directions of the “Agile Manifesto” – views of the originators. *Journal of Software Engineering Research and Development*, *6*(1), 15.

Holcombe, M. (2008). *Running an Agile Software Development Project* (1. p.). John Wiley & Sons, Incorporated.

Hunt, J. (2006). Feature-driven development. Teoksessa *Agile Software Construction* (ss. 161–182). Springer.

Ince, D. (2013). Attack tree. Teoksessa *A Dictionary of the Internet* (3. p., Vsk. 2013). Oxford University Press.

Johnson, L. R. (2017). *Community-Based Qualitative Research: Approaches for Education and the Social Sciences*.

Koch, A. S. (2005). *Agile Software Development: Evaluating the Methods for Your Organization*. Artech House, Inc.

Lahden kaupunki. (2019, kesäkuuta 12). *Lahden kaupungin verkkoon kohdistettu kyberhyökkäys*. Lahti.fi. https://www.lahti.fi/ajankohtaista/uutiset/lahden-kaupungin-verkkoon-kohdistettu-kyberhyökkäys

Leopold, K., & Kaltenecker, S. (2015). *Kanban Change Leadership: Creating a Culture of Continuous Improvement* (1. p.). John Wiley & Sons, Incorporated.

Measey, P., Berridge, C., Gray, A., Wolf, L., Oliver, L., Roberts, B., Short, M., & Wilmshurst, D. (2015). *Agile Foundations: Principles, Practices and Frameworks* (1. p.). BCS, The Chartered Institute for IT.

Merkow, M. S., & Raghavan, L. (2010). *Secure and Resilient Software Development*. Auerbach Publications.

Mills, J., & Birks, M. (2014). *Qualitative Methodology: A Practical Guide*. SAGE Publications, Inc.

Muniz, J., & Lakhani, A. (2013). *Web Penetration Testing with Kali Linux*. Packt Publishing, Limited.

Murray, A. P. (2016). *The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond*. John Wiley & Sons, Incorporated.

Najera-Gutierrez, G., & Ansari, J. A. (2018). *Web Penetration Testing with Kali Linux* (3. p.). Packt Publishing.

OWASP. (2020). *OWASP Top Ten*. https://owasp.org/www-project-top-ten/

Pollack, J., Helm, J., & Adler, D. (2018). What is the Iron Triangle, and how has it changed? *International Journal of Managing Projects in Business*, *11*(2), 527–547.

Pries, K. H., & Quigley, J. M. (2010). *Scrum Project Management* (1. p.). CRC Press LLC.

Project Management Institute. (2014). *Navigating complexity: A practice guide*. Project Management Institute.

Project Management Institute. (2017). *Agile Practice Guide*. Project Management Institute, Inc.

Ríos-Mercado, R. Z., & Ríos-Solís, Y. A. (Toim.). (2012). *Just-in-time systems* (60. p.). Springer.

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2009). *Research methods for business students* (5th ed). Prentice Hall.

Smith, B., Lam, K., & LeBlanc, D. (2009). *Assessing Network Security*. Microsoft Press.

Stober, T., & Hansmann, U. (2010). *Agile software development: Best practices for large software development projects*. Springer.

Tietosuojavaltuutetun toimisto. (2020). *Tietojen kalastelu*. Tietosuojavaltuutetun toimisto. https://tietosuoja.fi/tietojenkalastelu

Toikko, T., & Rantanen, T. (2009). *Tutkimuksellinen kehittämistoiminta: Näkökulmia kehittämisprosessiin, osallistamiseen ja tiedontuotantoon*. Tampere University Press.

Verizon. (2019). *2019 Data Breach Investigations Report*. Verizon Enterprise. https://enterprise.verizon.com/en-au/resources/reports/dbir/

Weidman, G. (2014). *Penetration Testing: A Hands-On Introduction to Hacking*. No Starch Press, Incorporated.

Whitaker, A., & Newman, D. P. (2005). *Penetration testing and cisco network defense: An ethical hacking handbook* (1st ed.). Cisco Press.

Wysocki, R. K., Kaikini, S., & Sneed, R. (2013). *Effective Project Management: Traditional, Agile, Extreme*. John Wiley & Sons, Incorporated.

LIITTEET

Liite A: Haastattelurunko (salattu)

Liite B: Penetraatiotestauksen vaiheet kohdeyrityksessä (salattu)