



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Agile-menetelmät

Case: ICT-alan yritys

Röyhkiö, Minna

2012 Leppävaara

Laurea-ammattikorkeakoulu
Laurea Leppävaara

Agile-menetelmät
Case: ICT-alan yritys

Röyhkiö, Minna
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Syyskuu, 2012

Röyhkiö, Minna

Agile-menetelmät Case: ICT-alan yritys

Vuosi 2012 Sivumäärä 46

Tämän opinnäytetyön tarkoituksena on ollut perehtyä yleisimpiin Agile-ohjelmistokehitysmenetelmiin sekä tutkia toimeksiantajan käyttämien Agile-menetelmien tehokkuutta ja toimivuutta yksikössään. Lisäksi tavoitteena on ollut kartoittaa toimeksiantajan yksikön työviihtyvyyttä sekä työmotivaatiota. Toimeksiantajalle laadittiin myös raportti, jossa analysoitiin tarkemmin toimeksiantajan käyttämien Agile-menetelmien tehokkuutta ja toimivuutta heidän työympäristössään, tuotiin esille tutkimuksen pohjalta esiin tulleita mahdollisia ongelma-kohtia sekä esitettiin parannus- ja kehitysehdotuksia.

Opinnäytetyön tutkimuksellinen osuus toteutettiin kvalitatiivisella eli laadullisella tutkimusmenetelmällä valitsemalla tutkimusstrategiaksi tapaustutkimus. Tapaustutkimuksen aineisto kerättiin teemahaastattelun avulla haastatteleamalla kaikki kohdeorganisaation työntekijät. Teemahaastattelussa oli kolme eri teemaa: teema 1: Agile käsitteenä ja työympäristönä, teema 2: työmotivaatio, työssä jaksaminen ja työkuorma ja teema 3: parannus - ja kehitysehdotukset.

Haastattelututkimuksesta kävi ilmi, että kohdeorganisaation yleinen työilmapiiri oli rento ja positiivinen. Työntekijät olivat motivoituneita ja sitoutuneita työntekoonsa ja työnteko myös koettiin mielekkääksi ja tehokkaaksi. Agile-menetelmään siirtymisen koettiin yleisesti ottaen tuoneen selkeyttä sekä työtehtävien hoitamiseen että läpinäkyvyyttä koko kohdeorganisaation toimintaan. Näistä seikoista huolimatta kehittämiskohteita löytyi muun muassa työntekijöiden kompetenssien lisäämiseen, sprintin työtehtävien suunnitteluun sekä testauksen käytäntöihin liittyvissä asioissa. Tästä opinnäytetyöstä saatujen kehitysehdotusten avulla kohdeorganisaation on mahdollista aloittaa toimintaansa kehittäminen vielä nykyistä tehokkaammaksi.

Röyhkiö, Minna

Agile software development methods: a case study of an ICT company

Year	2012	Pages	46
------	------	-------	----

The purpose of this thesis is to examine the most well-known Agile software development methods and research the effectiveness and functionality of the used Agile methods by the principal's organisation. In addition, the objective is to research the work well-being and the work motivation of the organisation. Also a report for the principal organisation was created. The report analyses the effectiveness of the used Agile methods and functionality, highlights the possible challenges and gives suggestions for improvements.

The research-based section of this thesis was realised by using the qualitative research method and choosing case study as a research strategy. The empirical data was collected by using the theme interview method, by interviewing all employees of the target organisation. The interview had three different themes: 1) Agile as a concept and as a working environment 2) work motivation, work well-being and work load and 3) proposals for improvement and development.

The outcome of the research was that the work atmosphere of the target organization was relaxed and positive. Employees were motivated and committed to their work. They also felt that the working was sensible and effective. It was common experience that the transition to the Agile method had brought clarity to managing the work assignments and transparency to the whole organisation. In spite of these issues some subjects were found to be developed further, such as increasing the competences of the employees, the sprint task planning process development and testing practices development. The target organisation can start its development work with the help of the development proposals that have been received from this thesis.

Key words Agile software development methods, scrum, testing

Sisällys

1	Johdanto.....	6
2	Agile-menetelmät.....	7
2.1	Extreme Programming (XP).....	9
2.2	Scrum.....	12
2.3	Agile-menetelmät ja testaaminen	15
3	ICT-alan yritys	17
3.1	Kohdeorganisaatio	17
3.2	Kohdeorganisaation käyttämät Agile-menetelmät	17
4	Tutkimusmenetelmät ja tutkimuksen toteuttaminen	23
4.1	Tutkimusmenetelmän valinta	25
4.2	Teemahaastattelututkimuksen tulokset ja tulosten analysointi	26
4.2.1	Teema 1: Agile käsitteenä ja työympäristönä	26
4.2.2	Teema 2: työmotivaatio, työssä jaksaminen ja työkuorma	30
4.2.3	Teema 3: parannus- ja kehitysehdotukset	33
5	Pohdinta ja johtopäätökset	37
	Lähteet	41
	Kuviot	42
	Liitteet.....	43
	Liite 1: Teemahaastattelulomake suomeksi.....	43
	Liite 2: Teemahaastattelulomake englanniksi	45

1 Johdanto

Tämän opinnäytetyön tavoitteena on ollut perehtyä yleisimpiin Agile-ohjelmistokehitysmenetelmiin Extreme Programming (XP) ja Scrum, Agile-ympäristössä tapahtuvaan testaamisprosessiin ja työtiimien kokoonpanoihin. Lisäksi tavoitteena on ollut tutkia toimeksiantajan käyttämien Agile-menetelmien toimivuutta yksikössään sekä laatia toimeksiantajalle raportti, jossa analysoidaan tarkemmin toimeksiantajan käyttämien Agile-menetelmien toimivuutta työympäristössään, tuodaan esille tutkimuksen pohjalta esiin tulleita mahdollisia ongelmakohtia sekä esitetään parannus- ja kehitysehdotuksia.

Toimeksiantajana opinnäytetyössäni oli globaalin ICT-alan yrityksen IT-organisaatio. Tutkimus tuli ajankohtaiseksi, koska kohdeorganisaation ohjelmistokehitysprojekteissa oli otettu Agile-menetelmät käyttöön noin vuosi sitten, ja yksikön esimiehet halusivat tarkemmin selvittää, miten voitaisiin edelleen tehostaa Agile-menetelmien käyttöä heidän työympäristössään. Tähän liittyi oleellisesti myös organisaatiossa tehdyt muutokset, joissa työntekijöiden työnkuvat ja tiimit oli muodostettu uudelleen juuri Agile-ympäristöön soveltuvaksi. Työt tässä organisaatiossa tehtiin useissa eri projekteissa/projektitiimeissä, joissa oli mukana sekä yrityksen omia työntekijöitä että alihankkijoiden työntekijöitä. Organisaatiolla oli menossa jatkuvasti useita eri projekteja, joissa työskenneltiin Agile-ympäristössä. Tutkimuksen avulla toimeksiantaja halusi samalla myös kartoittaa työntekijöidensä yleistä työviihtyvyyttä ja työmotivaatiota.

Tutkimuksen kohteena oleva IT-organisaatio on osa ICT-alan yrityksen laajempaa IT-yksikköä. Tässä opinnäytetyössä keskitytään pelkästään kohdeorganisaatioon ja ICT-alan yrityksen muu IT-yksikkö on tämän tarkastelun ulkopuolella.

Tutkimuksen keskeisiä kysymyksiä olivat muun muassa: Onko ICT-alan yrityksen IT-organisaation käyttämiä Agile-menetelmiä sovellettu työympäristöönsä tehokkaimmalla mahdollisella tavalla? Olisiko toimintamallia mahdollista vielä parantaa tehokkaammaksi ja toimivammaksi? Onko ICT-alan yrityksen IT-organisaation työtiimien kokoonpano nyt muodostettu niin, että ne soveltuvat parhaiten Agile-ympäristöön vai pitäisikö työtiimien kokoonpanoja vielä räätälöidä tukemaan entistä paremmin Agile-ympäristöä? Miten työntekijät itse kokevat Agile-ympäristössä työskentelemisen? Minkälainen vaikutus sillä on työntekijän ammattitaitoon, työtehokkuuteen, työmotivaatioon?

Agile- eli ketterä ohjelmistokehitysmenetelmä on termi, joka kuvaa useita erilaisia ohjelmistojen kehitykseen käytettäviä menetelmiä. Agile-menetelmiksi katsotaan ne ohjelmistokehitysmenetelmät, joissa toteutuvat seuraavat ominaisuudet: inkrementaalisuus

(menetelmä on vähittäin kasvava ja nopeasyklinen ja julkaistavat tuotteet ovat pieniä kokonaisuuksia), tiivis yhteistyö ja kommunikointi asiakkaan ja ohjelmistokehittäjän välillä, vaivattomuus (menetelmä on helppo oppia ja sitä on helppo muunnella ja myös dokumentointi on kattavaa ja järjestelmällistä) sekä joustavuus, jolloin on mahdollisuus myös tehdä viime hetken muutoksia kehitettävään tuotteeseen (Abrahamsson, Salo, Ronkainen & Warsta 2007, 17).

2 Agile-menetelmät

Monet tekijät ovat johtaneet siihen, että nykyään yritysten on pystyttävä reagoimaan nopeasti muutoksiin. Täten myös ohjelmistokehitysprosessien on pysyttävä menossa mukana. Jos ohjelmistonkehitysprosessi on liian raskas ja pitkäkestoinen, voi sovellus olla jo vanhentunut, kun se julkaistaan. Useissa yrityksissä onkin jo käytössä Agile-ohjelmistokehitysmenetelmiä, jotka sopivat hyvin tämä hetken tilanteeseen. Agile-menetelmät mahdollistavat nopean kehitysprosessin sisältäen nopeudestaan huolimatta kaikki ohjelmistokehityksen suunnitteluun ja toteutukseen kuuluvat vaiheet kuten suunnittelun, määrittelyn, toteutuksen ja testauksen.

Agile-ohjelmistokehitysmenetelmälle oleellista on sen nopeus, joustavuus ja läpinäkyvyys. Kohdeorganisaation johdon (Kohdeorganisaation johdon haastattelu 2, 2012) mukaan juuri kyseisten ominaisuuksien avulla saadaan tuotettua asiakkaalle juuri se tuotos/tuote, joka on tärkein eli tuottaa asiakkaalle suurimman arvon. Tämä myös laskee riskiä nopeasti, koska saadaan tuotettua asiakkaan haluamat tärkeimmät asiat nopeasti. Tällöin voidaan heti laskea myös riskiä tyytymättömästä asiakkaasta.

Ohjelmistokehitystyössä kehitystiimi keskittyy kehitystyössään aluksi vain välttämättömiin toiminnallisuuksiin; näin tiimi toteuttaa ja toimittaa tuotokset nopeasti tilaajalle / asiakkaalle kerätäkseen palautetta tuotoksestaan. Saamansa palautteen pohjalta kehitystiimi sitten tekee tuotokseensa tarvittavia muutoksia ja lisäyksiä. (Abrahamsson ym. 2007, 17.) Eli käytännössä Agile -ympäristössä kehitystyö etenee askel kerrallaan niin, että sovellustuotokseen lisätään vähitellen lisää uusia toiminnallisuuksia ja ominaisuuksia aina ihan sovelluksen viimeiseen vaiheeseen saakka. Kaiken aikaa asiakas ja kehitystiimi toimivat läheisessä yhteistyössä ja kommunikoivat keskenään aktiivisesti. Abrahamsson ym. (2007, 17) toteavat, että oleellista on myös kehitysprojektin selkeä ja kattava dokumentointi. Agile-ohjelmistokehitysmenetelmissä iteratiivisuus on kaiken a ja o – prosessissa ovat limittäin kaikki ohjelmistokehityksen suunnitteluun ja toteutukseen kuuluvat vaiheet: suunnittelu, määrittely, toteutus ja testaus.

Agile manifeston (2001) mukaan Agile-ohjelmistokehityksen 12 keskeistä periaatetta ovat:

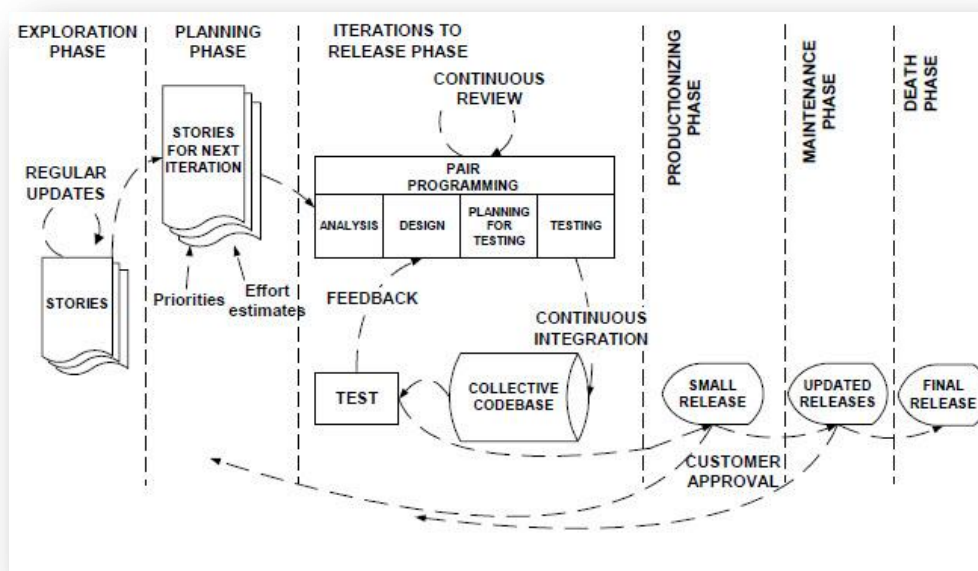
- ”Tärkein tavoitteemme on tyydyttää asiakas toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti
- Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyyn edistämiseksi
- Toimitamme versioita toimivasta ohjelmistosta säännöllisesti, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä
- Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan
- Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä
- Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu
- Toimiva ohjelmisto on edistymisen ensisijainen mittari
- Ketterät menetelmät kannustavat kestävään toimintatapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen
- Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä
- Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista
- Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä
- Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.”

Käytössä olevia Agile-ohjelmistokehitysmenetelmiä on useita erilaisia. Tähän opinnäytetyöhön valittiin niistä kaksi yleisesti käytettyä menetelmää Extreme Programming (XP) ja Scrum. XP:n ja Scrumin eroavaisuudet käyvät tarkemmin ilmi seuraavissa luvuissa, mutta yhteenvetona

todettakoon, että konkreettiset erot XP:n ja Scrumin välillä ovat muun muassa rooleissa: XP-menetelmässä on enemmän ja erilaisia nimettyjä rooleja kuin Scrum-menetelmässä. Lisäksi XP- ja Scrum-menetelmien kehitysprosessit eri vaiheineen eroavat aika lailla toisistaan.

2.1 Extreme Programming (XP)

Yksi käytetyimmistä Agile -ohjelmistokehitysmenetelmistä on Extreme Programming (XP). XP on Kent Beckin 1990-luvun lopulla kehittämä ohjelmistokehitysmenetelmä. Kuten kuviosta 1 ilmenee, Extreme Programming (XP)-menetelmän elinkaareissa on viisi vaihetta: tutkimusvaihe eli exploration phase, suunnitteluvaihe eli planning phase, iteraatiosta julkaisuun-vaihe eli iterations to release phase, tuotteistamisvaihe eli productionizing phase sekä ylläpitovaihe eli maintenance & death phase (Abrahamsson ym. 2007, 19).



Kuvio 1: Extreme Programming (XP)-menetelmän elinkaaren havainnointikuva (Abrahamsson ym. 2007, 19).

Tutkimusvaihe kestää muutamasta viikosta muutamaa kuukauteen. Vaiheen aluksi asiakas ilmoittaa halukkuutensa olla mukana ensimmäisessä julkaisussa tuotteeseen lisättävän ominaisuuden kuvauksen eli story cardin avulla. (Abrahamsson ym. 2007, 20.) Tämän vaiheen aikana projektitiimi päättää, mitä työkaluja, käytänteitä ja teknologiaa tulevat käyttämään kyseisessä ohjelmistokehitysohjelmassa.

Suunnitteluvaiheen aikana sovitaan, mitkä ominaisuudet sisältyvät ensimmäiseen julkaisuun (release). Tämän vaiheen aikana myös eri tuotteisiin lisättävien ominaisuuksien ominaisuusvaatimukset eli storyt asetetaan tärkeysjärjestykseen. (Abrahamsson ym. 2007,

20.) Kun nämä edellä mainitut seikat on sovittu, ohjelmoija ja projektitiimi sopivat tarkemman aikataulun etenemiselle.

Iterations to release phase, vapaasti suomennettuna iteraatiosta julkaisuun -vaihe, sisältää useita iteraatioita, joiden aikana tuotteisiin lisätään uusia ominaisuuksia, jotta päästään ensimmäiseen julkaisuun. Abrahamssonin ym. (2007, 20) mukaan tämä tapahtuu käytännössä niin, että edellisessä vaiheessa asetettu aikataulu pilkotaan pienempiin osiin eli iteraatioihin. Nämä iteraatiot kestävät yhdestä neljään viikkoa. Ensimmäinen iteraatio luo koko kehitettävän tuotteen arkkitehtuurin. Tässä kehitysvaiheessa asiakas päättää, mitkä ominaisuudet valittavana olevista vaihtoehtoista eri iteraatioihin otetaan. Joka iteraatiossa suoritetaan myös toiminnalliset testaukset. Viimeisen iteraation jälkeen tuote on tuotantovalmis.

Tuotteistamisvaihe eli productionizing phase pitää sisällään lisää testaamista, jotta voidaan varmistaa tuotteen toimivuus ennen sen julkaisua asiakkaalle. Tässä vaiheessa voidaan vielä huomata uusia muutoksia, jotka pitää lisätä kehitettävään tuotteeseen. Tässä tapauksessa iteraatiot pitää lyhentää yhden viikon mittaisiksi. Mahdolliset ideat ja kehitysehdotukset voidaan myös dokumentoida ja siirtää toteutettavaksi myöhäisempään vaiheeseen. (Abrahamsson ym. 2007, 20.)

Ylläpitovaiheen eli maintenance phase aikana tuote on jo otettu asiakkaan käyttöön. Kehitetty tuote täytyy pitää projektitiimin toimesta käynnissä samalla, kun tuotteeseen kehitetään uusia ominaisuuksia. Tähän vaiheeseen sisältyy myös asiakastuki. Ylläpitovaiheessa projektitiimin kokonpano usein muuttuu, jotta tiimissä on tähän vaiheeseen tarvittavat kompetenssit. Loppuvaihe eli death phase eli tarkoittaa sitä, ettei tuotteeseen ole käytännössä enää lisättäviä ominaisuuksia ja asiakas on tyytyväinen saamaansa tuotteeseen kaikin puolin. Myös tuotteen kehitykseen liittyvä dokumentointi pitää olla kirjoitettu yksityiskohtaisesti. Toisaalta loppuvaiheeseen tuotteen kehityksessä voidaan päätyä myös silloin, kun tuote ei ole pyydetyn mukainen tai sen jatkokehittäminen on tullut liian kalliiksi. (Abrahamsson ym. 2007, 20.)

Smithin ja Sidkyn (2009, 35) mukaan XP:n vahvuuksia ovat muun muassa asiakasorientoituneisuus, toistuvan testauksen kautta saatava laatu, projektin statuksen hyvä näkyvyys sekä hyvä tuki häilyville vaatimuksille. XP:n heikkouksia puolestaan ovat seuraavat ominaisuudet, kuten että tiimi pitää sisältää vastuuntuntoisia työntekijöitä ja heidänlaistensa löytäminen ei ole itsestäänselvyys, XP on riippuvainen testaamisesta, XP ei sovellu kovin hyvin isoille projekteille sekä tiimin jäsenten pitää sijaita samassa lokaatiossa.

XP - prosessissa on eri rooleja, joista jokaisella on oma tarkoituksensa. Roolit ovat Beckin kehittämän menetelmän mukaan (Abrahamsson ym. 2007, 21) ohjelmoija eli programmer, asiakas eli customer, testaaja eli tester, mittaaaja eli tracker, valmentaja eli coach, konsultti eli consultant, sekä johtaja eli Manager (Big Boss).

Ohjelmoija on keskeinen henkilö – hän kirjoittaa mahdollisimman selkeän ohjelmakoodin sekä sen testauksen ollen samalla aktiivisessa yhteistyössä muiden ohjelmoijien sekä projektitiimin jäsenten kanssa. Asiakas antaa käyttötapaukset eli storyt eli projektitiimille, suunnittelee toiminnallisia testejä, asettaa käyttötapausten eri vaatimukset sekä priorisoinnit ja päättää myös siitä, milloin hänen mielestään käyttötapaus on toimitettu niin, että sen voi hyväksyä (Abrahamsson ym. 2007, 21.)

Testaajat auttavat asiakasta toiminnallisten testien suunnittelemisessa. He myös suorittavat toiminnallisia testejä säännöllisesti sekä kommunikoivat testien tuloksista muulle projektitiimille. Testaajien vastuulla on myös testaamiseen liittyvien työkalujen ylläpito. (Abrahamsson ym. 2007, 21.)

Mittaaaja antaa palautetta arvioimalla sekä projektitiimin edistymistä asetettuihin tavoitteisiin ja aikatauluihin nähden että projektitiimin konkreettista edistymistä. Hän antaa myös parannusehdotuksia tulevien tavoitteiden ja aikataulujen suunnitteluun. (Abrahamsson ym. 2007, 21.)

Valmentaja tuntee hyvin XP-prosessin. Hän opastaa projektitiimiä XP:n toimintavoista sekä ohjaa miten XP:tä tulee soveltaa juuri tähän kyseiseen projektiin. (Abrahamsson ym. 2007, 22.)

Konsultti on projektin niin sanottu ulkopuolinen ”vieraileva” jäsen. Hänellä on hyvin vahva tekninen tietämys aiheesta. Projektitiimi saa konsultilta tarvittaessa apuja ongelmatilanteisiinsa. (Abrahamsson ym. 2007, 22.)

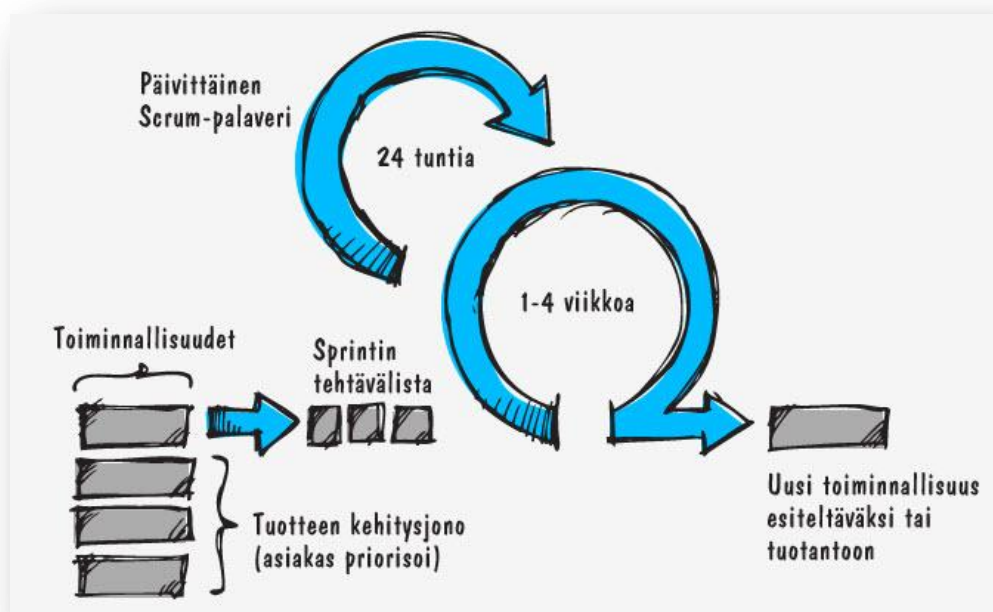
Johtaja tekee konkreettiset päätökset, joista hänellä on myös vastuu. Johtaja on aktiivisessa kommunikoinnissa projektitiimin kanssa ja voi joskus joutua tekemään radikaalejakin ratkaisuja saadakseen eliminoitua mahdolliset projektia haittaavat vaikeudet. (Abrahamsson ym. 2007, 22.)

2.2 Scrum

Scrum -ohjelmistokehitysmenetelmä on periaatteessa viitekehys, jonka avulla voidaan suunnitella ja kehittää monimutkaisiakin tuotteita. Scrum-viitekehityksen sisällä voi käyttää hyvinkin erilaisia prosesseja ja tekniikoita; näin on mahdollista räätälöidä eri kehitysympäristöihin parhaiten soveltuvat prosessit ja käytänteet. Scrum on kevyt ja helposti ymmärrettävä, mutta toisaalta sen kokonaishallinnointi voi olla haasteellistakin. (Schwaber & Sutherland 2011, 4.) Scrum-ohjelmistokehitysmenetelmä sopii parhaiten sellaiseen kehitysympäristöön, jossa tilanteet ovat vaikeasti ennustettavissa.

Scwaberin ja Sutherlandin (2011, 3) mukaan Scrum perustuu empiiriseen prosessinhallinta-teoriaan. Tämä tarkoittaa sitä, että käytettävä informaatio pohjautuu sekä kokemukseen että päätösten tekemiseen jo tiedossa olevien faktojen pohjalta. Scrumille ominaista onkin juuri sekä ennustettavuuden optimoiminen että mahdollisten riskien kontrolloiminen. Näiden saavuttamiseen Scrumissa käytetään lähestymistapaa, jota kutsutaan iteratiivis-inkrementaaliseksi eli suomennettuna toistavaksi ja lisääväksi lähestymistavaksi. Läpinäkyvyys, tarkastelu ja sopeuttaminen ovat avaintekijöitä tässä empiirisessä teoriassa.

Alla olevasta kuviosta 2 käy hyvin ilmi, miten Scrum-prosessi toimii. Keskeinen käsite on sprint. Sprint on useimmiten 1-4 viikon mittainen aikajakso, minkä aikana projektitiimi suunnittelee, analysoi, testaa sekä dokumentoi valittuihin ominaisuuksiin liittyviä oleellisia tietoja jatkuvasti (Smith & Sidky 2009, 32). Lisäksi sprintin päätteeksi esitellään valmis tuoteversio tai sprintin aikana toteutetut toiminnallisuudet.



Kuvio 2: Scrumin prosessikaavio (Reaktor-sivusto).

Hyvin usein toimitaan niin, että projektitiimi pitää päivittäisen tilannekokouksen eli Daily Scrumin katselmoidakseen edellisen kokouksen jälkeen toteutuneet seikat, sen päivän työlliställä olevat seikat sekä mahdolliset esiin tulleet viat. Kun sprint on saatettu loppuun, tuotokset esitellään asiakkaalle ja projektitiimi ja asiakkaat päättävät yhdessä, tehdäänkö tuotteeseen/sovellukseen muutoksia vai pidetäänkö tuotetta jo hyväksyttynä (Smith & Sidky 2009, 33).

Smithin ja Smidkyn (2009, 33) mukaan Scrumin vahvuuksia ovat muun muassa priorisoitu toimitus eli prioritized delivery: ominaisuudet toimitetaan jaksoissa ja tämä on suoraan sidoksissa business arvoon, läpinäkyvyys ja avoimuus eli status transparency, tiimin vastuullisuus eli team accountability sekä tuotosten jatkuva toimitus eli continuous delivery. Scrumin heikkouksia puolestaan ovat: Scrum-projektitiimin pitää itse identifioida käytettävät toimintatavat ja käytänteet, jotka parhaiten soveltuvat juuri heidän ympäristöönsä, toimiakseen menestyksekkäästi, Scrum-tiimi tarvitsee osaavan Scrum Masterin sekä se, että Scrum haluaa asiantuntijoita, joilla on oman erityisosaamisensa lisäksi myös laaja-alainen perusosaaminen kaikkia Scrum-tiimissä suoritettavia työtehtäviä ajatellen. Asiantuntijatiimissä olleelle voi tuntua haasteelliselta ollakin tiimissä, jossa kuka tahansa voi tehdä periaatteessa mitä tehtävää tahansa.

Scrum-kokonaisuuteen kuuluvat Scrum-tiimit eri rooleineen sekä tapahtumat, tuotokset ja säännöt. Schwaber & Sutherland (2011, 4) toteavat, että Scrum-tiimiin kuuluvat yleensä tuoteomistaja eli Product Owner, Scrum master sekä kehitystiimi eli Development Team. Scrum-tiimit toimivat itsenäisesti eli käytännössä kukaan ulkopuolinen ei ohjaa tiimin toimintaa, vaan Scrum-tiimi on itseohjautuva ja päättää itse miten työnsä suorittavat. Tällainen toimintapa jättää tilaa myös luovuudelle ja joustavuudelle. Tiimin kehitystoiminta tapahtuu niin, että tuotetta kehitetään toistavasti ja lisäävästi palautteet huomioiden.

Schwaberin ja Sutherlandin mukaan (2011, 5, 6) Scrum-tiimien kokojen suositellaan olevan 3-9 henkilöä. Alle kolmen henkilön tiimi on niin pieni, ettei tuotetta välttämättä pystytä kehittämään riittävälle tasolle. Yli yhdeksän henkilön tiimi on taas puolestaan liian iso ja kehitystoiminnasta saattaa tulla sekava ja vaikeasti koordinoitava kokonaisuus.

Tuoteomistaja eli Product Owner tarkoittaa yhtä henkilöä, joka on vastuussa kehitettävän tuotteen arvosta sekä tuotteen kehitysjonon hallinnoinnista; tuoteomistajan tekemiä päätöksiä pitäisi koko organisaation noudattaa. Tuotteen kehitysjonon hallinnoimiseen kuuluvat muun muassa eri vaiheiden selkeä kirjallinen dokumentointi, pitäminen kehitysjonon vaiheet oikeassa järjestyksessä jotta päästään haluttuun tavoitteeseen, kehitysjonon läpinäkyvyyden ja ymmärtämisen varmistaminen sekä varmistaminen, että kehitystiimi tietää mitä pitää tehdä. (Schwaber & Sutherland 2011, 5.)

Scrum master on vastuussa siitä, että kaikki Scrum-tiimiin kuuluvat tietävät mitä Scrum-ympäristössä työskentely konkreettisesti tarkoittaa ja että kaikki varmasti käyttävät Scrumia. Schwaberin & Sutherlandin (2011, 6) mukaan Scrum masterin velvollisuuksiin kuuluvat erilaiset palvelut tuoteomistajalle, kehitystiimille sekä organisaatiolle. Scrum master palvelee tuoteomistajaa muun muassa niin, että ehdottaa erilaisia keinoja tuotteen kehitysjonon hallinnoimiseen, on jatkuvassa yhteydessä kehitystiimiin kehitettävään tuotteeseen liittyvistä yksityiskohdista sekä kannustaa ja perehdyttää scrum-tiimiläisiä kehitysjonon tehokkaista ja toimivista työskentelytavoista. Kehitystiimiä Scrum master palvelee puolestaan muun muassa niin, että yrittää saada kehitystiimistä mahdollisimman itseohjautuvan ja kompetenssien suhteen yhtenäisen tiimin (kaikki osaavat kaikkea) sekä saamaan heitä kehittämään mahdollisimman laadukkaita tuotteita. Scrum masterin palvelut organisaatiolle pitävät sisällään muun muassa organisaation, sidosryhmien ja työntekijöiden perehdyttämisen Scrum-menetelmään sekä Scrumin käyttöönottovaiheessa että sen jälkeen, sekä scrum-tiimin ja itse Scrum-prosessin tuottavuuden ja tehokkuuden kasvattaminen tarvittaessa yhteistyössä muiden Scrum mastereiden kanssa.

Kehitystiimiin eli Development Teamiin kuuluvat ovat alansa ammattilaisia. Heidän vastuullaan on kehitysjonon kehittäminen käyttökelpoiseksi tuoteversioksi sprintti sprintiltä.

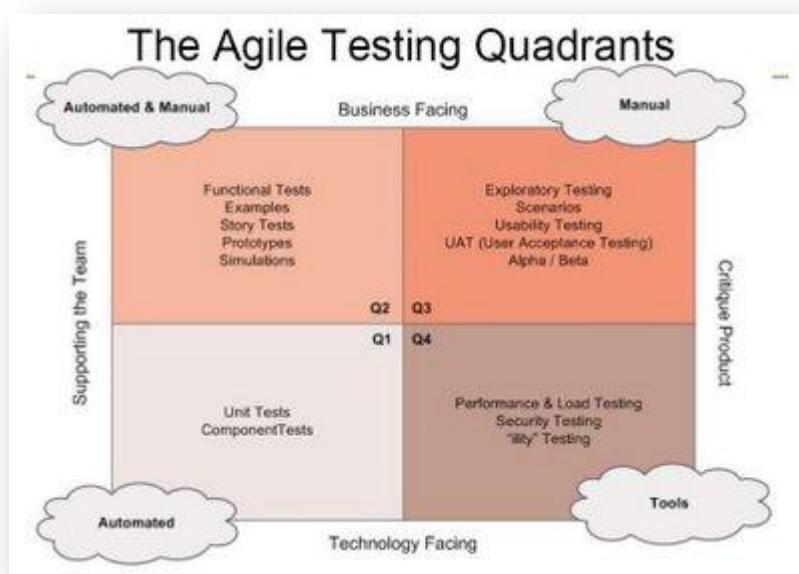
Kehitystiimissä tulisi olla vähintään 3 henkilöä, jotta tiimillä on tarpeeksi kattava osaaminen kaikilla tarvittavilla osa-alueilla. Kehitystiimi päättää itse, miten he tuoteversiotyönsä tekevät eli kehitystiimi on itseohjautuva. (Schwaber & Sutherland 2011, 5.) Koko kehitystiimi on yhdessä vastuussa kehitystyöstä, vaikka työnkuvat olisivatkin erilaiset tiimin sisällä.

Scrum -ympäristössä keskeisenä seikkana on sprintti. Sprintti on 1-4 viikon mittainen aikajakso, jonka päätteeksi esitellään valmis tuoteversio / sprintin aikana toteutetut toiminnallisuudet. Sprintti sisältää suunnittelupalaverin, päiväpalaverin, kehitystyön, sprinttikatselmuksset sekä sprintin retrospektiivin eli työtapojen tarkastelukokouksen. (Schwaber & Sutherland 2011, 7).

Muita Scrumin keskeisiä käsitteitä ovat tuotteen kehitysajon eli Product backlog, sprintin tehtävälista eli Sprint backlog sekä valmiin määritelmä eli Definition of Done (DoD). Reaktor-sivuston (2012) mukaan tuotteen kehitysajon eli Product backlog tarkoittaa käytännössä priorisoitua listaa siitä, mitä kaikkea ollaan kehittämässä. Lista päivittyy ja tarkentuu jatkuvasti projektin edetessä ja on apuna projektin seuraamisessa ja hallinnoimisessa. Sprintin tehtävälista eli Sprint backlog kuvaa yksityiskohtaisesti kaikki ne seikat, joita sprintin aikana toteutetaan. Valmiin määritelmä eli Definition of Done (DoD) on säännöstö, jossa on määritelty ne kriteerit, joiden täytyttyä sprintissä tehdyt toiminnallisuudet on täytetty.

2.3 Agile-menetelmät ja testaaminen

Agile -kehitysympäristössä testaaminen on hyvin tärkeässä roolissa. Tuotteen kehittäminen tapahtuu iteraatioittain vaihe vaiheelta ja joka iteraatiossa on mukana luonnollisena osana myös testaaminen, kun iteraatioon lisättyä ominaisuutta testataan (kuvio 3).



Kuvio 3: The Agile Testing Quadrants (Gregory 2009, 10).

Agile-kehitysympäristössä jo hyvin aikaisessa vaiheessa suunnitellaan, mitä konkreettisesti testataan ja millä menetelmällä. Testauksessa käytetään samoja testaustasoja kuin perinteisessä vesiputousmallinkin testaamisessa, kuten yksikkötestaus, integrointitestausta, järjestelmätestaus sekä hyväksymistestausta (Vuori 2010, 9, 60).

Testaaminen ei ole vain yhden henkilön, vaan koko tiimin vastuulla. Daviesin & Sedleyn (2009, 123) mukaan koko kehitysprosessin ajan tuotteen kehittämiseen osallistuvilla henkilöillä pitää olla keskenään saumaton ja aktiivinen yhteistyö, jotta kaikki ymmärtävät konkreettisesti minkälaista tuotetta ja minkälaisilla toiminnallisuuksilla sitä ollaan kehittämässä. Tämä on edellytyksenä sille, että voidaan yksityiskohtaisesti suunnitella kehitettävälle tuotteelle tarvittavat testaukset ja että ne myös saadaan tehtyä ajallaan.

Nykyään on hyvin yleistä, että vaikka yrityksessä tai organisaatiossa käytetäänkin ohjelmistonkehityksessä Agile-menetelmää, niin tuotteen testaaminen tapahtuu usein vielä manuaalisesti. Davies & Sedley (2009, 137, 138) toteavat, että suositeltavaa olisi uudistaa koko ohjelmistokehitysprosessia niin, että voitaisiin mahdollisuuksien mukaan käyttää testausvetoista kehittämistapaa eli Test-Driven Development-menetelmää (TDD). Käytännössä TDD tarkoittaa sitä, että kehitysvaiheen aluksi kehittäjä kirjoittaa testiskriptin valmiiksi ohjelmistokoodilleen, mitä on kehittämässä ja tarkistaa, että kyseinen testiskripti todellakin epäonnistuu. Tämän jälkeen kehittäjä kirjoittaa ohjelmistokoodin, joka sitten läpäisee kirjoitetun testiskriptin. Näin toistetaan ja tehdään aina uusia ja uusia osia kehitettävään

ohjelmistoon. Jokaisen läpäistyn testauksen jälkeen eliminoidaan päällekkäisyydet ja yhdistetään koodi-osiot vähitellen yhtenäiseksi kokonaisuudeksi, kunnes kehitystyö on valmis.

Yrityksen tai organisaation siirtyminen käyttämään TDD-menetelmää on pitkä prosessi ja siirtymisen pitää tapahtua vaiheittain pikku hiljaa. Davies & Sedley (2009, 137, 152) toteavatkin, että oleellista on, että koko kyseinen organisaatio on hyväksynyt ja sisäistänyt TDD-menetelmän käytön ja on samaa mieltä testausstrategiasta. Koko ohjelmistokehitykseen kuuluva tiimi pitää kunnolla perehdyttää ja opettaa tekemään automaattiseen testaukseen käytettäviä testiskriptejä, jotta TDD-menetelmään pystytään käyttämään tarkoituksenmukaisesti.

3 ICT-alan yritys

Tutkimuksen kohteena oleva IT-organisaatio kuuluu ICT-alan yritykseen, joka toimii maailmanlaajuisesti lähes kaikissa maanosissa työllistäen useita kymmeniä tuhansia ihmisiä. Yrityksen toimialana on mobiilituotteiden valmistus sekä monipuolisten mobiilisovellusten kehitys.

3.1 Kohdeorganisaatio

Tutkimuksen kohteena oleva IT-organisaatio on osa ICT-alan yrityksen laajempaa IT-yksikköä. Tässä opinnäytetyössä keskitytään pelkästään tutkimuksen kohteena olevaan IT-organisaatioon ja ICT-alan yrityksen muu IT-yksikkö on tämän tarkastelun ulkopuolella.

Tutkimuksen kohteena oleva IT-organisaatio on mukana useissa eri ohjelmistokehitysprojekteissa sekä prosessien kehittämisprojekteissa. Suuri osa näistä projekteista on ICT-alan yrityksen eri tuotantolaitoksiin liittyviä (Kohdeorganisaation johdon haastattelu 1, 2012).

3.2 Kohdeorganisaation käyttämät Agile-menetelmät

Tutkimuksen kohteena olevan IT-organisaation toiminta pohjautuu koko IT-yksikölle laadittuun strategiaan. Kohdeorganisaatiossa käytettävä Agile-menetelmä on tälle yksikölle räätälöity Scrum-menetelmä. Kohdeorganisaation räätälöityyn Agile-ympäristöön liittyy myös oma erityinen termistönsä. Oleelliset käsitteet ovat työpyyntö eli need, iso vaatimus eli epic, yhteen julkaisuun liittyvä vaatimus eli feature, scrum tiimin työpyyntö eli user story sekä konkreettinen työtehtävä eli task (Kohdeorganisaation johdon haastattelu 1, 2012).

Need tarkoittaa konkreettista (kirjallista) työpyyntöä, joka voi olla mikä tahansa tuotteeseen tai prosessiin liittyvä muutospyyntö. Tämä voi käynnistää joko hyvin pienen tai hyvin suuren vaatimuksen eli requirementin. Epic puolestaan tarkoittaa isoa business requirementtia eli vaatimusta, joka voidaan implementoida moniin tuotteisiin tai moniin julkaisuihin releaseihin. Jokaiselle epicille määritellään omistaja eli Epic Owner, joka pitää ylätasolla huolta epicistä. Tämä rooli vastaa lähinnä projektipäällikön roolia ja kyseiseen rooliin valitaan esimerkiksi IT-organisaatiosta osaamisensa, työtehtäviensä tai vastuunsa puolesta soveltuva henkilö. Feature tarkoittaa sellaista vaatimusta, mikä sopii yhteen releaseen eli julkaisuun. Feature on priorisoitu Product backlogiin eli tuotekehitysjonoon ja on osa tuotejulkaisun suunnitelmaa eli roadmapia. User story tarkoittaa Scrum-tiimille tehtyä vaatimusta/työpyyntöä, mikä implementoidaan yhden sprintin aikana. Task tarkoittaa kaikkia niitä konkreettisia toimia/työtehtäviä, joita Scrum-tiimi suorittaa että user story saadaan implementoitua. (Kohdeorganisaation materiaali 2012.)

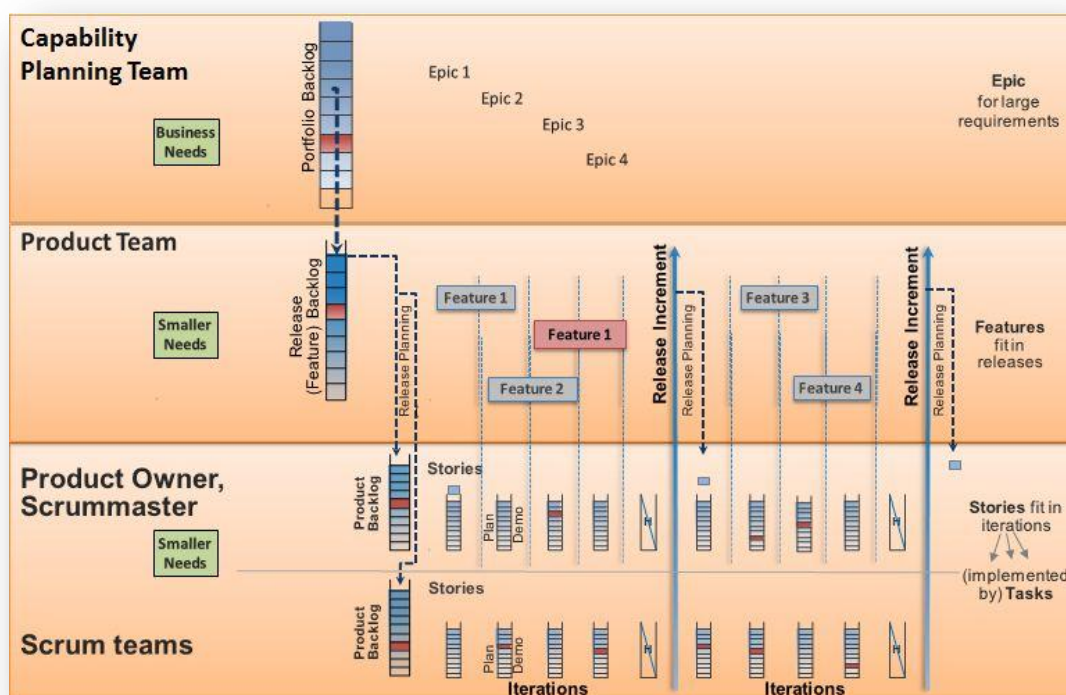
Kohdeorganisaatio muodostuu kahdesta eri kokonaisuudesta, joista käytetään tässä opinnäytetyössä nimityksiä A-osa ja B-osa. A-osan kokoonpanoon kuuluu tuotetiimi eli Product Team sekä 2 Scrum-tiimiä. B-osa muodostuu yhdestä tuotetiimistä sekä yhdestä Scrum-tiimistä. Sekä A- että B-osien tuotetiimeihin kuuluvat linjaesimies (Line Manager), tuoteomistaja (Product Owner), pääkehittäjä (Lead Developer), muutosjohtaja (Transition Manager) sekä prosessikehittäjä (Process Developer) -rooleissa olevat henkilöt. A-osan toiseen Scrum-tiimiin kuuluvat Scrum master sekä 8 asiantuntijaa eli Specialistia ja A-osan toiseen Scrum-tiimiin puolestaan kuuluvat Scrum master ja 4 asiantuntijaa. B-osan Scrum-tiimiin kuuluu puolestaan Scrum master sekä 9 asiantuntijaa. (Kohdeorganisaation materiaali 2012; kohdeorganisaation johdon haastattelu 1, 2012.)

Tuotetiimi eli Product Team saa toimeksiantonsa eli featuren suunnittelutiimiltä eli Capability Planning Teamilta, joka vastaanottaa ison business vaatimuksen eli epicin. Capability Planning Teamiin kuuluu valmiussuunnittelija (Capability Planner), prosesseista vastaavia henkilöitä sekä IT-arkkitehti. (Kohdeorganisaation materiaali 2012; kohdeorganisaation johdon haastattelu 1, 2012.) Tämä tiimi vastaa epicin priorisointijärjestyksestä, omistajuudesta sekä julkaisun eli releasen hallinnasta eli käytännöstä kaikesta siitä mitä, miten ja koska jotakin tehdään kyseisen epicin suhteen.

Tuotetiimi päättää puolestaan yhteen julkaisuun liittyvän vaatimuksen eli featuren osalta mitä, miten, koska ja missä vaiheessa jotakin tehdään. Tuoteomistaja eli Product Owner päättää siitä, mitä featurelle tehdään ja koska. Lead Developer taas päättää, miten se toteutetaan. Transition Manager puolestaan jakaa valmiin tuotteen eteenpäin käyttöönotto- eli deployment-vaiheessa. Prosessikehittäjä (Process Developer) taas päättää mitä prosesseja kehitetään. Product Team käy läpi saamansa featuret ja valmistelee ne Scrum-tiimeille

tuotteen kehitysjonoon eli backlogiin. (Kohdeorganisaation materiaali 2012; kohdeorganisaation johdon haastattelu 1, 2012.) Yhdestä featuresta voi tulla monta työpyyntöä eli storya, joita sitten kehitetään sprintissä: 1 story - 1 sprint, 2 story - 2 sprint ja niin edelleen.

Scrum-tiimi saa työpyyntönsä eli storyt Product Teamilta ja pilkkoo saamansa storyt konkreettisiksi toimiksi/työtehtäviksi eli taskeiksi, joita Scrum-tiimit sitten suorittavat sprinteissä. Scrum-tiimien Specialistit toimivat jokainen eri työtehtävissä, joita ovat koodaaminen (coding), testaaminen (testing), dokumentointi (documenting), koulutus (training), toimeenpano/käyttöönotto (deployment), suunnittelu (planning) sekä tuki (support). (Kohdeorganisaation materiaali 2012; kohdeorganisaation johdon haastattelu 1, 2012.) Specialistit toimivat kukin aina samassa roolissa/työtehtävässä projekteista riippumatta. Kuvioista 4 ilmenee kohdeorganisaation Agile-menetelmän prosessinkuvaus.



Kuvio 4: Kohdeorganisaation käyttämän Agile -menetelmän prosessinkuvaus (Kohdeorganisaation materiaali 2012).

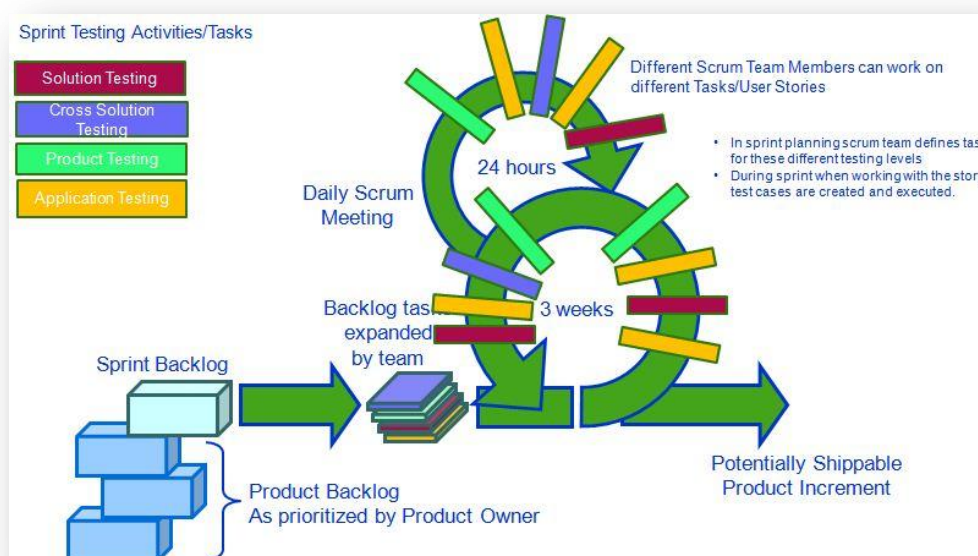
Kohdeorganisaation sprintit kestävät kolme viikkoa. Julkaisukalenteriin eli release-kalenteriin merkataan yksityiskohtaisesti sprinttien tapahtumat (kuvio 5). Tämä auttaa aikataulujen suunnittelussa ja kokonaisuuden hallinnassa. (Kohdeorganisaation materiaali 2012; kohdeorganisaation johdon haastattelu 1, 2012.)

Haasteena sprinteissä on työkuorman jakaantuminen - miten saada Scrum-tiimin työtehtävät eli taskit suunniteltua siten, että töitä olisi tasaisesti sprintin jokaiselle viikolle. Esimerkkinä tästä testausta tekevät Scrum-tiimiläiset – heidän työtehtävänsä eivät saisi kuormittua epätasapainoisesti sprintin jälkimmäisille viikoille, vaan töitä pitäisi olla jo sprintin ensimmäisestä viikosta lähtien. (Kohdeorganisaation johdon haastattelu 1, 2012.) Tätä haastetta yritetään kohdeorganisaatiossa saada ratkaistua koko ajan. Myös tämän opinnäytetyön odotetaan tuovan kyseiseen ongelmaan ratkaisukeinoja.

	Tammikuu	Helmikuu	Maaliskuu	Huhtikuu	Toukokuu	Kesäkuu
1				1		
2	Sprint planning			2		
3				3	1	
4		1 Halfway review		4	2	
5		2	1 Rel. X planning	5	3	
6		3	2	6	4	1
7		4	3	7	5	2
8		5	4	8	6	3
9		6	5 Sprint planning	9	7 Sprint planning	4
10		7	6 Rel. Y Demo	10	8	5
11		8	7	11	9	6
12		9	8	12	10	7
13		10	9	13	11	8
14		11	10	14	12	9
15		12	11	15	13	10
16		13 Sprint planning	12	16 Sprint planning	14	11
17		14	13	17	15	12
18		15	14	18 Halfway review	16 Rel. Y pre plan	13
19		16	15	19	17	14
20		17	16	20	18	15
21		18	17	21	19	16
22		19	18	22	20	17
23	Sprint planning	20	19	23	21	18
24		21	20	24	22	19
25		22	21	25	23	20
26		23 Rel. X pre plan	22	26	24 Rel. Y planning	21
27		24	23	27	25	22
28		25	24	28	26	23
29		26	25	29	27	24
30		27	26 Sprint planning		28 Sprint planning	25 Sprint planning
31		28	27		29 Rel. X demo	26
		29	28			27 Halfway review
			29			28
						29

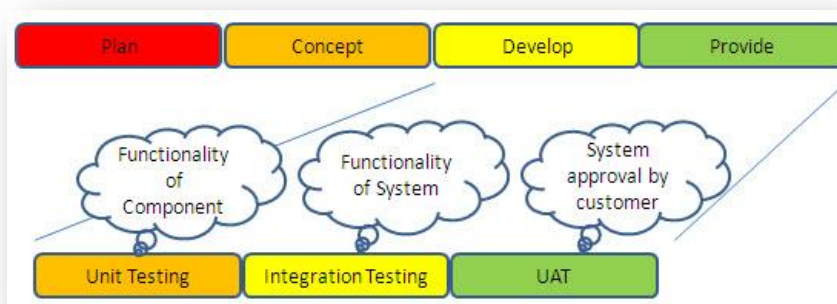
Kuvio 5: Esimerkki kohdeorganisaation käyttämästä julkaisukalenterista (Kohdeorganisaation materiaali 2012).

Testaaminen on yksi oleellinen osa kohdeorganisaation työkenttää ja pohjautuu koko IT-yksikölle laadittuun testausstrategiaan. Testausstrategiassa on laadittu tarkka ohjeistus siitä, minkälaisia testausmenetelmiä organisaatiossa käytetään missäkin kehitysvaiheessa. Kuvio 6 havainnollistaa testausten jakautumista sprintin eri vaiheisiin.



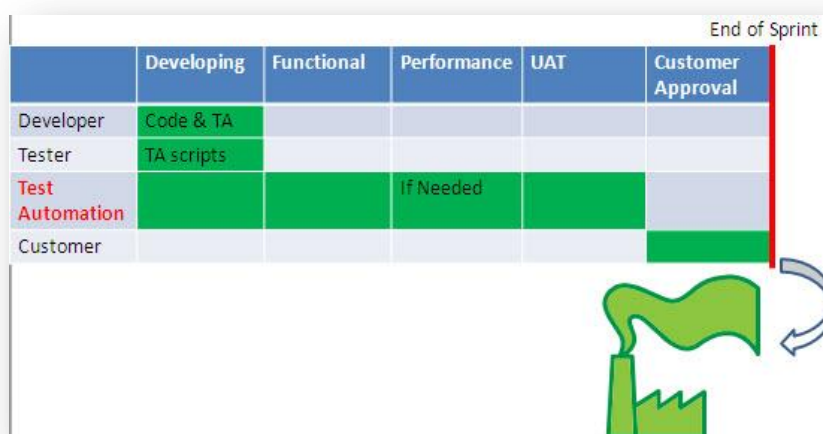
Kuvio 6: Kohdeorganisaation testausprosessin kuvaus sprintissä (Kohdeorganisaation materiaali 2012).

Ennen Agile-kehitysympäristöön siirtymistä kohdeorganisaation työskentelymallissa korostettiin kattavaa testausta, joka suoritettiin lähinnä massiivisina useiden viikkojen mittaisina testausjaksoina (kuvio 7). Nykyisessä Agile -ympäristössä tämän mallin toteuttaminen lyhyessä jaksossa manuaalisesti ei onnistunut tarpeeksi kattavasti, koska kyseisessä mallissa keskityttiin vain muutokseen ja kasvatettiin samalla niin sanottua teknistä velkaa eli sellaisen toiminnallisuuden määrää, jota ei testata muutoksien yhteydessä laisinkaan (Kohdeorganisaation johdon haastattelu 2, 2012).



Kuvio 7: Kohdeorganisaation vanhan testauskäytännön kuvaus (ICT-alan yrityksen materiaali 2012).

Jotta edellä mainittu ongelma saatiin ratkaistua, oli kohdeorganisaation testauskäytäntöjä muutettava paremmin Agile-ympäristöön sopivaksi. Näin ollen otettiin käyttöön automaattinen testaus (Test Automation), jolloin keskityttiin eri komponenttien, sovellusten sekä sovellusten välisten riippuvuuksien testaamiseen. Automaattisella testauksella ja erityisesti käyttämällä regressiotestaamista, teknistä velkaa saatiin pienennettyä sprintti sprintiltä. (Kohdeorganisaation johdon haastattelu 2, 2012.) Kuvista 8 ilmenee, minkälainen on kohdeorganisaation nykyinen testausprosessi.



Kuvio 8: Kohdeorganisaation nykyinen testausprosessi (ICT-alan yrityksen materiaali 2012).

Agile-kehitysympäristön toimivuuden ja sujuvuuden kannalta keskeisessä osassa ovat toimivat työkalut, jotka tukevat kehitysympäristön eri vaiheita. ICT-alan yritys ja täten luonnollisesti myös kohdeorganisaatio olivat valinneet keskeiseksi työkalukseensa Accept IT toolin. Accept Corporationin kehittämä työkalu on käytössä useissa nykypäivän yrityksissä ja kehitetty nimenomaan Agile-ympäristön tarpeisiin soveltuvaksi. Työkalu tukee hyvin Agile-ympäristöä sen monipuolisuutensa vuoksi.

Accept IT toolissa on monia Agile -ympäristöä tukevia ominaisuuksia, kuten resurssisuunnittelun apuväline eli Burndown. Sillä voidaan seurata, miten työtehtävä on konkreettisesti toteutunut verrattuna suunnitelmaan. Tämä auttaa myös tulevaisuuden resurssisuunnitelman tekemistä. Työkalusta näkyvät myös muun muassa eri työtehtävien priorisoinnit, tekniset toteutukset, hyväksynnät ja työmäärän arviot. Accept IT toolista voidaan valita eri tiimille kohdennettu näkymä ja pääsee näin ollen suoraan katsomaan kyseiselle tiimille oleellisia tietoja aiheista vaatimukset (requirements), suunnitelmat (roadmaps), isot työpyynnöt (needs), tiimit (teams), työtehtävät (tasks) ja raportointi (reporting).

Kohdeorganisaation koko Agile -ympäristö on nivoutunut Accept IT toolin ympärille - käytännössä kaikki tapahtuu Accept IT toolin kautta. Kaikki projekteihin liittyvä kommunikaatio (muutokset, työpyynnöt, vastaukset ja kommentoinnit) sekä dokumentaatio tapahtuvat Accept IT toolin kautta ja tieto jää näin ollen kaikkien saataville. Työkalu käytännössä korvaa sähköpostitse käytävän viestittelyn projektien osalta lähes kokonaan.

4 Tutkimusmenetelmät ja tutkimuksen toteuttaminen

Tutkimusmenetelmät voidaan käytännössä jakaa kahteen eri tyyppiin, kvantitatiivisiin eli määrällisiin ja kvalitatiivisiin eli laadullisiin menetelmiin. Leskisen mukaan (Leskinen 1995, 13) kvantitatiivisella ja kvalitatiivisella tutkimuksella on useita eroavaisuuksia. Kvantitatiivista tutkimusta luonnehditaan selittäväksi tutkimukseksi ja kvalitatiivista tutkimusta puolestaan ymmärtäväksi tutkimukseksi. Kvalitatiivinen tutkimus on riippuvainen teoriasta, kun taas kvantitatiivinen tutkimus ei niin suoranaisesti ole. Kvantitatiivinen tutkimus perustuu otannan edustavuuteen, kun taas kvalitatiivinen perustuu teoriaan. Kvantitatiiviselle tutkimukselle ominaista on se, ettei mittaaminen ole sattumanvaraista, kun taas kvalitatiiviselle tutkimukselle ominaista on se, että aineiston analyysi on uskottavaa ja arvioitavaa. Oleellista kvantitatiiviselle tutkimukselle on myös se, että siinä on tilastollisia korrelaatioita, kun taas kvalitatiiviselle tutkimukselle oleellista on juuri tulkinat ja yksityiskohtaiset kuvaukset.

Tässä opinnäytetyössä käytettiin kvalitatiivista tutkimusmenetelmää. Kvalitatiivisen tutkimuksen tavoitteena on empiiriseen aineistoon perustuva asioiden ja ilmiöiden tulkitseminen sekä niiden kuvaileminen (Leskinen 1995, 13). Itse tutkimusstrategiana voidaan käyttää useitakin eri vaihtoehtoja, kuten tapaustutkimusta, mikä valittiin tämänkin opinnäytetyön tutkimusstrategiaksi.

Aaltolan ja Vallin (2007, 185) mukaan ”tapaustutkimukselle on luonteenomaista, että yksittäisestä tapauksesta (tai pienestä joukosta toisiinsa suhteessa olevia tapauksia) tuotetaan yksityiskohtaista, intensiivistä tietoa”. Aineistonkeruu tapahtuu käyttämällä eri tiedonkeruumenetelmiä.

Tapaustutkimus on kokonaisvaltainen ja sen teossa on mahdollista käyttää eri tutkimusmenetelmiä, kuten kvantitatiivisia tai kvalitatiivisia menetelmiä. Tapaustutkimuksen tutkimusprosessin on oltava näkyvä, jotta tutkimustulosten johtopäätökset voidaan todistaa lukijalle niin, että tutkimuksen luotettavuus on helposti nähtävissä. (Aaltola & Valli 2007, 185-186.)

Eri näkökulmista katsottuna tapaus-käsite tarkoittaa eri asioita. Aaltolan & Vallin (2007, 187) mukaan tapaus voi tarkoittaa tutkimuksen kohdetta eli objektia, tutkimuskohteen osaa eli yksikköä tai havaintoa. Tutkimuksen kohteesta eli objektista puhutaan, kun kyseessä on menetelmällinen kielenkäyttö. Tilastollisen tutkimuksen yhteydessä puolestaan käytetään nimitystä tutkimuskohteen osa eli yksikkö. Kyselylomaketutkimuksen ollessa kyseessä käytetään tapauksesta termiä havainto.

Tapaustutkimuksesta voidaan sanoa, että se on lähestymistapa. Tapaustutkimus on erittäin joustava ja monipuolinen. Tämä ominaispiirre mahdollistaa tutkittavan asian kokonaisvaltaisemman tutkimisen ja ymmärtämisen. Oleellista tapaustutkimusta tehdessä on se, että tutkija itse konkreettisesti tietää mitä ja miten hän tutkimuksen kohteena olevaa asiaa tutkii. (Aaltola & Valli 2007, 194.)

Tutkimuksen aineistoa voidaan kerätä useilla eri metodeilla, kuten esimerkiksi haastattelun avulla. Haastattelua käytettiin aineiston hankintametodina myös tässä opinnäytetyössä. Opinnäytetyön haastattelutyypinä oli teemahaastattelu.

Haastattelulle on ominaista, että se on etukäteen suunniteltu. Lisäksi se on haastattelijan hallinnoima sekä motivoima luottamuksellinen vuorovaikutustilanne. Haastattelu sopii aineiston keruumenetelmäksi muun muassa silloin, kun tutkimusaiheiden järjestyksiä vaihdellaan, halutaan mahdollisimman suuri vastausprosentti, tutkittavana on herkkiä ja tunteita herättäviä asioita sekä halutaan hyvin kuvailevia vastauksia. (Hirsjärvi & Hurme 1985, ks. Metsämuuronen 2006, 113.)

Haastattelu aineiston hankintamenetelmänä soveltuu erittäin hyvin lukuisiin eri tilanteisiin ja on hyvin miellyttävä keino aineiston keräämiseen. Negatiivisena seikkana mainittakoon vaativa ja työläs tulosten analysointi. Haastatteluja on useita erityyppisiä. Ne voidaan jakaa muun muassa strukturoituihin, puolistrukturoituihin ja avoimiin haastatteluihin. Strukturoitua haastattelua käytetään silloin, kun kyseessä on yhtenäinen ryhmä, haastateltavia on paljon ja haastattelu halutaan saada tehtyä nopeasti. Kysymykset ovat lomakkeella samassa järjestyksessä. Puolistrukturoitua haastattelua eli teemahaastattelua puolestaan käytetään tilanteissa, joissa tutkimuksen kohteena on hyvin tunteita herättäviä asioita ja halutaan saada selville heikosti tiedostettuja asioita. Teemahaastattelussa on ennalta valitut teemat, mutta itse haastattelutilanteessa kysymysten muoto ja esittämisjärjestykset voivat vaihdella hyvinkin vapaasti. Avoin eli ei-strukturoitu haastattelu on käytännössä avoin keskustelutilanne, joka etenee haastateltavan ehdoilla - haastatteliija ei johda eikä hallinnoi keskustelua. Avoin haastattelu valitaan aineiston keruumenetelmäksi muun muassa silloin, kun tutkittavana on kaukana menneisyydessä olevat asiat, tutkittavien määrä on pieni sekä

tutkittava aihe on hyvin henkilökohtainen. (Hirsjärvi & Hurme 1985, ks. Metsämuuronen 2006, 112-115.)

Teemahaastattelu soveltuu sekä kvantitatiiviseen että kvalitatiiviseen tutkimusmenetelmään. Hirsjärven & Hurmeen mukaan oleellista teemahaastattelulle on se, ”ettei se edellytä tiettyä kokeellisesti aikaansaatua yhteistä kokemusta, vaan lähtee oletuksesta, että kaikki yksilön kokemuksia, ajatuksia, uskomuksia ja tunteita voidaan tutkima tällä menetelmällä.” Teemahaastattelun määritelmässä ei ole määrätty ennalta, montako haastattelua pitää suorittaa eikä sitä, miten syvällisesti aihetta haastattelussa käsitellään. Keskeisessä roolissa ovat teemat, joiden aihe-alueista haastattelussa keskustellaan ja joiden mukaan haastattelussa edetään. Haastateltavien tulkinnat hänen kertomistaan asioista korostuvat teemahaastattelutilanteessa erityisen paljon. Myös vuorovaikutus haastattelijan ja haastateltavan välillä on keskeisessä roolissa siihen, minkälaisen merkityksen asiat sitten saavat. Puolistrukturoiduksi menetelmäksi teemahaastattelun tekee se, että haastattelun teemat ovat kaikille haastateltaville samat vaikka kysymysten muoto ja järjestys sitten vaihtelevatkin. (Hirsjärvi & Hurme 2000, 48.)

4.1 Tutkimusmenetelmän valinta

Tämän opinnäytetyön tutkimuksellinen osuus toteutettiin kvalitatiivisella eli laadullisella tutkimusmenetelmällä valitsemalla tutkimusstrategiaksi tapaustutkimus. Tapaustutkimus soveltuu hyvin juuri tutkimuksen kohteena olevan kaltaisen, yksittäisen organisaation tutkimiseen. Tapaustutkimuksen aineisto kerättiin teemahaastattelun avulla haastatteleamalla kaikki tutkimuksen kohteena olevan IT-organisaation työntekijät, 24 henkilöä.

Teemahaastattelussa oli kolme eri teemaa: teema 1: Agile käsitteenä ja työympäristönä, teema 2: työmotivaatio, työssä jaksaminen ja työkuorma ja teema 3: parannus - ja kehitysehdotukset. Teemahaastattelulomakkeet ovat liitteissä 1 ja 2.

Teemahaastattelut suoritettiin kohdeorganisaation tiloissa olevassa neuvotteluhuoneessa kesän 2012 aikana. Haastattelut suoritettiin arkipäivisin toimistoajan klo 8-17 puitteissa. Yhden haastattelutuokion kesto oli noin 30-45 minuuttia/haastateltava. Haastattelut suoritettiin pääsääntöisesti suomen kielellä, mutta koska muutama henkilö oli eri maasta, heille pidettiin haastattelut englanniksi. Haastattelutilanteet nauhoitettiin. Lisäksi haastattelun aikana tietoja kirjattiin manuaalisesti ajatuskarttaan.

4.2 Teemahaastattelututkimuksen tulokset ja tulosten analysointi

Aineiston purkaminen ja analysointi toteutettiin tietokoneella, Microsoft Office-työkalujen avulla. Äänitallenteet litteroitiin ja haastateltavien vastaukset eroteltiin toisistaan kirjaintunnisteella. Teemahaastattelun avulla saatu aineisto analysoitiin haastatteluteemoittain käyttämällä apuna nelikenttä-analyysiä eli vahvuudet/mahdollisuudet/heikkoudet/uhat - näkökulmaa.

4.2.1 Teema 1: Agile käsitteenä ja työympäristönä

Suurimmalle osalle haastatelluista Agile-käsite oli aika vieras ennen sen käyttöönottoa kohdeorganisaatiossa. Siitä ei tiedetty joko yhtään mitään tai korkeintaan tiedettiin hyvin suppeasti mitä Agile -termi ylipäätään tarkoittaa. Vain ihan muutamalla haastatelluista oli konkreettisesti työkokemusta Agile-ympäristössä työskentelemisestä ennen kuin menetelmä otettiin käyttöön kohdeorganisaatiossa.

Pääsääntöisesti vastaajat olivat sitä mieltä, että yrityksen tarjoamia koulutuksia aiheesta oli ollut riittävästi tarjolla ja oli vain itsestä kiinni, osallistuiko niihin vai ei. Tosin koulutuksen sisältö olisi saanut olla yksityiskohtaisempaa ja räätälöidämpää juuri kohdeorganisaation ympäristöä ja tarpeita ajatellen. Aika moni kuitenkin oppi ja opetteli Agile-menetelmän konkreettisesti oman työnsä kautta ilman aiheeseen liittyvää koulutusta ja koki oppineensa aiheesta näin riittävästi. Suurin osa haastateltavista oli sitä mieltä, ettei enää tätä nykyä uusia Agileen liittyviä koulutuksia tarvita, mutta ne jotka olivat koulutuksen kannalla, toivoivat sen olevan lähinnä kertaustyyppistä ja asennemuutokseen liittyvää koulutusta sekä verkostoitumistapaamisia eri yrityksissä työskentelevien vastaavaa menetelmää käyttävien kollegoiden kanssa. Näiden lisäksi toivottiin myös koulutusta Agilen uusista virtauksista. Osa haastateltavista koki, että myös kohdeorganisaatioon Agile-menetelmän käyttöönoton jälkeen liittyneille olisi pitänyt järjestää koulutusta toimintatapoihin ja käytettäviin työkaluihin liittyen.

Scrum-tiimien kokoonpanoja ja kokoja lähes kaikki haastateltavat pitivät sopivina. Koettiin myös, että oikeat henkilöt olivat oikeissa tiimeissä. Teemahaastattelussa todettiin, että jos tiimeiltä haluttaisiin nopeampia tuloksia, niin sitten pitäisi myös resurssajakin olla enemmän käytettävissä.

Suurin osa teemahaastatteluun osallistuneista oli sitä mieltä, että heidän käyttämänsä Agile-menetelmä oli hyvin selkeä ja tehokas tapa työskennellä ja oli selkeyttänyt kohdeorganisaation toimintaa huomattavasti. Koettiin, että lyhyissä 3 viikon säännöllisissä sykleissä oli miellyttävää työskennellä, koska Scrum-tiimien työtehtävät, kuten kehitystyö ja

testaaminen, oli pilkottu pienempiin kokonaisuuksiin. Näin niitä oli huomattavasti helpompi hallita. Vahvuudeksi koettiin myös se, että tiimit olivat itseorganisoituvia ja tiimin sisällä pystyttiin useimmiten päättämään kuka teki sprintin aikana mitään. Useat haastateltavat kokivat erittäin positiivisena asiana sen, että Agilen myötä oli mahdollista sprintin ajan keskittyä nimenomaan itselle määritellyn työtehtävään ja kaiken muun, mitä ei kyseiselle sprintille oltu suunniteltu, pystyi sulkemaan pois. Agile-menetelmä oli tuonut kohdeorganisaation toimintaan myös avoimuutta – näkyvyys sekä Scrum-tiimien välillä että menossa ja tulossa oleviin työtehtäviin oli lisääntynyt. Työjonot olivat selkeitä ja jokainen tiesi konkreettisesti aina kolme viikkoa kerrallaan, mitä työtehtäviin kuului. Lisäksi kommunikaatio tiimiläisten välillä oli haastateltavien mielestä erittäin tehokasta ja aktiivista.

Alihankkijoiden kanssa tehtävä yhteistyö koettiin selkeäksi, koska heille pystyttiin rajaamaan selkeästi jokin työtehtävä tai asia, jonka he sitten tekivät sovituksessa ajassa. Alihankkijoita myös pidettiin taitavina ja osaavina henkilöinä. Koettiin myös, että pääsääntöisesti he noudattivat kohdeorganisaation sääntöjä ja ohjeistuksia hyvin ja tekivät juuri ne asiat, joita heitä oli pyydettykin tekemään. Myös henkilön omalla persoonalla koettiin olevan merkitys siihen, miten hyvin yhteistyö kohdeorganisaation ja alihankkijan kesken onnistui.

Haastateltavista suurin osa oli sitä mieltä, että Agile-menetelmä sopi hyvin kohdeorganisaation eri ohjelmistokehitysprojekteihin sekä prosessien kehittämisprojekteihin, koska pystyttiin keskittymään juuri siihen tiettyyn asiaan, mikä Product teamilta tuli kulloinkin tehtäväksi. Agile-menetelmällä työskentely toi projekteihin myös kaivattua järjestelmällisyyttä.

Haastateltavien mielestä Agilen käyttäminen oli tuonut selkeyttä ja nopeutta testaamiseen, koska testattavat kokonaisuudet olivat nyt pienempiä ja selkeämpiä. Testattavaa ainesta tuli tätä nykyä jopa päivittäin. Näin ollen myös palautteen antaminen testattavasta kokonaisuudesta oli nopeaa. Testaaminen koettiin myös helpommaksi kuin ennen, koska voitiin keskittyä vain jonkin tietyn asian testaamiseen, eikä tarvinnut testata kaikkea mahdollista samaan aikaan. Scrum-tiimeissä yritettiin koko ajan toimia niin, että myös testaajilla olisi riittävästi töitä heti sprintin alusta lähtien, eikä vain sprintin loppuvaiheessa. Automaatiotestausta pyrittiin käyttämään enenevässä määrin.

Haastateltavat kokivat, että Agile-ympäristössä työskentelemisen hallinnoimiseen tarvittiin jokin toimiva työkalu. Kohdeorganisaatiolla oli käytössään Accept IT tool (AIT), joka koettiin tärkeäksi. AIT:n koettiin tuovan läpinäkyvyyttä siihen, mitä kukin oli tekemässä ja mitä kukin oli tehnyt. Tämän lisäksi AIT:n käyttö toi yhdenmukaisuutta, koska kaikki kirjasivat tietonsa samaan paikkaan. AIT:n task board-näkymää useimmat haastateltavat pitivät hyvänä ja selkeänä. AIT:ta pidettiin yleisesti ottaen työkaluna, joka tuntui ensin monimutkaiselta,

mutta kun sitä oppi käyttämään, niin sen tarjoamia ominaisuuksia pidettiin hyvinkin monipuolisina ja toimivina.

Teemahaastattelusta kävi ilmi, että kohdeorganisaation Scrum-tiimien tapa toimia Agile-ympäristössä oli hyvinkin erilainen. Osa Scrum-tiimeistä noudatti hyvinkin tarkasti sovitun Agile-käytänteitä, mutta osa Scrum-tiimeistä taas vapaammin. Haastateltavat kokivat myös, että osa Scrum-tiimeistä oli alkanut huomaamattaan pikku hiljaa luisua jonkin verran pois hyvin alkaneista Agile-käytänteistään.

Osa haastateltavista koki myös byrokratian lisääntyneen, kun kaikki tekeminen piti kirjata ylös työkaluihin. Osa kirjasi samoja tietoja jopa kahteen eri järjestelmään (Jira + AIT). Tämän koettiin vievän työstä tehokkuutta pois, kun aikaa koettiin menevän niin paljon tietojen kirjaamiseen.

Osa haastateltavista koki, että kaikki kohdeorganisaation työntekijät eivät olleet vielä saaneet sisäistettyä Agile-menetelmän toimintatapoja ajatusmaailmaansa. Toisin sanoen koettiin, etteivät ihan kaikki työntekijät olleet vielä täysin sitoutuneita kohdeorganisaation käyttöönotettavaan toimintatapaan, vaan osa heistä toimi vielä osittain vanhan toimintatavan mukaan. Koko scrum-tiimin pitäisi olla motivoitunut ja sitoutunut, jotta Agile-menetelmää voitaisiin käyttää sujuvasti.

Osa haastatelluista oli sitä mieltä, etteivät ylempi johto ja kohdeorganisaatio puhuneet samaa kieltä. He kokivat, ettei ylempi johto oikein ymmärtänyt kohdeorganisaation käyttämää toimintatapaa, koska se vaati usein tekemään asioita työjonojen ja sprinttien ulkopuolelta. Tämä oli ristiriidassa Agilen periaatteiden kanssa ja teki työnteon ajoittain aika haasteelliseksi.

Alihankkijoiden kanssa tehtävä yhteistyö koettiin niin, että vaikka alihankkijoita pidettiinkin osaavina henkilöinä, heidän kanssaan tehtävä yhteistyö koettiin haasteelliseksi, koska monet heistä eivät olleet työskennelleet fyysisesti samalla paikkakunnalla kohdeorganisaation kanssa ja yhteydenpito oli tapahtunut pelkästään virtuaalisesti. Eri paikkakunnalla työskennelleiden alihankkijoiden työskentelytavan ei oikein koettu olevan Agilea, koska päivittäinen työskentely heidän kanssaan ei ollut niin tehokasta ja tiivistä, mitä sen Agile-ympäristössä pitäisi olla. Esimerkiksi päivittäisten scrum-kokousten pitäminen oli ollut haastavaa, kun suurin osa alihankkijoista oli kokouksessa läsnä pelkästään puhelinyhteyden kautta. Heidän myös koettiin jäävän ulkopuoliseksi monista konkreettisista työasioista juuri siksi, etteivät he olleet fyysisesti läsnä. Jos alihankkijat siirtyisivät samoihin toimistotiloihin, yhteistyön sujuvuutta ei koettu ongelmalliseksi. Alihankkijoille oli usein määritelty jokin prosentuaalinen määrä, minkä verran kyseinen henkilö oli tiimin käytettävissä ja näin ollen suunniteltaessa

sprintin tehtävälistaa koettiin hankalaksi arvioida, kuinka paljon kyseistä henkilöä sitten voitaisiin konkreettisesti työllistää sprintin ajaksi. Koettiin myös, ettei oikein voitu kontrolloida sitä, miten he tekivät töitä. Agileen siirtymisen koettiin olevan ainakin osasy alihaankkijoiden kanssa tehtävän yhteistyön päättymiseen kohdeorganisaatiossa.

Testaamiskäytänteistä useat haastateltavat kokivat, että testaamisen pitäisi olla enemmän automatisoidumpaa kuin mitä se nyt oli. Manuaalista testaamista oli käytössä vielä liian paljon. Myös testattavien kokonaisuuksien koettiin välillä olevan liian isoja, koska tällöin testaajat joutuivat odottelemaan testattavaa pitkään ja testaamaan pääsi vasta sprintin loppuvaiheessa.

Erilaisia haasteita Agile-ympäristössä työskentelyssä koettiin olevan paljon. Suurimmiksi haasteiksi koettiin olevan sprinttien ajan töiden suunnittelu niin, että työt jakautuisivat tasaisesti koko sprintin ajalle. Koettiin, että oli vaikea arvioida suunnitteluvaiheessa etukäteen, kuinka paljon mihinkin työtehtävään eli storyyn tuli menemään aikaa. Oli myös käynyt niin, ettei sprintin aikana oltukaan pysytty aikataulussa eikä saatu kehitettävää tuotetta valmiiksi. Aika usein kolmen viikon sprintti-jaksolle tehdyt suunnitelmat muuttuivat. Suunnitelmuutokset olivat joko korjauksia tai sellaisia tilanteita, joissa jokin kriittinen työtehtävä ajoi sprintille suunniteltujen työtehtävien ohi. Muita haasteita koettiin olevan, että koko Agile-kokonaisuuden hahmottaminen ja oman ajatusmallin muuttaminen Agiletta tukevaksi oli osalla vielä kesken. Osa koki myös, että aluksi tuntui siltä, ettei työntekijöihin luoteta vaan että heitä kytätään. Lisäksi haasteeksi koettiin se, että kohdeorganisaatio ei olekaan autonominen ja heitä ohjataan ylemmästä johdosta käsin. Tämä toi ajoittain erittäin suuria ristiriitatilanteita. Hankaliksi koettiin myös tilanteet, joissa jotkin kohdeorganisaation ulkopuoliset tahot, joiden kanssa ohjelmiston kehitysprojekteja yhdessä tehtiin, eivät olleet vielä siirtyneet käyttämään Agile-menetelmää.

Myös työkalujen käyttö, kuten Accept IT tool ja kehitystyössä käytettävät työkalut, koettiin haasteellisiksi, vaikka tärkeässä roolissa olivatkin. Suurin osa haastatelluista koki, että Accept IT toolin pitäisi olla käyttäjäystävällisempi ja selkeämpi. Heistä tuntui, että työkalu oli visuaalisesti aika sekava ja että siellä oli liikaa tietoa esillä samaan aikaan – myös sellaista tietoa, mitä usea työntekijä ei edes käyttänyt tai tarvinnut. Yleinen mielipide oli, että AIT:ta pitäisi jatkokehittää käyttäjäystävällisemmäksi, koska siitä puuttui muun muassa tietojen järjestelymahdollisuus, haku-toiminto sekä ominaisuus, että työkalu muistaisi viimeisen näkymän.

4.2.2 Teema 2: työmotivaatio, työssä jaksaminen ja työkuorma

Konkreettisimmiksi muutoksiksi työtehtävien hoitamisessa koettiin projektien kestot ja niiden laajuudet. Aiemmin projektit olivat massiivisia kokonaisuuksia, jotka saattoivat kestää 4-5 kuukautta, jopa vuodenkin. Usein saattoi käydä jopa niin, että projektit jäivät paikoilleen ja ne jäivät kesken. Nyt nykyisessä Agile-mallissa sprintillä oli tietty pituus ja siihen sovittiin tietyt asiat tehtäviksi. Oleellisia muita konkreettisia muutoksia olivat myös aiempaa tarkempi ja yksityiskohtaisempi dokumentointi sekä definition of done (dod) eli valmiin määritelmä. Monet olivat sitä mieltä että valmiin määritelmän mukaan tuli enemmän tehtyä asioita samalla tavalla ja huolehdittua siitä, että tietyt asiat myös tulivat tehtyä. Myös suhtautumistavassa avunpyyntöihin koettiin tapahtuneen muutoksia, nyt asenne apua pyydetessä oli kannustava ja positiivinen.

Työtavoissa tapahtuneita muutoksia oli muun muassa entistä intensiivisempi tiimityö. Aikaisemmin töitä tehtiin paljon itsekseen, mutta tätä nykyä töitä tuli tehtyä erittäin paljon läheisessä yhteistyössä muiden Scrum-tiimiläisten kanssa. Positiivisina muutoksina koettiin myös se, että sähköpostia ei enää tullut niin paljon kuin aikaisemmin, eikä tarvinnut istua palavereissa niin paljon kuin ennen. Koettiin, että pystyi keskittymään juuri oleellisiin asioihin. Agile oli tuonut myös läpinäkyvyyttä koko kohdeorganisaation toimintaan.

Suurin osa haastateltavista koki, että haluaa työskennellä Agile-ympäristössä jatkossakin, koska työympäristö rooleineen oli niin selkeä ja täsmällinen työrytmeineen ja pelisääntöineen. Myös sillä oli vaikutusta, minkälaista työtä tehtiin ja minkälaiden ihmisten kanssa. Koettiin, että Agile-menetelmällä työskentely ei ollut pelkästään tekniikan kehittämistä, vaan jatkuvasti kehitettiin myös omia henkilökohtaisia työmenetelmiä ja -tapoja.

Työmotivaatio oli yleisesti ottaen lähestulkoon kaikkien haastateltavien kohdalla hyvä. Suurin osa haastatelluista koki, että työmotivaatio oli parantunut Agile-menetelmän myötä, koska työnkuva oli selkeytynyt. Myös yhdessä tekeminen ja lyhyet iteraatiot olivat työmotivaatioon oleellisesti vaikuttavia seikkoja. Toimittiin enemmän tiiminä ja autettiin muita tiimiläisiä aina tarpeen vaatiessa. Sovellusten käyttöönotto-työmatkoja pidettiin tärkeinä ja niiden koettiin olevan oleellinen asia työmotivaatiota ajatellen koska toivat työhön sopivaa vaihtelua.

Suurin osa haastateltavista piti nykyisiä työtehtäviään riittävän monipuolisina. Uusia mielenkiintoisia työtehtäviä oli tullut usealle haastatelluista Agileen siirtymisen myötä. Oma aktiivisuus ja halukkuus olivat keskeisiä seikkoja työnkuvan monipuolisuuteen. Usean mielestä Agilen myötä työnkuva oli laajentunut ja selkeytynyt.

Sprinttien suunnittelua pidettiin kontrolloituna ja järjestelmällisenä ja Scrum-tiimi pystyi myös yleensä itse vaikuttamaan aikatauluihin. Tiimi katsoi yhdessä, mikä oli milloinkin toteutettavissa ja kannattiko ylipäättään toteuttaa mitään. Lyhyet iteraatiot mahdollistivat sen, että kehitettävää tuotetta voitiin tarkastella usein ja näin tuotteeseen voitiin piankin tehdä tarvittavat muutokset. Kehitettävät asiat olivat selkeästi pienempiä kokonaisuuksia kuin ennen. Ja täten myös asiakas sai tuotoksen nopealla aikataululla. Kehitettävä tuote oli aina tehtävä valmiiksi, eikä sitä voinut jättää vain roikkumaan ja kesken.

Noin puolet haastateltavista koki, että Agile-menetelmä oli selkeyttänyt oman roolin ymmärtämistä entisestään työtehtävissä. Toisaalta saman verran haastateltavista koki puolestaan, että heidän roolinsa oli yhtä selkeä kuin aikaisemminkin eikä Agilella ole ollut vaikutusta tähän asiaan.

Työnteko Agile-ympäristössä koettiin tehokkaaksi. Koska oli etukäteen määritelty tarkkaan, mitä piti tehdä ja tekemiset olivat pienempiä kokonaisuuksia, työ oli myös tehokasta. Myös välitön palaute siitä, oltiinko tehty oikeita asioita, saatiin nopeasti. Tällöin myös ongelmakohdat nousivat nopeammin esille ja niihin löydettiin ratkaisukin nopeammin. Tämä nosti myös tuotteen laatua. Lisäksi koettiin, että omia kehitysideoita otettiin hyvin positiivisesti vastaan ja niitä myös toteutettiin.

Työkuorma oli suurimmalla osalla haastateltavista sopiva, vaikka työkuorma jakaantuikin sprintille epätasaisesti. Sprintin suunnittelu oli työkuorman kannalta oleellinen seikka – jos arvioitiin työtehtävien kestot ja niihin kuluva aika väärin, työkuorma kasvoi liian suureksi ja oli kova kiire. Sprintin suunnittelussa piti ottaa huomioon myös päivystystehtävät, ilman että oli konkreettisesti tiedossa etukäteen, kuinka paljon ne tulevat työllistämään. Työkuormaa on yritetty tasapainottaa niin, että kun on rauhallisempaa, niin viimeisellä sprinttiviikolla tutkaillaan jo seuraavan sprintin työpyyntöjä. Työkuormaan vaikutti oleellisesti myös se, että kuinka paljon haali itselleen töitä sprintin ajaksi sekä se, tuliko ennalta arvaamattomia ongelmatilanteita ja oliko sprintin työtehtävien lisäksi tiedossa päivystysluontoisia töitä.

Vaikka useimpien haastateltavien mielestä esimies oli aiemminkin ollut hyvin tavoitettavissa, niin monet haastatellut kokivat esimies-alaisuuden parantuneen edelleen, koska nyt hän oli aiempaa enemmän konkreettisesti mukana yhteisissä työtehtävissä ja oli täten myös enemmän tietoinen mitä alaiset tekivät. Myös esimiehen kanssa tapahtuva kommunikointi oli entistä aktiivisempaa ja yksityiskohtaisempaa. Lisäksi esimiehen omalla persoonallisuudella koettiin olevan vaikutusta siihen, minkälainen esimies-alaisuus kokonaisuudessaan oli.

Muutama haastateltavista koki, että työmotivaatioon vaikutti oleellisesti se, kenen kanssa oltiin yhteistyössä. Joidenkin tiettyjen henkilöiden kanssa työskentely koettiin haasteelliseksi

ja hankalaksi ja tämä vaikutti myös työmotivaatioon negatiivisesti. Myös kyllästymisen samaa toistaviin työtehtäviin koettiin laskevan työmotivaatiota. Tämän lisäksi yrityksen vaikealla tilanteella koettiin olevan negatiivinen vaikutus työmotivaatioon – tämä ei tosin johtunut Agilesta.

Osalla Agile-ympäristöön siirtyminen kavensi ensi alkuun työnkuvaa, mutta tilanne muuttui aika pian. Työnkuva oli kaventunut joillain esimerkiksi siitä syystä, että käytettävien sovellusten määrä oli laskenut osalla kohdeorganisaation työntekijöistä huomattavasti. Toisaalta taas toisilla oli käytettävien sovellusten määrä lisääntynyt. Koettiin, että jos työtehtävät eivät olleet kovin monipuolisia, niin sitten haluttiin syventyä vielä yksityiskohtaisemmin omaan työtehtäväkenttään. Ratkaisevaa tässäkin oli oma aktiivisuus ja oma-aloitteisuus. Muutama haastateltavista koki työnkuvansa myös sekavaksi. Tähän vaikutti se, minkälaisessa työtehtävässä henkilö oli.

Osa koki, että Agileen siirtymisen myötä oma päätösvalta oli pienentynyt entisestään. Osa heistä myös koki, että vain muutamat tietyt henkilöt saivat päättää asioista mitä ja miten kulloinkin tehdään. Toisaalta oltiin sitä mieltä, että Scrum-tiimit olivat jonkin verran itsenäistyneet, eivätkä tarvinneet kaikessa tekemisessään esimiestä enää siinä suhteessa kuin ennen. Scrum masterin rooli ja esimiehen rooli oli osan haastateltavan mielestä ehkä hieman sekava, epäselvää oli esimerkiksi se, että kummalta tulee kysyä lomista ja muista käytännön asioista.

Työn tehokkuutta koettiin alentavan lisääntynyt tietojen kirjaaminen sekä keskeytykset, eli tilanteet, joissa tuli työtehtäviä sprintin ulkopuolelta. Näitä tehtäviä olivat muun muassa päivystystehtävät, korjauspyynnöt sekä kriittiset työtehtävät, jotka menivät sprintille suunniteltujen työtehtävien ohi. Lisäksi koettiin, että työn tehokkuutta alensi se, että kaikki tiimiläiset eivät vielä olleet mukautuneet Agile-ajatusmaailmaan eivätkä täten toimineet Agilen toimintatapojen mukaisesti. Myös töiden epätasaisen jakaantumisen sprintin aikana koettiin heikentävän työn tehokkuutta. Edellä mainittujen lisäksi osa haastateltavista koki, että työn tehokkuutta vähensi myös se, että sprintit toivat kankeutta työn tekemiseen. Työmäärät eri sprinteissä vaihtelivat paljon ja vaikka olisi ehtinyt tekemään enemmänkin töitä, niin sitä ei saanut tehdä, koska kyseistä työtehtävää ei oltu määritelty kyseisen sprintin tehtävälisalle.

Vaikka jonkin verran olikin eroavaisuuksia Scrum-tiimien välillä, niin suurimman osan mielestä koodareiden ja testaajien työtehtävät painoutuivat liian epätasaisesti sprintin alku- ja loppupuolille. Koodareilla oli kiire sprintin alkupuolella ja testaajilla puolestaan työ kuormittui sprintin loppuun, jolloin heillä oli kiire. Osa haastateltavista piti tätä jopa stressaavana tilanteena. Lisähaasteen kiire-asiaan toi myös se, että kohdeorganisaatiossa oli

jonkin verran uudehkoja työntekijöitä, joilla ei ollut vielä niin vahvaa tieto-taitoa, jota kohdeorganisaation työtehtävissä tarvittaisiin. Lisäksi sprintin ulkopuolelta tulevat työtehtävät kuormittivat, koska tällöin piti tehdä sekä sprintille suunnitellut työtehtävät aikataulussa pysyen että ylimääräiset tehtävät. Osa myös koki, että Agile rajoitti liikaa tekemistä kun piti koko ajan miettiä, miten jokin työtehtävä Agilessa tehtäisiin.

4.2.3 Teema 3: parannus- ja kehitysehdotukset

Haastattelussa tuli esiin useita eri parannus- ja kehitysehdotusta. Suurin osa parannus- ja kehitysehdotuksista liittyi työntekijöiden kompetenssien sekä testausympäristön kehittämiseen. Myös Accept IT toolin ja sprintin kehittämiseen tuli useita ehdotuksia.

Useat haastateltavat olivat sitä mieltä, että Scrum-tiimiläisten kompetenssit pitäisi saada kehitettyä niin, että mahdollisimman moni pystyisi tekemään eri työtehtäviä. Eli käytännössä koodarin pitäisi opetella esimerkiksi prosessinkehitystä ja testaamista ja testaajan puolestaan koodaamista jne. Haastatteluissa ehdotettiin seuraavia konkreettisia toimia kompetenssien lisäämiseksi:

- Työntekijät vaihtelisivat säännöllisesti Scrum-tiimeistä toiseen eli olisivat työnkierrossa eri Scrum-tiimeissä esimerkiksi yhden tai kahden sprintin ajan tai vaikkapa yhden työtehtävän eli taskin ajan
- Koodaajat opettelisivat testaamista ja testaajat koodaamista jne.
- Työntekijät voisivat mennä työnkiertoon muihin Capability-yksikköihin
- Scrum master voisi olla kiertävä
- Koodareiden pitäisi aktiivisesti opetella kohdeorganisaation kehitettäväksi tulleita uusia sovelluksia, ettei olla tilanteessa, jossa vain tietty koodaaja tietää tietystä sovelluksesta
- Kouluttautumismahdollisuus esimerkiksi jossain ohjelmistotalossa, esimerkiksi koodaamista ajatellen
- Oleminen entistä enemmän konkreettisesti mukana sprintin työtehtävien suunnittelussa.

Accept IT tooliin (AIT) liittyvissä parannus- ja kehitysehdotuksissa haastateltavien mielestä Accept IT toolin pitäisi olla käyttäjäystävällisempi, selkeämpi ja nopeampi. Omaa näkymäänsä pitäisi myös itse pystyä räätälöimään niin, että sieltä voisi häivyttää kaikki ylimääräiset ja tarpeettomat tiedot välilehtineen näkyvistä. Työkalusta myös pitäisi saada itse poistettua tarvittaessa omia tekeleitä, tämä ei kaikilta onnistunut. Myös hakutoiminto pitää saada selkeämmäksi. Parannusta toivottiin myös AIT:n suorituskykyyn – se haluttaisiin paremmaksi tai ainakin se tulisi pitää vähintään nykyisellä tasolla. Lisäksi Jiran ja AIT:n välille haluttaisiin saada integraatio.

Useat haastateltavat kokivat, että AIT:ssa olevaa tietoa pitäisi saada listattua ja järjestettyä esimerkiksi aakkosjärjestykseen tai päivämääräjärjestykseen. Myös sille toivottiin muutosta, ettei ID nro vaihtuisi jatkuvasti. Nyt asia on näin kun tehdään ensin need, needistä tehdään story jne. Olisi oleellista myös sopia, mitä ID-nro kyseisestä asiasta käytetään, koska nyt ID-numeroa käytetään hyvin vaihtelevasti. ID-numero pitäisi myös pystyä kopioimaan, nyt se kaikilta onnistunut.

Haastateltavat olivat myös sitä mieltä, että kirjaamiskäytännöistä on sovittava eli miten needit, taskit jne. konkreettisesti AIT:hen kirjataan ja kuka kirjaukset konkreettisesti tekee. Näin voitaisiin välttyä tilanteelta, joissa needit on AIT:ssa useampaan kertaan ja väärin tehtyinä. Koettiin myös, että tietojen kirjaamisen tärkeyttä pitää painottaa, jotta kaikki kohdeorganisaation Scrum-tiimiläiset pitäisivät AIT:ta ajan tasalla omien työtehtäviensä tilanteesta ja etenemisestä.

Testaamiseen liittyvissä parannus- ja kehitysehdotuksissa useimmat haastateltavat olivat sitä mieltä, että testiautomaatiota pitäisi olla lisää. Käytännössä kohdeorganisaatiossa pitäisi olla nimetty henkilö hallinnoimaan testiautomaatio-kokonaisuutta. Testiautomaatioon liittyy läheisesti myös virtuaalilabra, jonka olemassaolo olisi myös tärkeää. Jotta testiautomaation käyttöä voitaisiin lisätä, pitäisi kohdeorganisaation henkilöiden opetella uusia kompetensseja, kuten testiautomaatioskriptien kirjoittamista. Eräs ajatus oli, että koodaaja ja testaaja toimisivat tässä läheisessä yhteistyössä ja tekisivät testiskriptejä yhdessä, kunnes testaajilla on tarvittavat kompetenssit tähän. Testiautomaatio-asioista kannattaisi myös kysellä kohdeorganisaation ulkopuolelta ja sitä kautta mahdollisesti saataisiin myös lisää arvokasta tieto-taitoa aiheeseen liittyen. Haastateltavien mukaan kaikkien kehitettävien applikaatioiden testausta ei tällä hetkellä tosin vielä voida automatisoida. Pitäisi lähteä liikkeelle siitä, että mietitään yhdessä, mitkä ovat ne tapaukset, jotka testataan manuaalisesti ja mitkä testiautomaation avulla.

Haastattelussa todettiin, että testaamisen ja ohjelmiston kehittämisen pitäisi olla enemmän integroituja keskenään, kuin mitä se nyt oli. Eli ne eivät saisi olla erillisiä prosesseja toinen

toisistaan vaan ne pitäisi nivouttaa enemmän yhteen niin, että koodaaja tietää miten hänen kehittämänsä ohjelmistonpätkä koodataan ja päinvastoin. Myös testattavien kokonaisuuksien pitää olla pienempiä, jotta testaamista olisi mahdollisimman aikaisessa sprintin vaiheessa.

Koettiin, että pitäisi myös enemmän testata ristiin asioita. Tällä hetkellä oli liikaa lokeroitunut kuka testaa mitään. Kaikkien pitäisi periaatteessa testata kaikkea. Osa oli myös sitä mieltä, että Scrum-tiimit päättäisivät aina itse, mitä testataan. Ehdotettiin, että tiimit testaisivat erityyppisiä asioita.

Alihankkijoiden kanssa tehtävään yhteistyöhön liittyen haastateltavien yleinen mielipide oli, että jatkossa jos kohdeorganisaatio tekee alihankkijoiden kanssa yhteistyötä, heidän pitää olla fyysisesti läsnä toimistolla, kuten kohdeorganisaationkin työntekijät ovat. Lisäksi heidän pitää olla myös kokoaikaisesti tiimin käytettävissä. Heidän pitäisi myös aktiivisesti käyttää kohdeorganisaation työkaluja ja päivittää työtehtävänsä tarvittaviin tietokantoihin, kuten muutkin tekevät. Alihankkijat pitäisi ylipäätään perehdyttää tarkemmin Agilen toimintatapoihin ja työkaluihin, koska myös heidän odotetaan toimivan samojen sääntöjen mukaan kuin organisaation omatkin työntekijät.

Yleisiin parannus- ja kehitysehdotuksiin liittyen usea haastateltava oli sitä mieltä, että kohdeorganisaation pitäisi panostaa työntekijöidensä yhteishenkeen ja kehittää tiimin keskinäistä toimintaa niin, että saataisiin tiimi puhaltamaan vielä enemmän yhteen hiileen. Koettiin, että Agile-menetelmien mukaan työskentely hakee vielä osittain muotoaan. Tilanteen parantamiseksi koettiin olevan oleellista, että kaikki kohdeorganisaation työntekijät pitäisi saada ajattelemaan asioista samalla tavalla. Näin työt saataisiin entistä sujuvammin tehtyä ja asiakaskin olisi tyytyväinen. Konkreettisenä ehdotuksena olisi esimerkiksi yleisluontoinen yhden tai kahden päivän mittainen koulutustilaisuus tiimiläisille, jossa muistutettaisiin Agilen periaatteista ja käytänteistä. Lisäksi toivottiin verkostoitumistilaisuuksia yrityksen ulkopuolelle, jotta kuulisi myös muissa vastaavissa yrityksissä työskentelevien toimintamalleja ja -tapoja. Näitä hyviä toimintamalleja ja -tapoja voitaisiin sitten mahdollisuuksien mukaan hyödyntää myös kohdeorganisaationkin toimintatavoissa. Lisäksi verkostoitumistilaisuuksilla olisi positiivinen vaikutus Agile-myönteistä asennemuutosta ajatellen. Myös tuotetiimin pitäisi enemmän verkostoitua oman organisaationsa ulkopuolelle yrityksen sisällä ja tarkastella, miten muissa tiimeissä toimitaan. Sieltäkin voisi saada uusia toimivia ja hyviä toimintamalleja ja -tapoja organisaation käyttöön. Myös ylemmälle johdolle toivottiin koulutusta Agilen toimintaperiaatteista, jotta voitaisiin jatkossa välttyä ristiriita-tilanteilta, joissa ylemmältä johdolta tuli kohdeorganisaatiolle työtehtäviä sprintin ulkopuolelta.

Muita konkreettisia yksittäisiä yleisiä parannus- ja kehitysehdotuksia olivat myös:

- Isojen projektien kokonaisuuksien koordinointiin tulisi kiinnittää enemmän huomiota.
- Tiettyjä rooleja, kuten Lead Developer, tulisi selkeyttää.
- Kokouskäytänteitä pitäisi kehittää tehokkaammiksi. Edelleen koettiin olevan liikaa turhia kokouksia ja monissa kokouksissa käsiteltiin samoja asioita. Myös joidenkin palaverien kestot tuntuivat yhä liian pitkiltä.
- Työpisteet pitäisi muuttaa esimerkiksi välisermejä siirtämällä niin, että Scrum-tiimeillä olisi vielä yhtenäisempi työskentelytila.
- Päivittäisille Scrum-tiimeille kaivattiin jotain muuta vapaamuotoisempaa tilaa, kuin mitä nykyinen neuvotteluhuone on.
- Pitäisi mitata säännöllisesti, miten työmäärä-arviot pitävät paikkaansa.
- Harjoittelijajärjestelmä: kun harjoittelijan tietyt kriteerit täyttyvät, hänet voitaisiin ottaa mukaan osaksi Scrum-tiimiä ja hän tekisi Scrum-tiimille määriteltyjä työtehtäviä oppien samalla Agile-työtavan.
- Tuotoksien vieminen järjestelmästä toiseen pitäisi olla automatisoidumpaa. Automatisoidumpaa pitäisi myös olla sovelluksen uuden version käyttöönottoprosessi eri vaiheineen vietäessä sitä testiympäristöön, koulutuksiin sekä tuotantoihin. Tällä hetkellä prosessissa on vielä liikaa manuaalista työtä.

Sprintteihin liittyvistä parannus- ja kehitysehdotuksista osa haastateltavista oli sitä mieltä, että sprintin pituuden voisi muuttaa kolmesta viikosta kahteen viikkoon. Tällöin myös sprinttien alkamisesta ja loppumisesta samaan aikaan voitaisiin luopua ja olisikin Scrum-tiimikohtaiset aikataulut. Eli käytännössä osa tiimeistä voisi olla kahden viikon sprinttijaksoissa ja osa kolmen. Tällöin tulisi panostaa erityisesti tiimien väliseen yhteistyöhön ja kommunikointiin, jotta tiedetään, mitä Scrum-tiimeissä tapahtuu.

Haastattelussa ehdotettiin myös sellaista työskentelytapaa, että vielä joustavampi toimintamalli sprintille olisi sellainen, että ei lyötäisikään etukäteen täysin lukkoon sprintin 3 viikon työtehtäviä, vaan Scrum-tiimit voisivatkin joustavammin ottaa työtehtäviä työjonosta tehtäväksi aina kun se on mahdollista. Eli käytännössä otettaisiin aina työjonosta prioriteettijärjestyksessä yksi tehtävä kerrallaan ja tehtäisiin se valmiiksi ja sen jälkeen otettaisiin työjonosta taas seuraavat tehtävät jne. Näin sitten tehtäisiin niin monta tehtävää, mitä sprintin ajan kolmessa viikossa ehditään. Tällä menetelmällä saataisiin työjonon

käsittelyä joustavammaksi ja ehdittäisiin tekemään todennäköisesti jopa enemmän työtehtäviä kuin mitä tätä nykyä ehditään kolmen viikon aikana tehdä. Tällöin myös nykyisenlaisesta sprint-suunnittelusta voitaisiin luopua, ja muissa yhteyksissä sitten arvioitaisiin storyjen kokoja. Työtehtävien koot olisivat näkyvissä työjonossa ja sieltä sitten otettaisiin korkeimmalla prioriteetillä oleva työtehtävä. Product team päättäisi mikä on prioriteettijärjestys. Muita konkreettisia sprinttiin liittyviä parannus- ja kehitysehdotuksia olivat: töiden (koodaaminen ja testaaminen) pitää jakautua tasaisemmin sprintin ajalle, töiden suunnittelua pitäisi kehittää ja ottaa käyttöön realistisemmat resursointiarviot sekä sprintin alussa tehtävät töiden määrittelyt pitäisi tehdä aiempaa tarkemmin.

5 Pohdinta ja johtopäätökset

Yleisesti ottaen voidaan todeta, että tämän opinnäytetyön tutkimuksen kohteena olevan organisaation yleinen työilmapiiri oli rento ja positiivinen. Kaikki työntekijät olivat motivoituneita ja sitoutuneita työntekoonsa. Myös organisaation johto oli sitoutunut tehtäväänsä ja oli aidosti kiinnostunut työntekijöistään ja heidän hyvinvoinnistaan. Oleellinen asia organisaation johdon ja työntekijöiden väliseen ilmapiiriin oli mitä ilmeimmin se, että kohdeorganisaation johto oli myös itse mukana organisaationsa työtehtävissä ja näin ollen tiesi konkreettisesti, minkälaisia työtehtäviä heidän organisaationsa työntekijöillä oli meneillään.

Jotta Agile-menetelmällä työskentely olisi mahdollisimman tehokasta, pitää ensin ymmärtää, mitä tämä käsite tarkoittaa. Oleellista on myös sisäistää oman työyhteisönsä käyttämät toimintamallit ja säännöt. Kohdeorganisaation työntekijöistä suurin osa oppi heidän käyttämänsä Agile-menetelmän konkreettisesti oman työnsä kautta. Pidempään kyseisessä organisaatiossa työskennelleille tämä oppimistapa oli toimiva, mutta uusille työntekijöille Agile-käsite vaatii pidemmän omaksumisajan, koska myös organisaatio itsessään on heille vieras. Jatkossa olisikin suotavaa, että kohdeorganisaation uusien työntekijöiden perehdyttämiseen Agile-menetelmään ja organisaation toimintatapoihin kiinnitettäisiin aikaisempaa enemmän huomiota. Yksi keino olisi esimerkiksi tehdä organisaation käyttämästä räätälöidystä menetelmästä ja toimintatavoista oma koulutuspakettinsa, jonka avulla sitten nimetty Scrum master aina perehdyttäisi uudet henkilöt kohdeorganisaation toimintamalleihin. Vaihtoehtoisesti voisi toimia myös niin, että jokaisen Scrum-tiimin Scrum master perehdyttäisi tiimiinsä tulevan uuden työntekijän kohdeorganisaation Agile-periaatteisiin. Tällä tavoin uusi työntekijä voisi nopeammin sisäistää uuden työympäristönsä. Kuten jo aiemminkin tässä opinnäytetyössä on todettu, Scrum master on hyvin keskeisessä asemassa. Hän vastaa konkreettisesti siitä, että organisaatiossa ymmärretään mitä Scrum on ja että organisaatiossa myös toimitaan Scrumin sääntöjen mukaisesti.

Kohdeorganisaation työntekijät suhtautuivat enimmäkseen positiivisesti Agile-menetelmään ja lähes kaikki heistä kokivat Agile-menetelmällä työskentelemisen selkeäksi ja mielekkääksi. Suurin osa kohdeorganisaation työntekijöistä oli tyytyväisiä tämänhetkisiin toimintatapoihinsa ja koki työympäristönsä jo nyt riittävän tehokkaaksi. Yleisellä tasolla työnteon teki mielekkääksi ja tehokkaaksi se, että tehtävät kokonaisuudet olivat pienempiä kuin ennen. Oleellisessa osassa tässä oli sprinteissä eli säännöllisissä kolmen viikon jaksoissa työskentely, jonka koettiin tuovan toivottua järjestelmällisyyttä työntekoon. Lisäksi koettiin erityisen positiivisena seikkana työnteon kannalta se, että Agileen siirtyminen oli tuonut läpinäkyvyyttä koko kohdeorganisaation toimintaan ja näin ollen tiedettiin tarkasti, mitä tapahtuu ja milloin. Läpinäkyvyshän on yksi Scrumiin liittyvästä ominaispiirteestä, johon liittyy muun muassa yhteisesti määritelty termistö sekä valmiin määritelmä. Nämä molemmat ominaispiirteet olivat tärkeässä roolissa myös kohdeorganisaatiossa.

Lisää tehokkuutta kohdeorganisaation toimintaan saataisiin muun muassa tasaamalla työntekijöiden kompetensseja niin, että käytännössä kaikki osaisivat tehdä kaikkea. Nyt kompetenssit eri henkilöiden välillä vaihtelivat liikaa ja työt monasti painottuivat sprintin alkuun tai loppuun työnkuvasta riippuen. Jos sama henkilö osaisi sekä koodata että testata, toimintaa saataisiin jo paljon tehokkaammaksi, koska näin saataisiin tuotteen kehitystyö ja testaaminen paremmin integroitumaan keskenään, eivätkä ne olisi enää niin erillisiä prosesseja. Tällöin sama henkilö voisi testata kehittämäänsä tuotetta aina kehitystyön edetessä. Samalla saataisiin myös tasoitettua työntekijöiden työkuormia. Lisäksi työntekijöiden ammattitaito kasvaisi entisestään. Säännöllinen työkierto eri tiimien välillä tai oman tiimin sisällä olisi yksi keino kompetenssien lisäämiseksi. Kohdeorganisaatio voisi ensi alkuun kartoittaa, ketkä kaikki olisivat halukkaita työnkiertoon. Suotavaa olisi, että mahdollisimman moni. Tämän jälkeen sovittaisiin konkreettisesti milloin ja miten työkiertoa aletaan toteuttaa. Sovituista työnkierroista myös pidettäisiin kiinni. Työnkierron avulla saataisiin myös opeteltua testiautomaatioon tarvittavia kompetensseja, kuten testiskriptien kirjoittamista sekä lisäksi opeteltua kohdeorganisaation käyttöön tulleita uusia applikaatioita. Myös työntekijöiden oma-aloitteista kompetenssien kehittämistä tulee kannustaa ja aktivoida ja erityisesti siihen pitää antaa mahdollisuus.

Kohdeorganisaation tulisi tulevaisuudessa lisätä testiautomaatiota mahdollisimman paljon, jotta testaaminen saataisiin kehitettyä paremmin Agile-ympäristöön soveltuvaksi. Testiautomaation käyttöönottoaminen on aina pitkä prosessi ja se vaatii tiimiltä sopeutumista ja monien uusien taitojen opettelemista. Testiautomaation käyttöönoton lisäämistä kohdeorganisaatiossa helpottaisi se, jos nimettäisiin tietty vastuuhenkilö, joka koordinoisi koko testiautomaatioprosessia tarvittavine perehdytyksineen, koulutuksineen ja laitteineen.

Myös sprinttien suunnittelua kehittämällä saataisiin enemmän järjestelmällisyyttä ja tehokkuutta työntekoon. Kohdeorganisaatio voisi aloittaa säännöllisen seurannan esimerkiksi kvartaaleittain siitä, miten sprinteille suunnitellut resursoinnit ovat toteutuneet. Tämä auttaisi tulevaisuudessa resursointisuunniteluissa saamaan sprinttien työmäärä-arvioita realistisemmiksi. Sprintin suunnitteluvaiheessa pitäisi olla läheisemmässä yhteistyössä myös ylemmän johdon kanssa, jotta he saisivat tarkemman kokonaiskuvan ja ymmärryksen siitä, minkälaisia työtehtäviä kullekin sprintille on aina suunniteltu. Näin voitaisiin yhdessä järjestelmällisemmin varautua heiltä tuleviin kriittisiin työtehtäviin niin, etteivät kriittiset työtehtävät ja sprintille suunnitellut työtehtävät kärsi toisistaan. Tärkeää olisi ylipäättään pitää ylemmän johdon kanssa säännöllisiä tilaisuuksia, joissa kerrattaisiin/käsiteltäisiin kohdeorganisaation toimintatavat ja mallit sekä sprinttien työtehtävien suunnittelut. Mukana kohdeorganisaatiosta voisivat olla esimerkiksi Product team sekä Scrum masterit.

Agile-menetelmä oli selvästi lisännyt monen kohdeorganisaation työntekijän työmotivaatiota ja työtehokkuutta, koska työskenteleminen oli tätä nykyä paljon selkeämpää. Tähän vaikutti erityisesti se, että sai keskittyä aina olennaiseen työtehtävään sprintin ajan ja työtehtävillä oli selkeät päättymis- ja loppumisajat. Työilmapiiriä ja samalla tehokkuutta saataisiin edelleen lisättyä kohdeorganisaation tiimihenkeä kehittämällä. Agilen peruseriaatteita ovatkin juuri rehellinen työtapa sekä hyvässä hengessä tapahtuva tehokas tiimityö. Verkostoitumistilaisuus yrityksen ulkopuolelle olisi erittäin tehokas keino, koska tällöin kuulisi myös vastaavissa yrityksissä työskentelevien toimintamalleja ja -tapoja, joita voitaisiin sitten mahdollisuuksien mukaan hyödyntää myös kohdeorganisaationkin toimintatavoissa. Mitä todennäköisemmin verkostoitumistilaisuuksilla olisi hyvin positiivinen vaikutus Agile-myönteistä asennemuutosta ajatellen niille henkilöille, jotka eivät uutta toimintatapaa vielä omakseen tunne. Jos verkostoitumistilaisuus yrityksen ulkopuolelle ei jostain syystä onnistu, niin vastaavasti eri yrityksistä voisi kutsua henkilöitä kohdeorganisaation vieraaksi viettämään esimerkiksi Agile-päivää. Näin saataisiin paljon arvokkaita tosielämän kokemuksia hyvistä käytänteistä jaettua. Syvällisempää Agile-näkemyksiä kohdeorganisaatiolle voisi tuoda myös se, että Scrum masterin rooli olisikin kiertävä eli käytännössä esimerkiksi scrum-tiimin sisällä Scrum masterina toimisi jokainen tiimin jäsen vuorollaan määrää-ajan, vaikkapa yhden kvartaalin kerrallaan. Näin Agile-toimintatavat ja -säännöt avautuisivat konkreettisemmin myös kohdeorganisaation muille henkilöille.

Kohdeorganisaation tämänhetkiset Scrum-tiimien kokoonpanot tuntuivat olevan toimivia juuri heidän Agile-ympäristöönsä. Scrum-tiimien kokojen suositellaan olevan 3-9 henkilöä, joten kohdeorganisaation Scrum-tiimien koot olivat 7-9 eli juuri olemassa olevien suositusten kokoisia. Jos kohdeorganisaation työntekijämäärä tulevaisuudessa kasvaa, heidän pitää miettiä Scrum-tiimien kokoonpanot uudelleen, jotta tiimit eivät kasva liian suuriksi.

Agile-ympäristössä työskentelylle on oleellista olla ajan hermoilla ja pitää työympäristönsä jatkuvassa kehityksessä. Koska koko kohdeorganisaatio on hyvin sitoutunut ja motivoitunut työntekoonsa, tällä organisaatiolla on hyvät edellytykset saada tulevaisuudessa kehitettyä Agile-ympäristöänsä vieläkin tehokkaammaksi. Organisaation kannattaisikin rohkeasti toteuttaa muun muassa tämän opinnäytetyön pohjalta tulleita erilaisia kehitysehdotuksia ja jatkossa myös suoraan työntekijöiltä tulevia kehitysehdotuksia. Kehittämiselle ja uudistamiselle on vaan annettava tarpeeksi aikaa ja prosessissa on edettävä pikku hiljaa vaiheittain niin, että saadaan kaikki työntekijät motivoitumaan ja sitoutumaan mukaan muutokseen.

Lähteet

Kirjalliset lähteet

Aaltola, J., Valli, R. 2007. Ikkunoita tutkimusmetodeihin I. 2. Painos. Juva: WS Bookwell Oy.

Davies, R., Sedley, L. 2009. Agile coaching. USA: The Pragmatic Bookshelf.

Hirsjärvi, S., Hurme, H. 2000. Tutkimushaastattelu. Helsinki: Yliopistopaino.

Leskinen, J. 1995. Kuluttajatutkimuskeskus. Laadullisen tutkimuksen risteysasemalla. Helsinki: Ykköspaino Oy.

Metsämuuronen, J. 2006. Laadullisen tutkimuksen käsikirja. Jyväskylä: Gummerus Kirjapaino Oy.

Smith, G., Sidky, A. 2009. Becoming Agile. New Delhi: Dreamtech Press.

Sähköiset lähteet

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. 2002. Agile Software development methods. Tulostettu 1.5.2012.
<http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>

Agile manifesto. Ketterän ohjelmistonkehityksen 12 periaatetta. 2001. Tulostettu 1.5.2012.
<http://agilemanifesto.org/iso/fi/principles.html>

Gregory, J. 2009. Focus Your Testing - Using the Agile Testing Matrix. Tulostettu 16.05.2012.
<http://janetgregory.ca/documents/Edmonton-Quadrants.pdf>

Kaner, C. 2006. Exploratory testing. Tulostettu 14.5.2012.
<http://www.kaner.com/pdfs/ETatQAI.pdf>

Reaktor. Ketterät menetelmät. Tulostettu 15.5.2012.
<http://reaktor.fi/osaaminen/agile/>

Schwaber, K., Sutherland, J. 2011. The Scrum guide. Tulostettu 3.5.2012.
<http://scrumwell.files.wordpress.com/2012/01/scrum-guide-2011-fi-v1.pdf>

Vuori, M. 2010. Ketterä testaus ja testaus ketterässä ohjelmistokehityksessä. Tulostettu 15.05.2012.
http://www.mattivuori.net/julkaisuuttelo/liitteet/kettera_testaus.pdf

Julkaisemattomat lähteet

ICT-alan yrityksen kohdeorganisaation materiaali. Tulostettu 30.04.2012.

ICT-alan yrityksen materiaali. Tulostettu 30.04.2012.

Kohdeorganisaation johto, ICT-alan yritys, haastattelu 1. 4.5.2012.

Kohdeorganisaation johto, ICT-alan yritys, haastattelu 2. 14.5.2012.

Kuviot

Kuvio 1: Extreme Programming (XP)-menetelmän elinkaaren havainnointikuva (Abrahamsson ym. 2007, 19).	9
Kuvio 2: Scrumin prosessikaavio (Reaktor-sivusto).	13
Kuvio 3: The Agile Testing Quadrants (Gregory 2009, 10).	16
Kuvio 4: Kohdeorganisaation käyttämän Agile -menetelmän prosessinkuvaus (Kohdeorganisaation materiaali 2012).	19
Kuvio 5: Esimerkki kohdeorganisaation käyttämästä julkaisukalenterista (Kohdeorganisaation materiaali 2012).	20
Kuvio 6: Kohdeorganisaation testausprosessin kuvaus sprintissä (Kohdeorganisaation materiaali 2012).	21
Kuvio 7: Kohdeorganisaation vanhan testauskäytännön kuvaus (ICT-alan yrityksen materiaali 2012).	21
Kuvio 8: Kohdeorganisaation nykyinen testausprosessi (ICT-alan yrityksen materiaali 2012).	22

Liitteet

Liite 1: Teemahaastattelulomake suomeksi

Teema 1: Agile käsitteenä ja työympäristönä

Mitä tiesit Agile ohjelmistonkehitysmenetelmästä ennen sen käyttöönottoa yksikössäsi?

Miten sinut perehdytettiin ja koulutettiin yksikkönne Agile-menetelmään? Oliko koulutus mielestäsi riittävää?

Pitäisikö mielestäsi järjestää säännöllisiä Agile-koulutuksia? Miksi ja minkälaisia?

Miten kuvailisit Agile-ympäristöä yleisellä tasolla ja sen toimivuutta omaa työnkuvaasi ajatellen?

Mitä mieltä olet yksikkönne projektiryhmien (Product team, Scrum team) kokoonpanoista ja kooista?

Miten kuvailisit Agile-ympäristöä ja sen toimivuutta alihankkijoiden kanssa tehdyn yhteistyön näkökulmasta?

Miten Agile-ympäristö on vaikuttanut alihankkijoiden kanssa tehtävään yhteistyöhön?

Miten mielestäsi käyttämäne Agile-menetelmä soveltuu yksikkönne eri

ohjelmistokehitysprojekteihin sekä prosessien kehittämisprojekteihin?

Kuvaile testaamisesta Agile-ympäristössä.

Kuvaile, mitkä ovat olleet suurimmat haasteesi Agile-ympäristössä työskennellessäsi?

Miten kuvailisit Agile-ympäristössä käytettäviä työkaluja, esimerkiksi Accept IT Toolia?

Onko työkalu mielestäsi oleellinen Agile-ympäristön toimivuutta ajatellen?

Teema 2: työmotivaatio, työssä jaksaminen ja työkuorma

Miten kuvailisit työmotivaatiosi tällä hetkellä? Koetko Agile-ympäristöön siirtymisen parantaneen vai huonontaneen työmotivaatiosi?

Minkälaiseksi kuvailisit työnkuvaasi tällä hetkellä? Koetko Agile-ympäristöön siirtymisen laajentaneen vai kaventaneen työnkuvaasi?

Koetko omat työtehtäväsi tarpeeksi monipuoliseksi tällä hetkellä? Perustele vastauksesi.

Minkälaisia konkreettisia muutoksia työtehtäviesi hoitamisessa sekä projektien läpiviemisessä tapahtunut Agile- ympäristöön siirtymisen jälkeen?

Miten selkeäksi koet oman roolisi ja vastuusi eri työtehtävissä tällä hetkellä? Koetko Agile-menetelmän selkeyttäneen oman roolisi ymmärtämistä tekemissäsi työtehtävissä/projekteissa?

Miten kuvailisit Agile-ympäristössä työskentelemisen tehokkuutta? Koetko oman työnteokosi olevan tehokasta? Perustele vastauksesi.

Miten kuvailisit työkuormaasi tällä hetkellä? Mitkä tekijät siihen vaikuttavat?

Miten Agile-menetelmän käyttö on mielestäsi vaikuttanut esimies-alais suhteeseen?
Haluatko työskennellä Agile-ympäristössä jatkossakin? Perustele vastauksesi.

Teema 3: parannus - ja kehitysehdotukset

Miten haluaisit yksikkönne tämänhetkistä Agile-ympäristöä kehitettävän?

Miten haluaisit yksikkönne projektiryhmien kokoonpanoja ja kokoja muutettavan/kehitettävän?

Miten haluaisit kehittää työtehtävien monipuolisuutta?

Miten haluaisit Accept IT toolia kehitettävän?

Miten Agile-ympäristöänne pitäisi mielestäsi kehittää testaamisen näkökulmasta?

Miten Agile-ympäristöänne pitäisi mielestäsi kehittää alihankkijoiden kanssa tehtävän yhteistyön näkökulmasta?

Onko sinulla yleisiä kehitysehdotuksia käyttämäänne Agile-ympäristöön?

Liite 2: Teemahaastattelulomake englanniksi

Theme 1: Agile as a concept and as a working environment

What did you know about the Agile software development methods before the method was taken into use in your organization?

How were you familiarized and trained to the Agile method of your organization? Was the training sufficient enough?

What is your opinion – is there need for regular Agile trainings? Describe why and what kind of trainings?

How would you describe the Agile working environment in general and it's functionally from the perspective of your daily work?

What is your opinion about the combination and the size of your organization's project teams (Product team, Scrum team)?

How would you describe the Agile working environment and it's functionally from the perspective of the co-operation with the subcontractors (externals)?

How the Agile environment has impacted on the co-operation with the subcontractors (externals)?

What is your opinion, how well does the Agile method fit for the different software development projects and process development projects of your organization?

Describe about the testing in the Agile environment.

Describe, which issues have been your biggest challenges when working in the Agile environment?

How would you describe the tools used in the Agile environment, such as Accept IT tool? What is your opinion, is the tool essential when thinking about the functionality of Agile environment?

Theme 2: work motivation, work well-being and work load

How would you describe your work motivation at the moment? How do you feel - has the Agile environment made you work motivation better or worse?

How would you describe your work load at the moment? How do you feel - has the Agile environment mad your job description wider or narrower?

How do you feel - is your job description versatile enough at the moment? Please give your reasons for your answer.

What kind of concrete changes has happened in your daily work or project follow-through after the Agile method was taken into use?

How clear are your own roles and responsibilities to you at the moment? Do you think that Agile method has clarified you to understand your own role better in your work / projects you are working with?

How would you describe the work effectiveness when working in the Agile environment? Do you think your own work is effective at the moment? Please give your reasons for your answer.

How would you describe your work load at the moment? Which elements are impacting on it? How do you feel the Agile method's usage has impacted the relation between the superior and the subordinate?

Would you like to work in the Agile environment also in future? Please give your reasons for your answer.

Theme 3: proposals for improvement and development

How would you like to develop further the present Agile environment of your organization?

How would you like the combination and the size of your organization's project teams (Product team, Scrum team) to be changed or developed further?

How would you like to develop further the versatility of the job descriptions of your organization?

How would you like to develop further Accept IT tool?

How would you like to develop further your Agile environment from the perspective of testing work tasks?

How would you like to develop further your Agile environment from the perspective of the co-operation with the subcontractors (externals)?

Do you have any other development suggestions for your Agile environment?