

Scrum palvelumuotoilun viitekehyksenä – soveltuvuuden tarkastelu

Otto Suomela



Sisällys

1	Johdanto	1
2	Palvelumuotoilu.....	3
2.1	Yleiset palvelumuotoilun menetelmät	4
2.2	Käyttäjäkokemus- ja käyttöliittymäsuunnittelu	7
3	Ohjelmistokehityksen synty ja sen ketteröityminen	9
3.1	Scrum	10
3.2	Yleiset ketterät kehittämisen- menetelmät.....	12
4	Tutkimuksen toteuttaminen	14
4.1	Tutkimusmenetelmä.....	14
4.2	Tutkimustehtävä, tutkimuskysymys ja tutkimuksen tavoite	15
4.3	Haastatteluiden toteutus.....	15
4.4	Aineiston analyysi	17
5	Tutkimustulokset	21
5.1	Palvelumuotoilutyön ero ohjelmistokehityksestä.....	21
5.2	Scrum- viitekehityksen soveltaminen ja sen tuomat hyödyt.....	24
5.3	Mahdolliset estävät syyt Scrumin- viitekehityksen toimivuuteen palvelumuotoilutyössä	27
6	Pohdinta.....	31
6.1	Tutkimuksen pätevyys ja luotettavuus.....	33
	Lähdeluettelo	34
	Liitteet.....	37

Tekijä tai tekijät Otto Suomela	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	Sivu- ja liitesivumäärä 36+1
Raportin nimi Scrum palvelumuotoilun viitekehysten – soveltuvuuden tarkastelu	
<p>Tiivistelmä</p> <p>Tutkimuksen tarkoituksena on selvittää, että soveltuuko Scrum- viitekehys palvelumuotoilutyöhön. Miten palvelumuotoilu eroaa perinteisestä ohjelmistokehityksestä ja mitä kokemuksia palvelumuotoilijat ovat kokeneet soveltaessaan Scrumia heidän työssään. Tutkimuksen tarkoituksena on selvittää palvelumuotoilijoiden mielteitä asiasta keskittyen työelämään. Johtaako Scrum tehokkaaseen työskentelyyn ja mitä muita hyötyjä se tarjoaa. Tutkimuksen avulla myös pystyttiin selvittämään minkälaisia heikkouksia Scrum-viitekehys on aiheuttanut.</p> <p>Tietoperustassa esittelen opinnäytetyöni kannalta oleelliset käsitteet sekä luon empiirille osalle teoriaperustan. Tietoperusosiossa määritellään palvelumuotoilun ja siihen oleellisesti kuuluvat mallit ja työkalut sekä esittelen Scrumin roolin ketterässä kehityksessä</p> <p>Opinnäytetyöni tutkimusmenetelmäksi valitsin kvalitatiivisen eli laadullisella tutkimusmenetelmän. Kun vastaajana on ihminen, kvalitatiivinen tutkimusmenetelmä on sopivin. Kvalitatiivisen tutkimuksen avulla yritetään hahmottaa ihmisten mielteitä, henkilökohtaisia kokemuksia sekä ihmisten maailmankäsitystä. Kysymysasetteluun, mitä, miksi ja miten vastaa laadullinen tutkimusmenetelmä. (Hirsjärvi, Remes & Sajavaara 2004, 151–155.)</p> <p>Valitsi tutkimukseni aineiston hankintamenetelmäksi teemahaastattelun ja haastattelut toteutettiin etänä COVID-tautitilanteen tilanteen vuoksi. Kvalitatiivisen tutkimuksen aineiston analyysimenetelmäksi valittiin aineistolähtöistä sisällönanalyysiä.</p>	
Asiasanat Palvelumuotoilu, Ketterä kehitys, Scrum- viitekehys	

1 Johdanto

Millä perusteella yritykset hakeutuvat ketterään kehitykseen? Onko kyseessä muoti-ilmiö vai voiko muutkin kuin ohjelmistokehittäjät nojautua ketterään kehitykseen työssään ja jopa hyötyä siitä?

Olen tammikuusta 2020 työskennellyt it-alalla osana palvelumuotoilutiimiä. Palvelumuotoilu on palvelujen innovointia, kehittämistä ja suunnittelua muotoilun keinoin, jossa käytetään ketteriä tutkimusmenetelmiä asiakasymmärryksen kartoittamiseksi. Vaikka palvelumuotoilu ei ole sama kuin ohjelmistokehitys ja vaikka niissä onkin eroja, niin voi niiden työskentelyperiaatteita tarkastella ristikkäin. It- palvelumme ohjelmistokehitystiimit ottivat Scrum- viitekehityksen käyttöön ketterään kehitykseen siirtyessä ja näin muiden mukana mukauduimme myös saman viitekehityksen alle. Mutta eräs kysymys on ilmentynyt jatkuvat kehitystyömme edetessä, kuinka tämä Scrum- viitekehitys soveltuu palvelumuotoilutyöhön? Näin opinnäytetyöni aihe syntyi ja tiesin mitä lähteä tutkimaan.

Tutkimuksessani pyrin selvittämään, kuinka hyvin Scrum- viitekehitys toimii palvelumuotoilutyössä. Onko tavanomaisessa ohjelmistokehityksessä eroavaisuuksia palvelumuotoilutyöhön ja minkälaisia palvelumuotoilijoiden kokemukset ovat olleet soveltaessaan Scrumia omassa työssään. Johtaako Scrum tehokkaaseen työskentelyyn ja tuoko se toiminnallista sekä taloudellista hyötyä?

Opinnäytetyössäni teoriapohjani käsittelee aluksi palvelumuotoilua, palvelumuotoilumenetelmiä sekä käyttäjäkokemus ja käyttöliittymäsuunnittelua. Luvussa 3 käsittelen ohjelmistokehityksen syntyä ja sen kehittymistä ketterään kehitykseen. Luvussa tärkein huomio kohdistuu Scrum- viitekehitykseen sivuten myös muita kehitettyjä ketteriä menetelmiä.

Keskeiset käsitteet opinnäytetyössäni ovat:

Palvelumuotoilu - Eng. Service Design. Palvelumuotoilun tavoite on keskittyä ymmärtämään syvällisesti palveluiden käyttäjiä sekä heidän motiivejaan ja tarpeitaan.

Design Sprint - Intensiivinen viiden päivän kestävä työpaja.

UI - Käyttöliittymä, englanniksi User Interface. Keskittyy tuotteiden visualisointiin.

UX - Käyttäjäkokemus englanniksi User Experience. Käyttäjäkokemuksen tarkoituksena on keskittyä ongelmien ratkomiseen ja tehdä tuotteista mahdollisimman helppokäyttöisiä sen käyttäjilleen.

Vesiputousmalli - Yksinkertainen ohjelmistoprojektien vaihemallina.

Ketterä kehitys - Tarkoituksena on tuoda ohjelmistokehittämisen prosesseihin iteratiivinen sekä inkrementaalinen lähestymistapa.

Scrum- viitekehys - Ketterä viitekehys. Kehitetty monimutkaisten tuotteiden kehittämiseen.

Scrum tuoteomistaja – Tuoteomistaja (eng. Product Owner) on henkilö, joka vastaa tuotteen sekä kehitystiimin työn arvon maksimoimisesta.

Scrum Master – Vastuuhenkilö. Scrum Masterin vastuulla on huolehtia kehitystiimin jäsenten ymmärryksestä Scrumista ja sen käytöstä. Scrum Master käyttää myös aikaansa järjestämälle Scrum- tiimin kehitysjonoa sekä järjestämällä erilaisia Scrum- tapaamisia.

Scrum kehitystiimi – Joukko kehittäjiä, joilla on valta päättää työnjaosta sekä tavoitteista.

Kanban- taulu - Taulu on luotu helpottamaan ohjelmistoprojektien hallintaa ja prosessien pysymistä aikataulussaan. Jaetaan kolmeen osaan, tehtävät työt (eng. To Do), työn alla olevat tehtävät (eng. Doing) sekä suoritettut tehtävät (eng. Done).

Scaled Agile Framework - Viitekehys on suunniteltu ohjelmistopainotteisten suurten organisaation tarpeisiin, mikä auttaa isoja yrityksiä hyödyntämään ketterän kehittämisen menetelmiä.

DevOps - Yksi jatkuvan kehittämisen viitekehysmalli, joka on rakennettu kahdeksasta seuraavasta periaatteesta: liiketoiminnan tarpeisiin keskittyminen, aikatauluissa pysyminen, yhteistyö, laadun takaaminen, rakenna vankka perusta kehittämiselle, kehitä iteratiivisesti, jatkuva selkeä kommunikointi sekä hallitsemisen osoittaminen.

2 Palvelumuotoilu

Lähdetään ensimmäisenä tarkastelemaan mitä palvelumuotoilu (eng. Service Design) oikein on? Palvelumuotoilun tavoite on keskittyä ymmärtämään syvällisesti palveluiden käyttäjiä sekä heidän motiivejaan ja tarpeitaan. Palvelumuotoilua hyödyntäen palveluita voidaan kehittää sellaisiksi, että asiakkailta on halu käyttää niitä ja ne olisivat samalla kannattavia liiketoiminnan näkökulmasta. Palvelumuotoiluun sopii kehittäjät, joilta löytyy empatiaa sekä kykyä asettua asiakkaan saappaisiin. Palvelumuotoilussa asiakkaiden kanssa kehitetään palveluita ja tekemistä pidetään iteratiivisena sekä kokeilevana. Käyttäjakeskeisen lähestymistavan palveluiden kehittämisessä luovat muotoilun prosessit ja menetelmät. Liiketoimintanäkökulmasta kehittämisen ytimessä on itse asiakasymmärrys. (Tuulaniemi 2011, 63,103.)

Itse palvelumuotoilu terminä on tullut tunnetuksi viime vuosina ja nykypäivänä tuskin voidaan puhua palveluista ilman, että palvelumuotoilu termi nousisi esille. Palveluiden suunnittelussa palvelumuotoilu on noussut hiljalleen keskeiseksi käsitteeksi, mutta onko kyseessä vain ohimenevä trendi? Palvelumuotoilu otetaan yleisesti avoimin mielin vastaan, koska sen koetaan olevan merkittävä asia nykyajan yhteiskunnassa sekä yritysten tarpeissa. Palvelumuotoilu sopii vaivatta sellaiseen työkuultuuriin, missä palveluiden suunnitteluprosessiin osallistuvat asiakkaat sekä työntekijät. (Koivisto, Säynäjäkangas & Forsberg 2019, 33.)

Palveluketjun keskiössä on ihminen. Palvelumuotoilussa tämä tarkoittaa asiakkaan palvelukokemusta. Kokemuksen optimointiin palvelumuotoilussa yritetään keskittyä kokemuksen tärkeimpiin pisteisiin mm. palveluprosesseihin, tiloihin, työtappoihin ja vuorovaikutukseen sekä parantamalla asiakkaan kokemusta poistamalla häiritseviä tekijöitä. (Tuulaniemi 2011, 26–27.)

Palvelumuotoilun kuudeksi keskeiseksi periaatteeksi Stickdorn (2018) on määritellyt;

- 1) Ihmiskeskeistä eli käyttäjälähtöisyyttä. Palveluita suunnitellessa pitäisi ottaa huomioon aina palvelun käyttäjän näkökulma.
- 2) Yhteistyössä tekeminen. Kaikkien osapuolten osallistaminen palvelumuotoilun prosessiin. Sidosryhmien mukana oleminen palvelumuotoiluprojektissa tulisi olla aktiivista.
- 3) Toistuva kehittäminen. Palvelumuotoilussa jatkuvan kehittämisen myötä päästään lopputavoitteisiin. Palvelumuotoilussa lähestymistapoja pidetään tutkivana, mukautuvana ja kokeilevana.

- 4) Havainnollistavaa. Palvelukokemuksen elinkaarta voidaan havainnollistaa visuaalisesti alkupisteestä päätökseen palvelupolun avulla.
- 5) Todellisuus. Käyttäjien tahto tulisi löytää todellisesta toiminnasta ja prototyyppien suunnittelu tulisi toteuttaa mahdollisimman realistisesti.
- 6) Kokonaisvaltaisuus. Palvelumuotoilussa palveluympäristön kokonaisuus on huomioitava tekemisessä. (Stickdorn, Hormess, Lawrence, Schneider 2018, 25–27.)

Digitalisaation myötä palvelumuotoilun tarve on yhä lisääntynyt. Sen jatkuvan läsnäolon kautta kasvotusten tapahtuvat asiakastapaamiset ovat vain vähentyneet ja asiakkaat ovat siirtymässä verkko- ja mobiilikäyttäjiksi. Palvelumuotoilulla voidaan tarjota työkaluja inhimillistämään teknologioita, jotta palvelukokemusta voidaan parantaa. (Reason, Lovlie, Flu 2016, 4.)

2.1 Yleiset palvelumuotoilun menetelmät

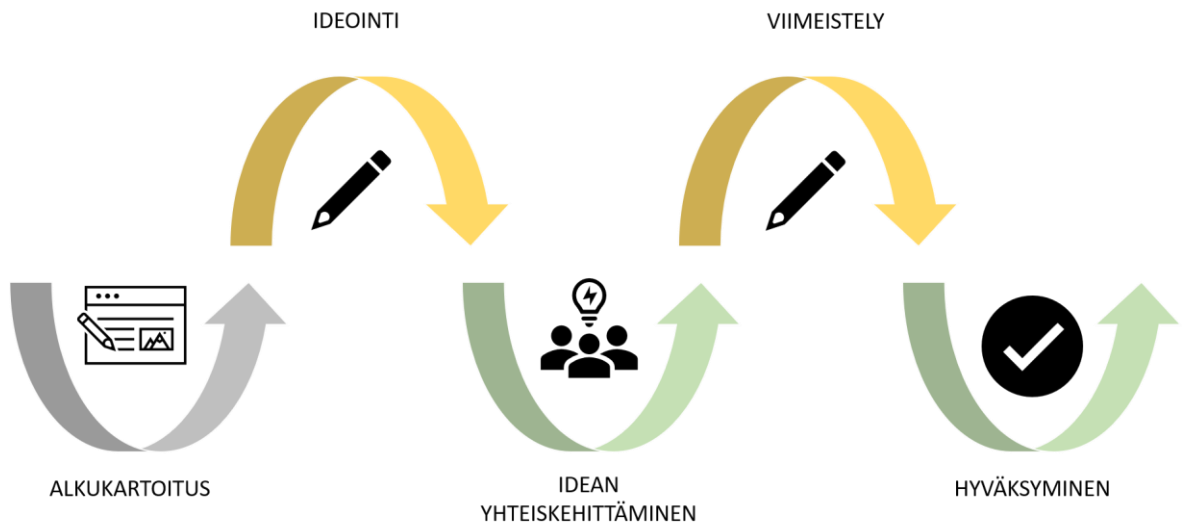
Visuaalista viestintää hyödynnetään palvelumuotoilussa hyvinkin paljon ja hyvä keino tuoda projektiin liittyviä oleellisia käsitteitä ja aiheita esille on luoda käsitekartta eli Mind map. Käsitekartalla autetaan hahmottamaan projektien kokonaiskuvaa sekä tuomaan esille asioita mitkä liittyvät ja ovat vuorovaikutussuhteessa toisiinsa. (Tuulaniemi 211, 140.)

Asiakasymmärryksen kartoittamista lisätään keskittymällä olemassa olevia taustatietoja sekä haastatteluiden tutkimiseen. Kun käyttäjistä saadaan kerättyä tarpeeksi tietoa, voidaan tiedon avulla rakentaa erilaisia käyttäjäpersoonia, käyttäjätarinoita ja palvelupolkuja palvelukokonaisuuden kuvaamista varten. (kuvat 1 ja 2)



Kuva 1. Käyttäjäpersoonia, joka edustaa tiettyä asiakaspiiriä

Kerättyjen tietojen pohjalta tehdään kuvitteellisia käyttäjäprofiileja, tarinoita tai palvelupolkuja. Nämä kuvitteelliset keinot ovat tärkeitä yrityksille, jotka pyrkivät tuottamaan parempaa asiakaskokemusta ja ymmärtää asiakkaidensa arvomaailmaa ja käyttäytymistä. (Tuulaniemi 2011, 154-155.)

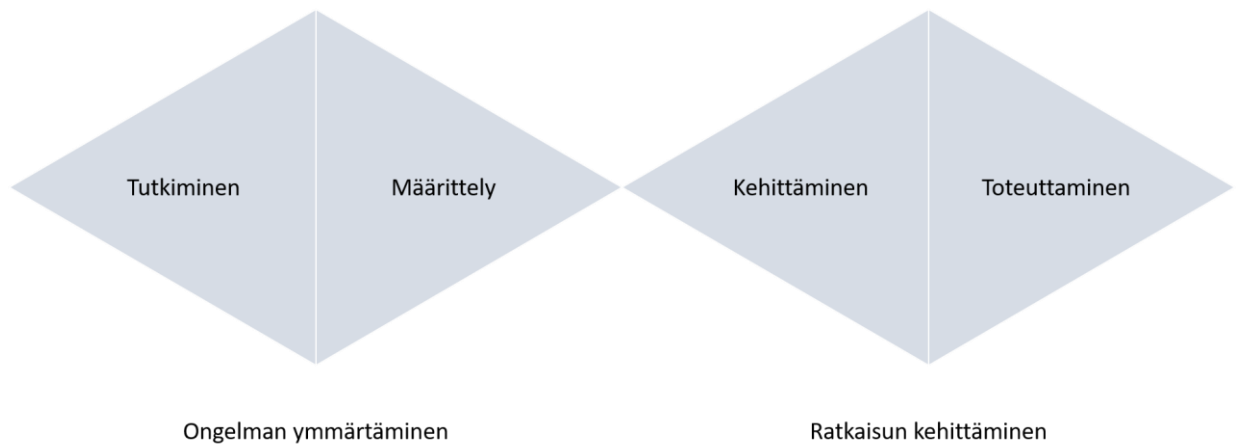


Kuva 2. Palvelupolku. (Mukaillen palvelub.blogspot.com)

Erialaisten profiilien, tarinoiden tai polkujen avulla voidaan hahmottaa käyttäjien tarpeita, jonka avulla voidaan määrittää projekteille suunnitteluajureita. Suunnitteluajurit määritetään siis projektille, kun asiakasymmärryksen kartoittamisen avulla löydetään jokin käyttäjän olemassa oleva tarve. Nämä suunnitteluajurit ajavat suunnittelutyötä. Kun esimerkiksi asiakasymmärrystä tutkimalla nousee esille, että käyttäjät kokevat uuden järjestelmän käytössä ongelmia, suunnitteluajurit voivat siinä tapauksessa johdattaa perehdytystilaisuuksien järjestämistä palveluun tutustumista varten. (Tuulaniemi 2011, 156–157.)

Tämän jälkeen voidaan aloittaa ideoiminen. Ideoinnin avulla voidaan rakentaa pohjaa tuleville palvelukonsepteille. Ideointi voi helposti laajentua turhankin isoksi ja tämän vuoksi on lähdettävä supistamaan ideointia luopumalla tai yhdistelemällä ideoita toteutettavissa oleviksi palveluaihioiksi. (Tuulaniemi 2011, 180.) Tätä tapaa kutsutaan Tuplatimantti-menetelmäksi (eng. Double Diamond). Tämän Tuplatimantin prosessimalli muodostuu neljästä eri vaiheesta. Ensimmäisenä prosessin vaiheena on tutkiminen (discover), toisena määrittely (define), kolmantena kehittäminen (develop) ja neljäntenä toteuttaminen (deliver). Englantilainen Design Council on kehittänyt Tuplatimantti prosessimallin. Sen tarkoituksena on kuvitteellisesti yksinkertaistaa palvelumuotoiluprosessia. Prosessimallin kuvasta

hahmottuu kaksi peräkkäistä timanttia, joihin sisälle on suomennettu edellä mainitut neljä vaihetta (Kuva 3). Ensimmäisellä timantilla pyritään kartoittamaan ratkaistavaa ongelmaa ja toisella timantilla pyritään kehittää ratkaisuja kyseiselle ongelmalle. (Design Council, 2019.)



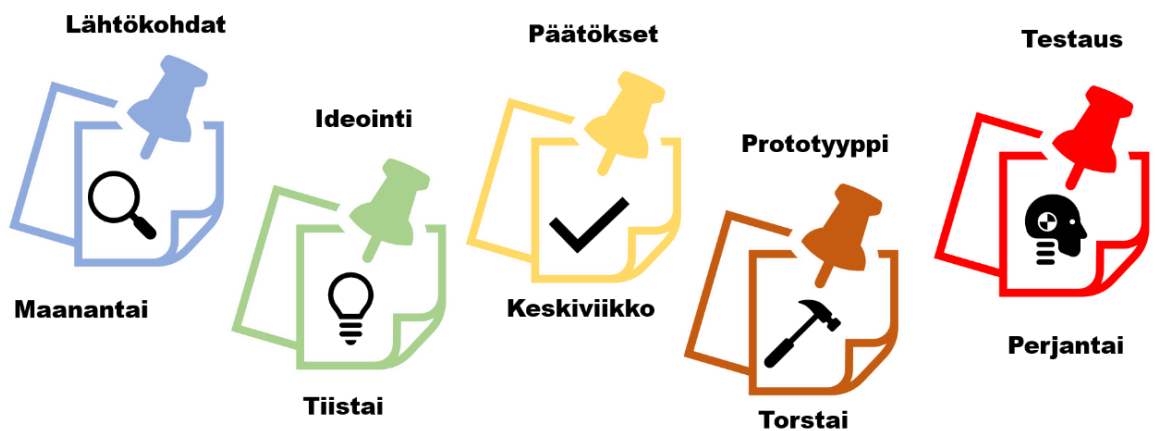
Kuva 3. Double Diamond - Tuplatimantti prosessimalli (mukaillen Design Council, 2019)

Ryhmätyöskentelypainotteisten työpajojen (eng. Workshop) tavoitteena on saada aikaan paljon innostavia ideoita. Työpajoissa keskitytään luomaan mahdollisimman paljon tietoa ja ideoita suhteellisen lyhyessä ajassa. Niitä voidaan hyödyntää asiakasymmärryksessä kartoittamalla esimerkiksi eri käyttäjäpersoonia. (Tuulaniemi 2011, 189.)

Intensiivistä viiden päivän kestävästä työpajasta kutsutaan Design Sprintiksi. Tämän intensiivisen työpajan tarkoituksena on auttaa tilanteissa, jossa palvelun testaus halutaan toteuttaa nopeasti ennen kuin pidempit kestoiseen kehitystyöhön investoidaan suuria summia resursseja. Tämä menetelmä auttaa tilanteisiin, joissa koetaan projektien olevan jumissa tai ajallisesti käytettävät resurssit ovat vähissä. Design Sprint on kehitetty selkeyttämään yhteisiä visioita sekä auttamaan projektien etenemistä. (Knapp, Zeratsky & Kowitz 2016, 26.)

Design Sprintin tehokkuus määrittyy sen selkeiden ja valvottujen työvaiheiden avulla, jotka muodostavat sprintin jokaiselle työpäivälle omat päätavoitteet. Sprintin päivien päätavoitteet ovat yleisesti jaotettu maanantaista perjantaihin (kuva 4). Maanantain tavoitteena on jakaa sprintin osallistujien tietämystä sekä määrittellä sprintin kohde. Tiistaina sprintin osallistujat luovat ratkaisuehdotuksia ja keskiviikkona käydään nämä ratkaisut läpi, joista äänestetään parhaimmat ideat. Ideoista luodaan keskiviikkona tarkka toteutussuunnitelma, jonka avulla torstaina voidaan rakentaa toimiva prototyyppi. Perjantai on design sprintin

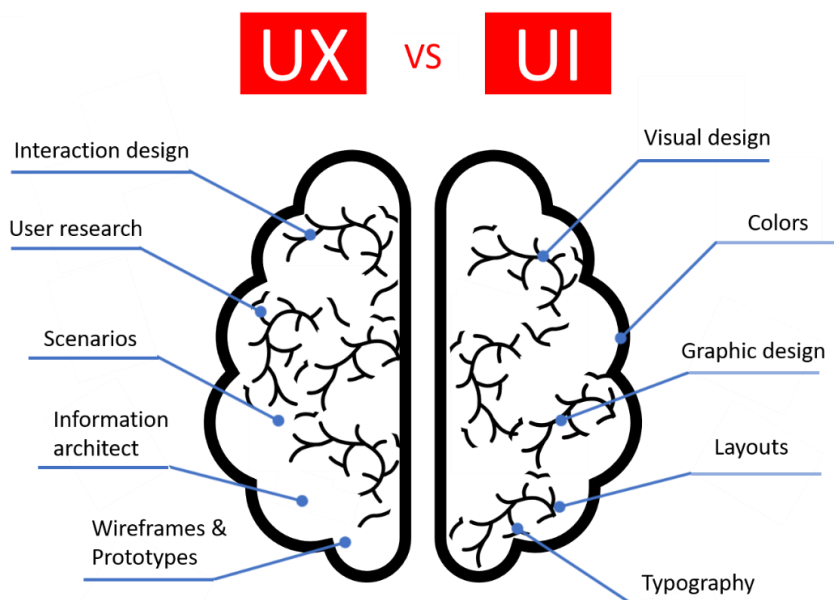
viimeinen päivä, mikä pyhitetään prototyypin testaukseen asiakkaiden avulla. Testauksessa keskitytään siihen, kuinka hyvin tuote palvelee asiakkaiden odotuksia ja mitä jatkokehityksiä ilmenee tulevaisuutta kohden. (Knapp, Zeratsky, Kowitz 2016, luku 3.)



Kuva 4. Design Sprint, Sprintin päivien päätavoitteet (mukaillen Knapp ym. 2016, 24)

2.2 Käyttäjäkokemus- ja käyttöliittymäsuunnittelu

Käyttöliittymäsuunnittelun ja käyttäjäkokemussuunnittelun erottaminen saattaa olla melko vaikeata, koska niiden välinen raja voi olla melkoisen pieni. Molemmilla on oma tarkoituksensa suunnittelutyössä, mutta voimme helpottaa niiden ymmärtämistä yksinkertaisella tavalla (Kuva 5). Käyttöliittymäsuunnittelussa ydin on tuotteen visualisoinnissa, kun taas käyttäjäkokemussuunnittelussa keskitytään tuotteen toiminnallisuuden suunnitteluun käyttäjän ja tuotteen välillä. Vaikka molempien töiden tarkoitusperät ovat toisistaan poikkeavia, eivät he tule toimeen ilman toistensa tukea. Siksi voimme nähdä useasti yhden henkilön hoitavan molempien suunnittelutöiden tekemistä. (Forsey, 2018.)



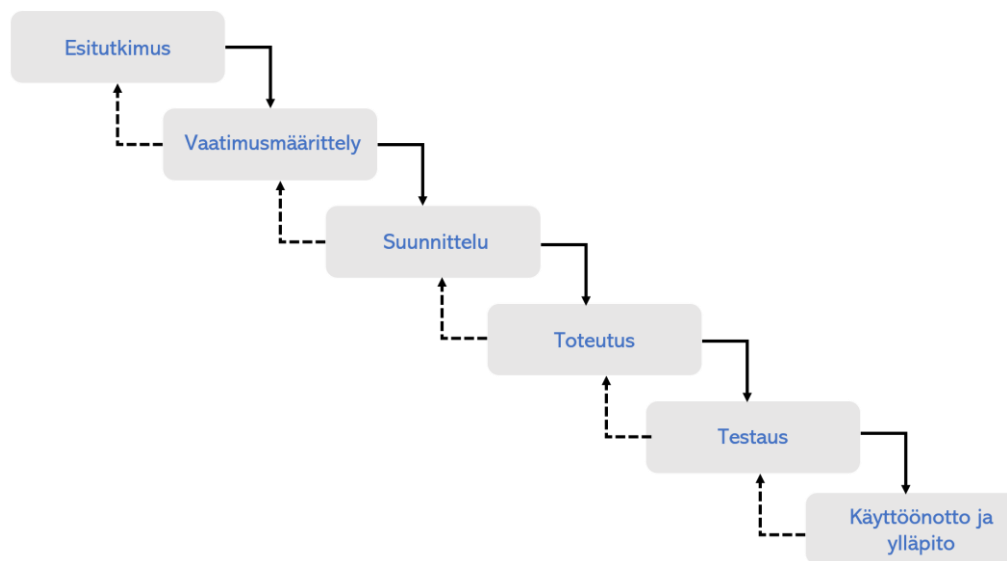
Kuva 5. UX vs UI design (mukaillen uxdesign.cc)

Käyttöliittymä, englanniksi User Interface (UI) keskittyy tuotteiden visualisointiin, mikä on käyttäjälle selvästi tuotteen näkyvin asia. Käyttöliittymäsuunnittelija keskittyy tuotteen väri-maailmaan, millä voidaan vaikuttaa paljon tuotteen miellyttävyyteen sillä jokainen väri vaikuttaa ja viestii omalla tavallaan asiakkaan silmiin. Suunnittelija keskittyy myös työssään typografiaan sekä elementtien asetteluun ja brändäykseen. Suunnittelutyössä elementtien asettelu on tärkeässä roolissa, sillä se vaikuttaa tuotteen selkeyteen. Jos elementtien asettelua ei suunnitella tarkkaan suunnittelutyön alusta pitäen, voi olla vaarana, että tuotteesta tulee sekava asiakkaan silmissä. (Virtanen, 2016.)

Käyttäjäkokemus tunnetaan englanninkielisestä sanasta User Experience (UX). Käyttäjäkokemuksen tarkoituksena on keskittyä ongelmien ratkomiseen ja tehdä tuotteista mahdollisimman helppokäyttöisiä sen käyttäjilleen. Suunnittelutyössä käyttöliittymäsuunnittelija keskittyy käyttäjän tunteiden herättämiseen, jotta suunnittelija pystyy tällöin ratkomaan käyttäjän osoittamia ongelmia. Tuotteen käyttökokemus on onnistunut, jos käyttäjästä herätetyt tunteet ovat positiivisia. Jos käyttäjästä ilmenee turhautumisen ja pettymyksen tunteita, voidaan todeta, että käyttökokemus on suunniteltu huonosti. Suunnittelijoiden tulee muistaa, että käyttäjä on suunnittelutyön keskiössä ja suunnittelutyö lähtee aina liikkeelle käyttäjästä ja hänen tarpeistaan. Käyttäjäkokemussuunnittelija tukeutuu työssään esimerkiksi erilaisiin käyttäjäkokemustutkimuksiin, minkä avulla hän voi luoda tuotteen käyttäjistä käyttäjäprofileja. Tutkimustyön avulla suunnittelija pystyy toteuttamaan tuotteelle uusia ratkaisuja hyödyntäen taitojaan luomalla erilaisia prototyyppejä ja rautalankamalleja. (Virtanen, 2016.)

3 Ohjelmistokehityksen synty ja sen ketteröittyminen

1970-luvulla Winston Royce esitteli julkaisemassaan artikkelissa prosessimallin, jonka kautta hän tuli luoneeksi niin sanotun vesiputousmallin (Waterfall) (Kuva 6). Tätä prosessimallia käyttävät yhä edelleen ohjelmistokehittäjät, jota pidetään ohjelmistoprojektien yksinkertaisimpana vaihemallina. Riippuen lähteestä vesiputousmalli koostuu viidestä seitsemään peräkkäisestä vaiheesta. Tiettyjä vaiheita voidaan yhdistää yksinkertaisiksi kokonaisuuksiksi tai vaiheita voidaan nimittää eri tavalla, mutta vaiheistuksen periaate pysyy aina samana. Vesiputousmallissa painopiste kohdistuu tarkkaan dokumentointiin ja näin ollen vaiheiden suunnittelu sekä dokumentointi hoidetaan tarkkuudella. (Ahonen 2010, 4–5.)



Kuva 6. Vesiputousmalli (mukaillen Ahonen 2010, 4)

Vesiputousmalli soveltuu parhaiten käytettäviksi projekteissa, joissa projektien vaatimukset eivät tule muuttumaan, vaan vaatimuksia voidaan pitää täysin selkeinä. Asiakasvaatimusten eläminen ja jatkuva muutos vaikutti uusien mallien kehittämiseen, jotka voivat lopulta korvata vanhentuneen vesiputousmallin (Ahonen 2010, 5). Tämän johti erilaisten ketterien menetelmien syntyyn.

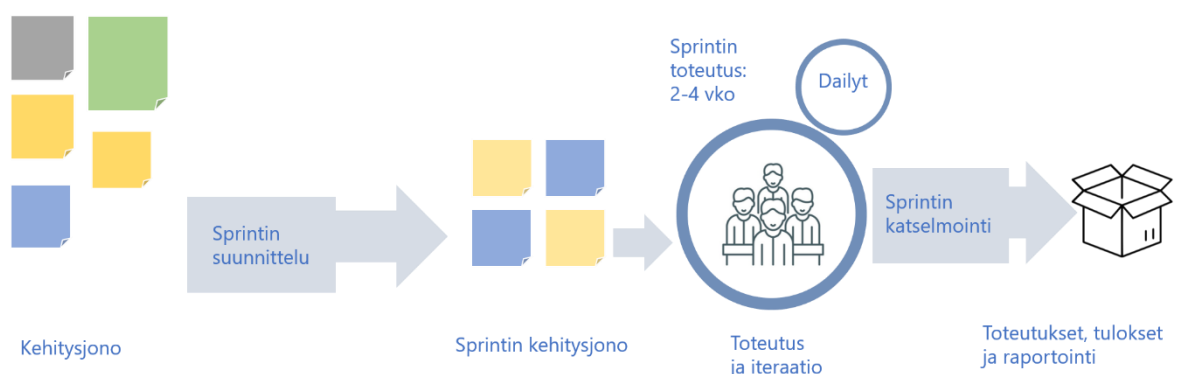
Bhavsar, Shah ja Gopalan (2020) tutkimuksessa kerrotaan ketterien menetelmien syntyneen 90-luvun loppupuolella. Ketterien menetelmien tarkoituksena on tuoda ohjelmistokehittämisen prosesseihin iteratiivinen sekä inkrementaalinen lähestymistapa. Ketterien menetelmien periaatteet määrittää julkaisu nimeltä *Agile Manifesto*. Julkaisun tehtävänä oli uudistaa ajattelutapaa kehittämisen perspektiivistä ja uusi ajattelutapa mahdollistaa ohjelmistokehittäjille suurempia rooleja. *Agile Manifesto*- julkaisu johti Scrum- viitekehyksen syntyyn. (Bhavsar, Shah & Gopalan 2020, 841.)

3.1 Scrum

Schwaber & Sutherland (2017, 3) määrittävät Scrum- viitekehyksen jäsenet voivat puuttua monimutkaisiin ja joustaviin ongelmiin toimittaen yhtäaikaaisesti tuottavia ja luovasti korkeimpia mahdollisia arvoja. Scrum- viitekehys koostuu tiimeistä, jotka sisältävät erilaisia rooleja, tapahtumia ja sääntöjä. Nämä osat ovat välttämättömiä Scrum- viitekehyksen toimimiseen sekä menestymiseen.

Scrum-viitekehyksen tiimi koostuu kolmesta roolista, Tuoteomistaja (Product Owner), Scrum Master ja kehitystiimi (Development team). Kehitystiimit ovat itseorganisoituvia ja he ovat tärkeimmässä asemassa Scrum- viitekehyksessä. Kehitystiimien koko on suhteellisen pieni. Pienen koon avulla tiimi pystyy toimia ketterästi ilman koordinoinnin kärsimistä sekä kooltaan tiimi on tarpeeksi suuri tuottamaan määrällisesti merkittävästi tuotoksia. Tiimin jäsenet ovat tasa-arvoisia riippumatta toistensa osaamisistaan. Kehitystiimin malli on suunniteltu tehostamaan joustavuutta, luovuutta ja tuottavaa työtä. (Schwaber & Sutherland 2017, 6.)

Scrumissa Tuoteomistaja on yksittäinen henkilö, jonka vastuulla on ajaa tuotteen ja kehitystiimin työn arvoa. Hänen vastuullensa kuuluu myös tuotteen kehitysjono (Product Backlog), joka koostuu erilaisista ensisijalle asetetuista tärkeistä tehtävistä sekä vaatimuksista lopputuotteen saavuttamiseksi. Kehitystiimi työskentelee kehitysjonon tehtäviä eli taskeja sovitussa ajanjaksoissa, joita kutsutaan Scrumissa sprinteiksi. Sprintin ajanjaksot vaihtelevat pituudeltaan kahdesta neljään viikkoon. (Lindström 2010). (Kuva 7)



Kuva 7. Scrum- viitekehys (mukaillen scrum.org)

Sprintti on kehitystiimin aikaraja, jonka aikana tiimi suorittaa tehtävälialta valittuja tehtäviä. Scrum- tiimi tapaa sprintin jokaisena päivänä yhteisessä Scrum- tapaamisessa eli Dailyssä. Näissä tapaamisissa kehitystiimin jäsenet pääsevät päivittäin jakamaan toisilleen

tärkeitä huomioita liittyen omaan työskentelyyn, kuten mahdollisista haasteista, esteistä tai tehdyistä töistä. Näiden tapaamisten avulla Scrum- tiimi kasvattaa läpinäkyvyyttä kehitystiimin jäsenten välillä sekä pystyvät auttamaan toisiaan mahdollisten ongelmien ennaltaehkäisemisessä. Sprintti päätetään järjestämällä Scrum- tiimin sisäinen retrospektiivi (Retrospective). Retrospektiivissä arvioidaan menneen sprintin onnistumiset ja kehityskohteet tulevaa alkavaa sprinttiä varten. Retrospektiivi tavoitteena on auttaa kehittämään tiimin toimintatapoja ja organisaatiota tunnistaa heidän potentiaaliset kykynsä (Lindström 2010.)

Scrum Master on vastuuhenkilö ja hän on yksi tärkeimmistä henkilöistä kehitystiimissä. Hänen vastuullensa kuuluu reagoida ilmeneviin esteisiin liittyen Scrum- viitekehykseen. Scrum Masterin vastuulla on huolehtia kehitystiimin jäsenten ymmärryksestä Scrumista ja sen käytöstä. Tuoteomistajalle Scrum Master tarjoaa tukeansa muun muassa varmistamalla, että sprintin tavoitteet, laajuus sekä tuotealue on ymmärretty Scrum- tiimissä. Scrum Master käyttää myös aikaansa järjestämälle Scrum- tiimin kehitysjonoa sekä järjestämällä erilaisia Scrum- tapaamisia. Scrum Master johtaa sekä valmentaa organisaatiota sillä Scrumin käyttöönottoa ja auttaa organisaatiota ymmärtämään prosessin käytäntöjä. (Schwaber & Sutherland 2017, 8.)

Scrum- viitekehysten suuri suosio on saavutettu sen yksinkertaisuuden, kattavien käyttötapauskaavioiden ja helpon käyttöönoton vuoksi. Kehittäjätiimin tuottavuutta parantavat Scrum- viitekehys mahdollistamat toiminnot sekä kannustimet (Johnson 2015).

Scrum- viitekehys poikkeaa paneutumalla tiimityöskentelyyn ja yhteistyön merkittävyyteen verrattuna muihin ketteriin menetelmiin. (Alliance 2020). Hierarkiaa saadaan vähennettyä Scrum- viitekehuksesta antamalla vastuuta sekä päätösvaltaa pienille tiimeille jo ennalta määritettyjen mallien sijasta. Nopeasti edistyvässä digitalisaatiossa organisaation henkilöiden tulisi olla mukautuvia, kehittyviä ja kestäväisempi. Tähän voidaan vaikuttaa levittämällä ketterää ajattelutapaa organisaation henkilöille. (Abraham 2019.)

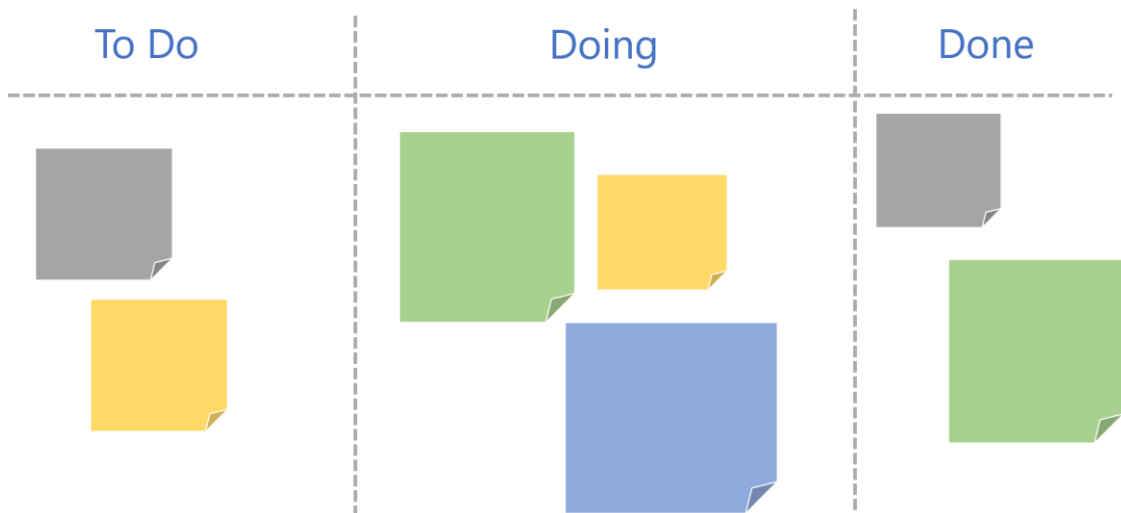
Ketterien menetelmien käyttöönotossa kommunikaatio on välttämätöntä. Ketterissä menetelmissä liiketoiminnan, teknologian sekä kehittäjätiimin välille vaaditaan selkeää ja järjestelmällistä kommunikaatiota. Tämä luo kaikkien osapuolten välille tarpeellisen sitoutumisen, ymmärryksen ja yhteistyön. (CGI 2016, 4.)

Ohjelmistokehittäjien toimintatavoissa Scrum- viitekehys tarvitsee asianmukaisen käyttöönoton, jotta voidaan parantaa kehitettävän palvelun tai tuotteen tuottavuutta. Scrum- tiimin sisäinen avoimuus, yhteistyö ja läpinäkyvyys vaikuttaa suoranaisesti tuotteen tuottavuuteen, laatuun sekä sen ominaisuuksiin. (Bhavsar, Shah & Gopalan 2020, 845.)

3.2 Yleiset ketterät kehittämisen- menetelmät

Kanban-malli

Kanban metodin kehitti David J. Andersson vuonna 2003, joka piti ohjelmistotuotteiden valmistumista ajallaan tärkeänä. Tämän myötä syntyi Kanban-taulu (Kuva 8). (Ashmore ym. 2015, 60–61.)



Kuva 8. Kanban-taulu. (mukaillen Atlassian Agile Coach)

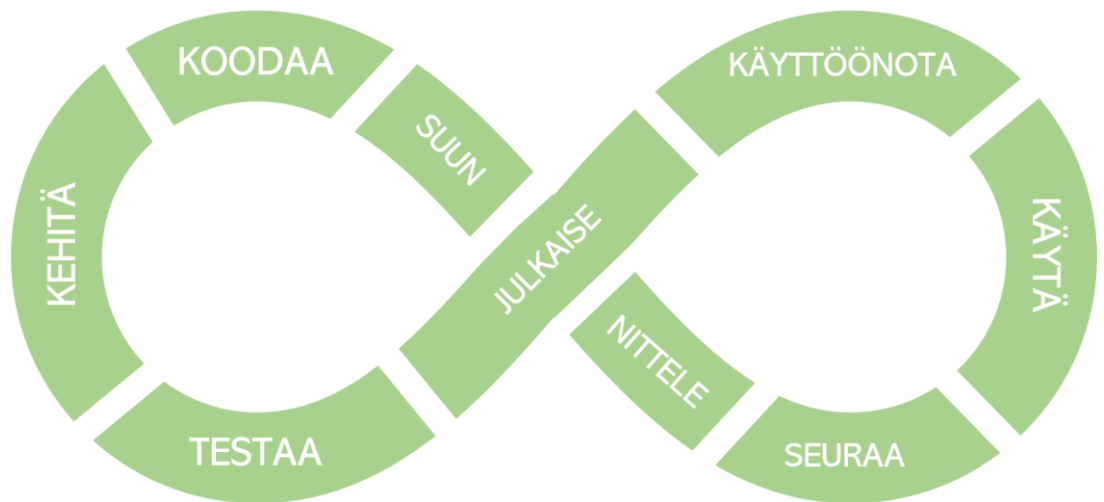
Kanban-taulu on yksinkertaisimmillaan seinälle sijoitettu fyysinen taulu, mikä on jaettu kolmeen osaan, tehtävät työt (eng. To Do), työn alla olevat tehtävät (eng. Doing) sekä suoritettut tehtävät (eng. Done). Taulu on luotu helpottamaan ohjelmistoprojektien hallintaa ja prosessien pysymistä aikataulussaan. (Ashmore ym. 2015, 60–61.)

Scaled Agile Framework

Yksi jatkuvaan kehitykseen suunniteltu skaalautuva viitekehysmalli on Scaled Agile Framework eli SAFe. Tämä viitekehys on suunniteltu ohjelmistopainotteisten suurten organisaation tarpeisiin, mikä auttaa isoja yrityksiä hyödyntämään ketterän kehittämisen menetelmiä. Tarkoituksena on pitää samalla kiinni jonkinlaisista organisaatorakenteista ja prosessien kontrolloinnista. Suurissa ohjelmistopainotteisissa projekteissa tiimit toimivat hyvin itsenäisesti. (Business Technology Standard. 2020, 5.3.)

DevOps- kehitysmenetelmä

DevOps- kehitysmenetelmä (kuva 9) on kehitetty yhdistämään kehittämisen (eng. development ja tuotannon (eng. operations) toisiinsa vuorovaikutuksen, integraatioiden ja yhteistyön kautta. Tämä malli sopii hyvin jatkuvan kehityskulun ajattelumalliin ja menetelmiin. Se mahdollistaa koodin käyttämisen tuotannossa nopeammin ja automaattisesti. DevOpsin avulla pystytään parantamaan organisaation nopeutta toimittaa ratkaisuja sekä palveluita. (Business Technology Standard. 2020, 5.3.)



Kuva 9. DevOps- kehitysmenetelmä (mukaiillen btmalli.fi, 5.3)

Dynamic Systems Development Method

Dynamic Systems Development Method (DSDM) on yksi jatkuvan kehittämisen viitekehysmalli, joka on rakennettu kahdeksasta seuraavasta periaatteesta: liiketoiminnan tarpeisiin keskittyminen, aikatauluissa pysyminen, yhteistyö, laadun takaaminen, rakenna vankka perusta kehittämiselle, kehitä iteratiivisesti, jatkuva selkeä kommunikointi sekä hallitsemisen osoittaminen. Tarkoituksenmukaisuus ja liiketoimintakartoitus lasketaan yhdeksi vaiheeksi menetelmän tarjoavassa nelivaiheisessa viitekehysessä. Kolme muuta vaihetta ovat toiminnallisten mallien ja prototyyppien testaus sekä suunnittelun ja versio toistoista ja käyttöönotoista.

4 Tutkimuksen toteuttaminen

Tässä kappaleessa esittelen työni empiirisen osan, mikä sisältää tutkimuksen tavoitteet, tutkimusmenetelmät, tutkimuksen toteutuksen, aineiston analyysin ja tulokset. Aloitan kertomalla tutkimukseen valitusta tutkimusmenetelmästä.

Tämän kappaleen jälkeen esittelen tutkimustehtäväni ja tutkimuskysymykseni, jonka jälkeen hahmotan tutkimuksen tavoitteet. Tutkimuskysymykseni haluaa löytää vastauksen siihen, että voidaanko Scrum- viitekehystä soveltaa palvelumuotoilussa? Opinnäytetyönsäni vertaan Scrum-viitekehyyksen soveltumista palvelumuotoiluntyöhön.

Tutkimuksen tavoitteiden jälkeen siirrytään haastattelun toteutuksen kerrontaan ja miten aineisto analysoitiin. Valitsi tutkimukseni aineiston hankintamenetelmäksi teemahaastattelun ja haastattelut toteutettiin etänä COVID-tautitilanteen tilanteen vuoksi. Kvalitatiivisen tutkimuksen aineiston analyysimenetelmäksi valittiin aineistolähtöistä sisällönanalyysiä.

4.1 Tutkimusmenetelmä

Opinnäytetyöni tutkimusmenetelmäksi valitsin kvalitatiivisen eli laadullisella tutkimusmenetelmän. Kun vastaajana on ihminen, kvalitatiivinen tutkimusmenetelmä on sopivin. Kvalitatiivisen tutkimuksen avulla yritetään hahmottaa ihmisten mielteitä, henkilökohtaisia kokemuksia sekä ihmisten maailmankäsitystä. Kysymysasetteluun, mitä, miksi ja miten vastaa laadullinen tutkimusmenetelmä. (Hirsjärvi, Remes & Sajavaara 2004, 151–155.) Näistä yllä mainituista syistä valitsin laadullisen tutkimusmenetelmän, sillä koin että sen avulla käsittelemään valitsemaani aihetta.

Kvalitatiivinen tutkimus suuntautuu kokonaisvaltaiseen tiedon hankintaan. Siinä tavoitteena on pyrkiä löytämään yhtäläisyyksiä kerätystä materiaalista. Kvalitatiivisen tutkimuksen tavoitteena on helpottaa tutkimuskohteen ymmärtämistä, käyttäytymistä ja päätöksenteon syitä. Haastattelut ovat yksi kvalitatiivisen tutkimuksen aineistonkeruumenetelmistä. (Hirsjärvi, Remes & Sajavaara 2004, 151–155.)

Laadullisessa tutkimuksessa tutkimuksen strategia on strukturoimaton ja tutkimuksen teorialat ovat sovellettavissa. Tutkimuksen joustavuutensa vuoksi haastattelut sopivat hyvin erilaisiin tutkimuksiin. Haastattelutilanteissa ollaan suorassa kontaktissa haastateltavan välillä ja tämän vuoksi on tyypillistä, että tutkimuksen tutkijan ja tutkittavan suhde on läheinen. (Hirsjärvi, Remes & Sajavaara 2004, 124.)

4.2 Tutkimustehtävä, tutkimuskysymys ja tutkimuksen tavoite

Tutkimuskysymykseni on, voidaanko Scrum- viitekehystä soveltaa palvelumuotoilussa? Opinnäytetyössäni vertaan Scrum-viitekehityksen soveltumista palvelumuotoilutyöhön. Tutkimustehtäväni on hahmottaa palvelumuotoilutyön ja ohjelmistokehityksen yhtäläisyyksiä sekä eroavaisuuksia sekä analysoida palvelumuotoilijoiden kokemuksia Scrum-viitekehityksestä.

Ketterään kehitykseen siirtyminen on ollut suuressa kasvussa kehityshankkeissa ja taakse on jätetty vanhentuneet kehitysmallit kuten ns. vesiputousmalli, jonka juuret juontuvat 1970-luvulle saakka. Nykyään monet alat haluavat siirtyä ketteriin menetelmiin, mutta epävarmuutta voidaan nähdä koska menetelmät ovat kehitetty pääasiallisesti ohjelmistokehittäjille. Voiko ketteristä menetelmistä Scrum-viitekehystä hyödyntää muissakin kuin ohjelmistokehityksessä?

Tutkimukseni tavoitteena on tutkia, kuinka palvelumuotoilutyön soveltuminen onnistuu nykyajan ketterässä Scrum-viitekehityksessä. Tarkoituksena on selvittää, että sopiiko Scrum-viitekehitys vain ohjelmistokehittäjien työhön vai parantaako viitekehitys jopa muiden alojen työskentelytapoja. Parantaako viitekehitys palvelumuotoilutyössä suunnittelijoiden reagoitavuutta, läpinäkyvyyttä, tehokkuutta, työhyvinvointia ja tuottaako suunnittelijat sen avulla uusia laadukkaampia ratkaisuja.

4.3 Haastatteluiden toteutus

Teemahaastattelu valikoitui tutkimukseni aineiston hankintamenetelmäksi. Haastatteluissa halusin selvittää, kuinka haastateltavat kokivat palvelumuotoilutyönsä siirtyttyä käyttämään ketterää kehitystä. Teemahaastattelulle ominaista on haastateltavien yhdistävä kokemus kyseisestä tilanteesta. (Hirsjärvi, S. & Hurme, H. 2015, 48). Haastattelukysymykset jaottelin kahteen aihealueeseen:

Palvelumuotoilutyön ja ohjelmistokehitystyön kartoittaminen:

- 1) Mikä on palvelumuotoilun / ohjelmistokehityksen merkitys?
- 2) Mitä tiimisi tekee, mitä työkaluja ja menetelmiä käytätte?

Ketterä kehitys, tarkastelussa Scrum-viitekehitys:

- 1) Kuinka tärkeänä koet ketterän kehityksen ja siihen siirtymisen?

- 2) Mitkä ovat kokemuksesi Scrum-viitekehyksestä?
- 3) Mitä Scrum-viitekehys on mahdollistanut työssäsi?
- 4) Millaiset asiat ovat mahdollisesti estäneet sujuvaa yhteistyötä?

Palvelumuotoilutyön ja ohjelmistokehitystyön kartoittamisen vaihetta voidaan kutsua alustavaksi kartoittamiseksi tutkimuksen kannalta. Tässä osiossa halusin saada selville haastateltavien taustat liittyen omaan työhön. Alustavat kysymykset pidettiin tarkoituksella helppoina, jotta haastatteluun syntyisi mahdollisimman rento ilmapiiri. Alustavien kysymysten jälkeen keskityin haastattelussa hahmottamaan minkälaista työtä palvelumuotoilijat sekä ohjelmistokehittäjät tekevät sekä minkälaisia työkaluja ja menetelmiä he käyttävät työssään. Tämän avulla pystyin muodostamaan haastateltavista käyttäjäpersoonat, jotka edesauttoivat minua vertailemaan samankaltaisuuksia sekä eroavaisuuksia palvelumuotoilutyön ja ohjelmistokehitystyön välillä.

Tämän jälkeen haastattelun tavoitteena oli selvittää, miksi ketterää kehitystä pidetään tärkeänä ja miksi koetaan sen olevan tärkeätä. Tässä osiossa keskityttiin myös siihen, että minkälaisia kokemuksia haastateltavilla on ollut ketterään kehitykseen siirtymisellä sekä, miten Scrum-viitekehys on soveltunut heidän työhönsä. Kysymykset myös keskittyivät siihen, mitä Scrum-viitekehys mahdollistaa haastateltavien työssä, ovatko he kokeneet onnistumisia työssään sekä mitä mahdollisia esteitä on syntynyt.

Haastatteluun valitsin kaksi palvelumuotoilijaa sekä yhden ohjelmistokehittäjän. Haastateltavat olivat tarkoituksella eri ikäisiä ja heillä oli erilaiset taustat, jotta saisin tutkimuksen kannalta mahdollisimman kattavan aineiston. Ohjelmistokehittäjän rooli haastatteluissa oli antaa osviittaa siitä, kuinka Scrum-viitekehys kuuluu toimia oikeaoppisesti kehittäjän työssä sekä mitä palvelumuotoilijat voivat ottaa opiksi soveltaessaan Scrum-viitekehystä omassa suunnittelutyössään.

Haastateltavat tavoiteltiin COVID-tautitilanteen tilanteen vuoksi sähköpostilla ja haastattelut toteutettiin jonkinlaisen haastateltavalle sopivan kokousalustan avulla. Haastattelut kestivät noin puoli tuntia, mutta haastatteluiden toteutukseen varattiin riittävästi aikaa jotta tuen mahdollisista teknisistä tai muista ongelmista. Haastattelun rento ilmapiiri taattiin aloittamalla keskustelemalla haastateltavien voinnista ja päivän kuulumisista. Rennon ilmapiiriin avulla voitiin varmistaa avoin ja luottamuksellinen tunnelma haastateltavan ja haastattelijan välille. Haastateltavat vaikuttivat kaikin puolin avoimilta ja jakoivat mielellään

mielipiteitään sekä kokemuksiin. Haastateltaville kerrottiin ennen haastatteluun siirtymistä, että haastattelut tullaan nauhoittamaan, mutta tallenteet pysyvät vain minun hallinnassani. Äänitykset toteutettiin kahdella eri tavalla välttääkseen tallenteiden korruptointia. Yksi tavoista oli kokousalustan oma nauhoitusominaisuus ja toinen oman nauhurin kytkeminen äänilaitteeseen. Haastatteluiden aikana kirjoitin myös muistiinpanoja, jotta tärkeimmät huomiot nousisivat esille jo litteroinnissa. Haastateltavat pysyvät anonyymeinä eikä haastateltavien tietoja jaeta kolmansille osapuolille.

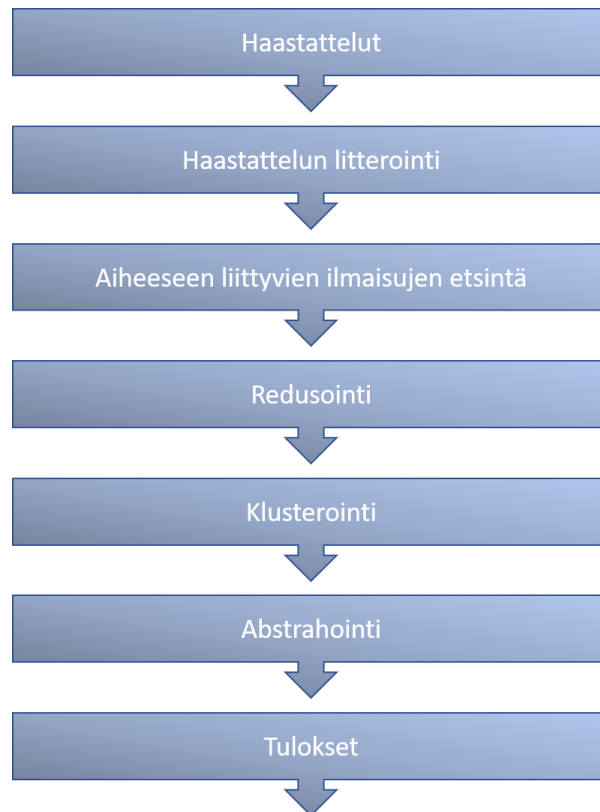
Haastatteluiden suoritettua aloitin haastatteluiden litteroinnin eli haastatteluiden kirjoittaminen sanallisesti puhtaaksi. Litterointi toteutettiin Microsoft Officen Word-tekstinkäsittelyohjelmalla. Tein litteroinnin sanatarkasti, mutta tarpeettomat asiat tutkimuksen kannalta hyppätiin yli. Selkeät litteroinnit helpottivat minua, kun siirryin sisällönanalyysin tekemiseen. Tutkimuksen kannalta merkittävät asiat yliviivattiin erilaisilla Word-tekstinkäsittelyohjelman tarjoamilla tekstin korostusväreillä. Litterointeja ei ole liitetty opinnäytetyöhön, koska haastateltavien henkilöllisyyden anonyymisyys haluttiin säilyttää, mutta ne löytyvät opinnäytetyön tekijän arkistoista.

4.4 Aineiston analyysi

Kvalitatiivisen tutkimuksen aineiston analyysimenetelmäksi valittiin aineistolähtöistä sisällönanalyysiä. Aineistolähtöisen sisällönanalyysin tehtävänä on rakentaa laadullisen tutkimuksen aineistosta selkeä sanallinen kuvaus muuttamatta laadullisen tutkimuksen sisältämää tietoa. Tarkoituksena on pilkkoa aineistoa pieniin palasiin ja esiin nousseille teemoille etsitään niitä kuvaavia käsitteitä. Tämän myötä voidaan koota aineisto uudelleen aukottomaksi kokonaisuudeksi. (Tuomi & Sarajärvi 2018, 121–122.)

Haastatteluiden jälkeen laadullisen tutkimuksen aineiston analyysi toteutettiin aluksi keräämällä yhteen vastaukset, jonka jälkeen sisällönanalyysin menetelmin aineistoa alettiin analysoida. Sisällönanalyysi jaetaan kolmivaiheiseksi prosessiksi: aineiston redusointi eli aineiston pelkistäminen, klusterointi eli ryhmittely ja abstrahointi eli aineiston käsitteellistäminen. (Tuomi & Sarajärvi 2018, 122.)

Aineistoanalyysin prosessia voidaan tarkastella kokonaisuudessaan kuvasta 10.

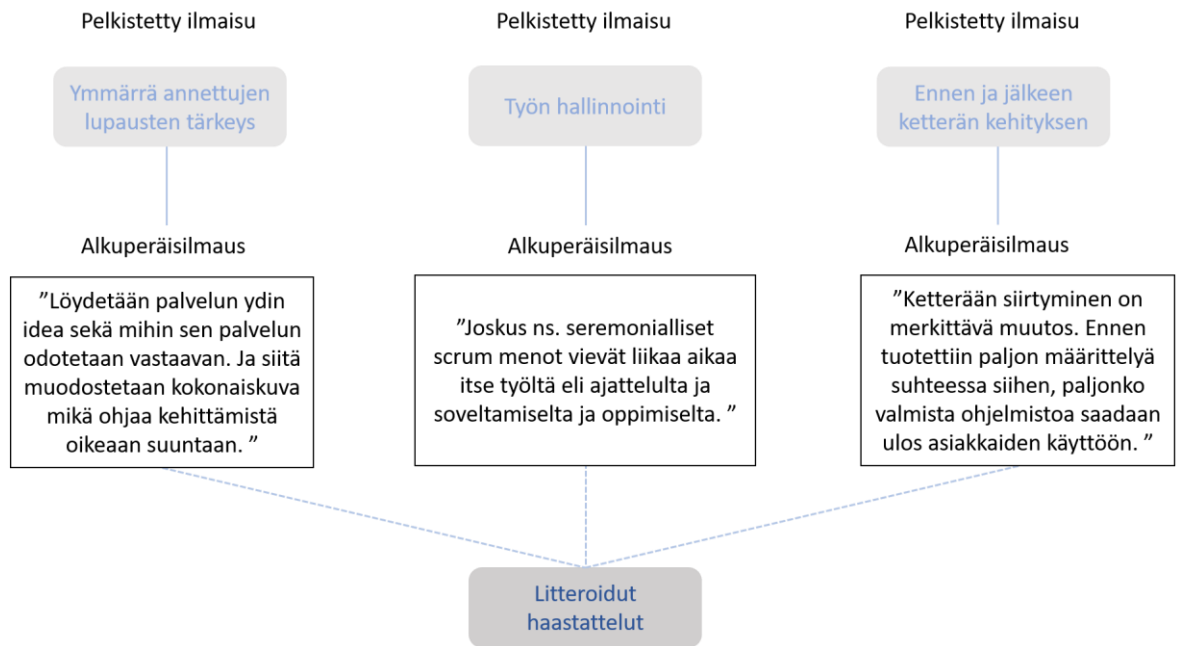


Kuva 10. Tutkimusprosessi kokonaisuudessa.

Aineiston pelkistämistä kutsutaan redusoinniksi. Redusoinnin tarkoituksena on pyrkiä löytämään tutkimuksen aineistosta tutkimuksen kannalta tärkeimmät oleelliset asiat.

Redusointi vaiheessa nostetaan esille tutkimustehtävään vastaavat ilmaisut. (Tuomi & Sarajärvi 2018, 123.)

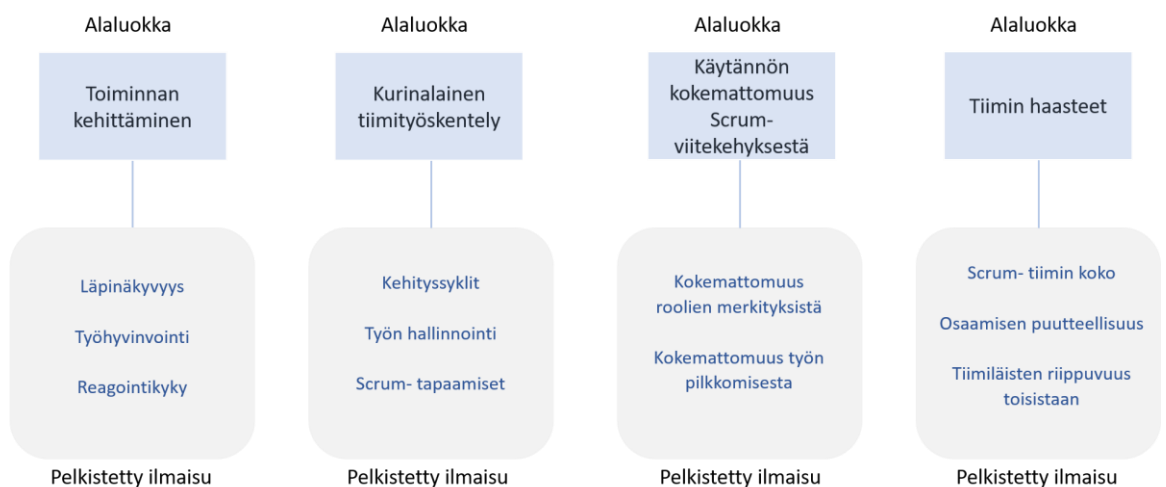
Aineiston redusointi eli aineiston pelkistäminen aloitettiin käymällä läpi haastatteluiden litterointeja ja haastatteluista esiin tulleista tärkeimmistä ilmaisuista. Ilmaisut olivat värikoodailla yliviivattu jo etukäteen, mikä auttoi redusointi vaiheessa pelkistämään niitä tarkemmiksi ilmaisuiksi. Tämän myötä toteutin luomalla PowerPointilla tutkimusaineiston pohjalta miellekartan, johon keräsin tutkimusaineiston yhtäläisyydet. Miellekarttaan ei lisätty litteroinnista esiin tulleita ilmaisuja, koska nämä olivat jo tehty etukäteen yliviivauksella. Esille tulleet yhtäläisyydet pelkistettiin miellekarttaan niitä kuvaaviksi ilmaisuiksi, jotka ovat informaation kannalta selkeämmässä muodossa. Tässä kuva 11, mikä kuvastaa redusointi vaihetta miellekartassa.



Kuva 11. Esimerkki redusoinnista

Klusteroinnin eli ryhmittelyn tarkoituksena on etsiä aineistosta samankaltaisuuksia tai eroavaisuuksia. Samankaltaisuudet ryhmitellään eri luokiksi ja tämän myötä muodostuu analyysiin alaluokat. Alaluokat nimetään kyseessä olevan luokan sisältöä vastaavan termin tai käsitteen mukaan. (Tuomi & Sarajarvi 2018, 124–125.)

Miellekartan avulla klusterointi vaiheessa eli ryhmittelyssä etsittiin pelkistettyjen ilmaisujen samankaltaisuudet ja tämän myötä pystyttiin ryhmittelemään ilmaukset omiin alaluokkiinsa. Kuvasta 12 voidaan tarkastella esimerkkinä klusterointi vaihetta, missä näkyy aineistosta esiin nousseet alaluokat.



Kuva 12. Esimerkki klusteroinnista

Abstrahoinnilla eli aineiston käsitteellistämällä erotetaan klusteroidusta aineistosta tutkimuksen kannalta oleellinen tieto, jonka perusteella muodostuu teoreettisia käsitteitä. Aineiston käsitteellistämistä jatketaan yhdistelemällä luokituksia niin pitkään kuin se koetaan tarpeelliseksi. (Tuomi & Sarajärvi 2018, 125–126.)

Aineisto abstrahointi vaiheessa yhdisteltiin sekä ryhmiteltiin klusterointi vaiheessa syntyneet alaluokat samankaltaisuuksien perusteella. Alaluokkien yhdistämisen jälkeen ryhmille etsittiin parhaiten kuvaava ilmaisu, josta muodostui kyseisen ryhmän yläluokka. Analyysin myötä esiin nousi kolme yläluokkaa, mitä voidaan tarkastella miellekartan (Kuva 13) abstrahointi vaiheesta.



Kuva 13. Abstrahointi vaiheesta syntyneet yläluokat

5 Tutkimustulokset

Tässä osiossa käydään läpi opinnäytetyöni tutkimustuloksia, jonka tavoitteena oli selvittää miten Scrum- viitekehys soveltuu palvelumuotoilutyöhön. Tuloksissa nousi esiin kolme eri osa-aluetta, joita ovat palvelumuotoilutyön ero ohjelmistokehitykseen, miten Scrum on soveltunut palvelumuotoilutyöhön ja mitä hyötyjä palvelumuotoilijat ovat kokeneet saaneensa siitä sekä mitkä asiat ovat haitanneet Scrumin toimivuutta palvelumuotoilutyössä. Haastatteluun valitsin kaksi palvelumuotoilijaa sekä yhden ohjelmistokehittäjän. Ohjelmistokehittäjän rooli tutkimustuloksissa oli antaa osviittaa siitä, kuinka Scrum-viitekehys kuuluu toimia oikeaoppisesti kehittäjän työssä sekä mitä palvelumuotoilijat voivat ottaa opiksi soveltaessaan Scrum-viitekehystä omassa suunnittelutyössään. Kaksi palvelumuotoilijaa on nimetty tutkimustuloksissa A1 ja A2. Ohjelmistokehittäjä on nimetty B1.

Ensimmäinen osa-alue käsittelee palvelumuotoilutyön eroavaisuutta ohjelmistokehitykseen. Nämä eroavaisuudet jaottelin kolmeen alaluokkaan, mitkä koin tärkeäksi osoittamaan, että molempien alojen työskentely saattaa olla erilaista, vaikka yhtäläisyyksiäkin löytyy.

Toisessa osa-alueessa keskitytään enimmäkseen siihen mitä hyötyjä palvelumuotoilijat ovat kokeneet Scrumista ja miten se on soveltunut heidän työhönsä. Ohjelmistokehittäjän vastuksia vertaillaan palvelumuotoilijoiden vastauksiin silloin, jos niiden välillä on eroavaisuuksia.

Kolmannessa osa-alueessa käydään läpi asioita mitkä ovat haitanneet Scrumin toimivuudessa tai mitä haittoja Scrum on aiheuttanut palvelumuotoilutyön onnistumisessa. Tässäkin osiossa ohjelmistokehittäjän vastuksia käytetään hyödyksi, jos haastateltavien välillä löydetään eroavaisuuksia.

5.1 Palvelumuotoilutyön ero ohjelmistokehityksestä

Palvelumuotoilutyön ero ohjelmistokehitystyöhön nousi esiin haastatteluiden aikana useaan otteeseen. Nämä eroavaisuudet jaottelin kolmeen alaluokkaan (Kuva 14), mitkä koin tärkeäksi osoittamaan, että molempien alojen työskentely saattaa olla erilaista, vaikka yhtäläisyyksiäkin löytyy.



Kuva 14. Palvelumuotoilutyön ero ohjelmistokehitykseen

Tiimien koko ja osaamisen monipuolisuus

Ohjelmistokehittäjä kuvailee tiimiänsä tiiviiksi joukoksi, mikä koostuu pääsääntöisesti suunnittelijoista- koodareihin. Jokainen kehittäjän osaaminen on melko yhtenäistä ja tämän vuoksi he voivat tukea tosiaan erilaisissa työtehtävissä.

B1 ”Tiimimme on aika tiivis joukko, meitä taitaa olla 8 kehittäjää. Voisin kuvailla meitä laajaksi joukoksi osaajia, osaamista löytyy suunnittelusta- koodaamiseen”

Palvelumuotoilijat kuvailivat tiimiänsä suureksi joukoksi, joilla on hallussa valtavasti erilaisia työkaluja sekä menetelmiä asiakasymmärryksen kartoittamista varten. Palvelumuotoilijat koostuvat käyttäjälähtöisistä- ja käyttöliittymäsuunnittelijoista. Vaikka osaamista löytyy paljon, niin jokaisen tiimin jäsenen henkilökohtainen osaaminen rajoittuu omalle mukavuusalueelle. Tiimissä ei ole valtavasti henkilöitä ketkä osaavat sekä käyttäjälähtöisistä- suunnittelua että käyttöliittymäsuunnittelijoista. Osaaminen rajoittuu aika lailla vaan palvelumuotoiluun eikä tiimissä löydy koodareita tai testaajia.

A1 ”Meillä palvelumuotoilutiimissä ei välttämättä ole niin laajaa osaamista, koska ei ole välttämättä niitä koodareita sekä testaaajia/testausosaamista.”

Työtehtävät/Tilaukset

Palvelukokemuksen käyttäjälähtöiseen suunnitteluun palvelumuotoilijoilla on hallussa valtavasti erilaisia menetelmiä ja työkaluja. Asiakasymmärryksen kartoittamista palvelumuotoilijat tutkivat keskittymällä olemassa olevia taustatietoja sekä palveluiden käyttäjien tutkimiseen. Työtehtävissään tiedon avulla palvelumuotoilijat rakentavat erilaisia käyttäjäpersoonia, käyttäjätarinoita ja palvelupolkuja palvelukokonaisuuden kuvaamista varten. Asiakasymmärryksen kartoittamista varten palvelumuotoilijoita tilataan erilaisia työpajoja. Työpajoissa keskitytään luomaan mahdollisimman paljon tietoa ja ideoita suhteellisen lyhyessä ajassa. Työtilaukset kallistuvat myös käyttöliittymäsuunnitteluun, missä palvelumuotoilijat rakentavat erilaisia käyttöliittymäprotoja. Palvelumuotoilijat eroavat ohjelmistokehittäjiin heidän työtilausten myötä. Palvelumuotoilijoilla on työnalla paljon erilaisia tilauksia, kun taas ohjelmistokehittäjät keskittyvät yhteen tuotteeseen.

A1 ”Meillä palvelumuotoilutiimissä on taas monta tilausta samanaikaisesti työn alla ja koko tiimi ei välttämättä tee sitä samaa yhtä tilausta”

A2 ”Me tarjoamme asiakkaille esimerkiksi intensiivisiä työpajoja design sprinttejä. Asiakasymmärrystä kartoitamme käyttäen apuna erilaisia menetelmiä kuten käyttäjäprofiilit, palvelupolut yms.”

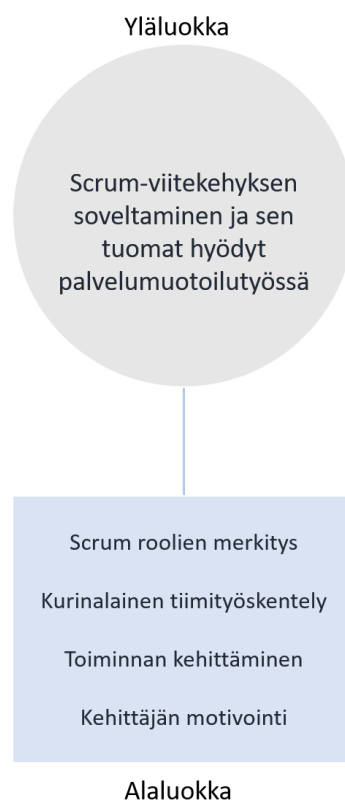
A2 ”Tarjoamme myös erilaista käyttöliittymäsuunnittelu osaamista asiakkaillemme.”

Ohjelmistokehittäjillä on aina melko selvät sävelet siitä mitä he tekevät ja minkälaisia työtilauksia heiltä pyydetään. Ohjelmistokehittäjät pääsääntöisesti keskittyvät yhden tuotteen kehitykseen. Ohjelmistokehittäjät keskittyvät pääsääntöisesti omaan tuotteen kehittämiseen, mutta he saattavat auttaa toisia kehitystiimejä tarpeen tullen. Nämä ohjelmistokehittäjät ja ohjelmoijat työskentelevät ohjelmistojen suunnittelun parissa, niiden toteutuksessa sekä kehityksen ja ylläpidon merkeissä.

B1 ” Kehitystiimimme toteuttaa yhtä tuotetta ja tietenkin ajan myötä saattaa tulla useampia tuotteita mutta yleensä yksi tuote on se, jota kehitetään suuremmin.”

5.2 Scrum- viitekehyksen soveltaminen ja sen tuomat hyödyt

Jokaista haastateltavaa yhdisti se, että ketterään kehitykseen siirtyminen on ollut uutta sekä kannattavaa. Ohjelmistokehittäjä on ollut osana ketterää kehitystä hieman pidempään kuin palvelumuotoilijat. Haastateltavien mielestä siirtyminen ketterään kehitykseen on tapahtunut onnistuneesti, vaikka on ollut paljon opittavaa. Scrum- viitekehys on alkuun suunniteltu ohjelmistokehittäjille ja tämän vuoksi haastatellun ohjelmistokehittäjälle Scrumin soveltaminen omaan työhön on sujunut melko vaivatta. (kuva 15).



Kuva 15. Scrum- viitekehyksen soveltaminen ja sen tuomat hyödyt

Scrum roolien merkitys

Kaikkien kolmen haastateltavan mielestä Scrumin roolituksilla on suuri merkitys erilaisissa kehitettävissä projekteissa. Jokaisen haastateltavan Scrum- tiimistä löytyy vaadittavat kolme erilaista Scrum- viitekehyksen roolitusta, mitkä ovat tuoteomistaja, kehitystiimi ja Scrum Master.

Palvelumuotoilijoiden mielestä tuoteomistajalla on tärkeä rooli Scrum- tiimissä, jopa tärkein. He näkevät, että ilman tuoteomistajan läsnäoloa kehitettävän tuotteen hallinnointi olisi huomattavasti haasteellisempaa Scrum-projektin aikana. Palvelumuotoilutiimissä tuoteomistaja toimii asiakkaan näkemyksen sanansaattajana ja hänen avullaan tärkeimmät kehitettävää tuotetta koskevat vaatimukset tullaan toteuttamaan. Tämä tuoteomistaja on selkeästi auttanut palvelumuotoilijoita keskittymään konkreettiseen työntekoon, koska heidän ei tarvitse käyttää aikaa asiakkaan näkemysten selvittelyyn. Yhden palvelumuotoilijan mielestä tuoteomistajalla on uudenlainen haaste edessä kehittymiseen liittyen, ja tämän esiin tulleen haasteen tulen kertomaan kolmannessa osiossa.

A1 ”Tuoteomistajan rooli on ehkä kaikista tärkein, tuoteomistajan pitäisi pystyä tekemään kehitystiimille selkeäksi mitä tehdään. Muuten kehitystiimi on aika eksyksissä sekä oman onnensa nojassa.”

Scrum-viitekehyksen soveltaminen palvelumuotoilutyöhön on osoittanut kuinka merkittävä Scrum Masterin rooli on. Scrum Masterin tarkoitus on tarjota apua Scrum- tiimeille viitekehykseen liittyvissä asioissa. Palvelumuotoilijat kokevat, että he voivat puhua Scrumiin liittyvistä ongelmistaan avoimesti Scrum Masterille ja osaavat odottaa vastineeksi Scrum viitekehykseen liittyvää koulutusta ja ohjausta. Heidän mielestään on tärkeitä, että tiimistä löytyy henkilö, joka pystyy äkillisesti ratkaisemaan ongelmia mitkä muuten voivat hidastaa työntekoa. Palvelumuotoilijoiden mielestä Scrum Masterin roolin omaksumisessa on kehitettävää, ja tämän esiin tulleen haitan tulen kertomaan tarkemmin kolmannessa osiossa.

A2 ” Scrum Master auttaa niin tuoteomistajaa kuin kehitystiimiä kehittämään tekemistä ja tukee Scrum- viitekehykseen liittyvissä asioissa”

Palvelumuotoilijoiden mielestä he eivät ole varsinaisia kehittäjiä vaan pitävät itseään enemmän suunnittelijoina. Heidän niin sanottu ”suunnittelijatiimi” on normaalia isompi mihin Scrum- viitekehyksessä on ajateltu, mutta työskentely sujuu hyvin sprintistä toiseen. He pitävät Scrum- viitekehyksen ajattelutavasta, että kehitystiimillä on valta päättää työnjaosta sekä tavoitteista. He pitävät toisiaan tasa-arvoisina, vaikka jokaisella on valtavasti osaamista eri asioista. Toisten tukeminen ja vastuun kantaminen on auttanut suunnittelijatiimin yhteishengen luomisessa.

A1” Olemme palvelumuotoilijat enemmän suunnittelijoita kuin kehittäjiä.”

Palvelumuotoilijoiden ajatukset roolituksen merkityksestä ovat yhtenäisiä ohjelmistokehittäjän kanssa. Scrum- viitekehityksen tarjoamat roolitukset toimivat palvelumuotoilutiimissä juuri niin kuin viitekehitys on suunnitellut.

Kurinalainen tiimityöskentely

Palvelumuotoilijoilla Scrumin kehitystyö tapahtuu 3 viikon sprinteissä eli kehityssykleissä. Palvelumuotoilijoiden sprintit alkavat yhteisellä sprintin suunnittelupalaverilla, mihin toivotaan kaikkien tiimiläisten osallistuvan. Sprintin suunnittelupalaverissa tehdään sprintin tehtävälista, minkä avulla hallinnoidaan tiimin työskentelyä seuraavat kolme viikkoa. Sprintin aikana tiimi pitää päivittäisiä Scrum- tapaamisia (eng. daily). Sprintin päätteeksi he toteuttavat yhteisen sprintin katselmoinnin.

Palvelumuotoilijat ovat kokeneet tärkeäksi sprintin suunnittelupalaverit, koska se on keskeinen osa kehitysjonon työstössä. Ilman sitä he eivät tietäisi mitä olisivat tekemässä kolmen viikon kestävän sprintin aikana. Tämän vuoksi kehitysjono koetaan tärkeäksi ja heidän mielestään se kuuluu suunnitella tarkkuudella, koska kehitysjono määrittää ne toteutettavaksi valitut tehtävät mitkä koetaan liiketoiminnan kannalta tärkeäksi. Kehitysjonon työstössä palvelumuotoilijat ovat havainneet ongelmia, johtuen useista työtilauksista. Tästä tulen kertomaan tarkemmin kolmannessa osiossa.

A1 ” Kehitysjono on ihan ehdoton edellytys, että työt ylipäättänsä edes sujuvat. Kehitysjonon backlogin työstämiseen varataan riittävästi aikaa ja sitä käydään yhdessä tiimin kanssa läpi, jotta tiimille on selkeätä, että mitä siellä on tulossa ja mitä voidaan odottaa.”

Ilman sprintin tehtävälistaa työnteko olisi sekavaa ja hidasta. Tehtävälistaa suunnitellessa palvelumuotoilijat valitsevat jokaiselle toimeksiannolle vastuuhenkilöt. Sen avulla taataan, että tarvittavat tehtävät saadaan tehtyä. Tehtävälistaan sprintin tehtävät pilkotaan mahdollisimman pieniksi palasiksi ja tämä on auttanut merkittävästi työn tehokkuuteen sekä läpinäkyvyyteen. Tehtävien pilkkomisen avulla voidaan myös jakaa työtakkaa suunnittelijoiden kesken.

Päivittäisillä Scrum- tapaamisilla palvelumuotoilutiimi koordinoi työtään jakamalla toisilleen tietoa omista tekemisistään. He ovat kokeneet, että haasteista ja tehdyistä töistä puhuminen levittää tietämystä tiimin sisällä ja mahdollisiin ongelmiin voidaan tarttua nopeasti ennen suurempaa vahinkoa.

Nämä vastaukset ovat todella yhtenäisiä ohjelmistokehittäjän vastausten kanssa. Kurinalainen työnteko on selkeästi auttanut palvelumuotoilijoita työssään, työtehokkuus on lisääntynyt ja mahdollisiin haasteisiin ehditään ennakoida ennen suurempaa vahinkoa. Tässä osiossa hyödyt Scrum- viitekehyksen soveltamisesta palvelumuotoilutyöhön ei eroa ohjelmistokehittäjiin. Kaikkia haastateltavia yhdisti yhteinen haitta liittyen liiallisiin Scrum-tapaamisiin. Tästä tulen kertomaan tarkemmin kolmannessa osiossa.

Toiminnan kehittäminen

Molemmat palvelumuotoilijat ovat kokeneet, että lyhyissä ja tiiviissä jaksoissa työskentely on auttanut valtavasti tiimin tuottavuutta ja sen läpinäkyvyyttä. Lyhyiden jaksosten avulla voidaan tarkastella tiimin toimintaa ja mukautua helposti sekä äkillisesti mahdollisiin muutoksiin. Jokaisen sprintin päätteeksi palvelumuotoilijat toteuttavat retrospektiivin. Retrospektiivi auttaa tarkastelemaan sprintin onnistumisia toimintatapojen perspektiivistä. Heidän mielestään on hyvä käydä läpi säännöllisesti asioita, mitkä olivat toimineet hyvin sprintillä sekä mitä on parannettavaa tulevalle sprintille. Jatkuva toiminnan kehittäminen auttaa tuottavuuteen ja työtyytyväisyyteen.

Kehittäjän motivointi

Scrumia käyttäessä palvelumuotoilijat ovat kokeneet motivoivaksi sen, että tuotteen tilaa voidaan jatkuvasti tarkastella ja vaikuttavat tekijät tuotteen edistymiseen ovat selkeästi havaittavissa. Tehtävien pilkkomisella saadaan suuretin työt näyttämään pieneltä ja hoidettavilta lyhyissä sprinteissä.

A2 ”Scrum- tiiminä työskentely on ollut motivoivaa, koska olemme voineet luoda yhteiset tavoitteet tiimillemme sekä saamme itse vaikuttaa omaan työhömme ja työskentelytapoihin.”

5.3 Mahdolliset estävät syyt Scrumin- viitekehyksen toimivuuteen palvelumuotoilutyössä

Tässä osa-alueessa käymme esille tulleet haitat Scrumin toimivuudesta. Kaikki haastateltavat ovat kokeneet haasteita ja nämä eivät ole täysin samoja eri tiimeissä. Vastauksista voidaan huomata, että palvelumuotoilijoilla on ollut enemmän haasteita Scrum- viitekehyksen soveltamisessa omassa työssään. Haasteet johtuvat enimmäkseen käytännön kokemuksen puutteesta. (kuva 16)



Kuva 16. Mahdolliset estävät syyt

Resurssit

Yksi keskeinen haitta kaikkien haastateltavien mielestä oli ajanhallinta. Scrum- viitekehityksen myötä työkalenterit täyttyvät pakollisista Scrum- seremonioista. Näitä pakottavia seremonioita ovat esimerkiksi kehitysjonontyöstö, dailyt ja ylimääräiset katselmoinnit. Sekä palvelumuotoilijat, että ohjelmistokehittäjät kokivat seremonioiden vievän tärkeitä aikaa itse työltä ja ajan niukkuus voi aiheuttaa valtavaa turhautumista tiimien sisällä.

A2 ”Joskus ns. seremonialliset Scrum menot vievät liikaa aikaa itse työltä eli ajattelulta ja soveltamiselta ja oppimiselta. Toisaalta projektinhallinnan eli projektin päätyminen ja loputtaminen ja siihen menevä viikoittainen aika syö turhaan muiden resurssien hallinnan taitoa.”

B1 ”Ongelminä olen huomannut työssäni projektinhallinnan ja siitä johtuvat resurssoinnin ongelmat. Aikaa on niukasti jatkuvien kokousten takia.”

Palvelumuotoilijoille on myös ilmentynyt ajallinen resurssipula Scrum- viitekehityksen soveltamisen aikana. Heidän työnsä keskittyy käyttäjälähtöiseen suunnitteluun ja tämän vuoksi

monet projektit ovat riippuvaisia muiden ihmisten, loppukäyttäjien ja palveluasiakkaiden osallistamisesta. Näiden ihmisten saaminen osaksi suunnittelutyöhön pienellä aikavälillä tuottaa hyvin harvinaista ja tämä ongelmia työn valmistumiseen ajallaan.

Tiimin haasteet

Palvelumuotoilijalla osaaminen on erittäin monipuolista, mutta on havaittavissa vahvaa henkilöitymistä. Projektien tekeminen on riippuvainen työntekijän osaamisesta. Scrum-tiiminä palvelumuotoilijat ovat myös liian suuri, 13 suunnittelijaa. Tämän vuoksi palvelumuotoilijat ovat luoneet tiimin sisälle erilaisia pienryhmiä, minkä avulla voidaan taata työtilausten valmistumista ajallaan ja annetaan tukea tilausten tekemiseen. Tällä tavalla saadaan osaamista levitettyä ja tiimiläisten ei tarvitse yksin huolehtia tilausten suorittamista.

Käytännön kokemus Scrum- viitekehyksestä

Kaikille haastateltaville on tarjottu koulutusta Scrum perusteille, mutta tämäkin ollut vain perusteellista. Perusteet ovat hallinnassa ja ymmärrystä Scrumista löytyy, mutta kokemus vaikuttaa siihen, kuinka Scrum toimii käytännössä. Käytännön kokemuksen avulla osataan hahmottamaan minkä verran sprintillä saadaan työtä aikaiseksi. Ja vaikka kuinka hyväksi haastateltavat ovat kokeneet tehtävien pieniksi pilkkomisen niin välillä tehtävien arviointi on ollut haasteellista.

B1 "Tehtävien väärin arviointi on lisännyt tiimissämme työtakkaa. Arvioimme useasti tehtävät liian pieniksi mitä ne ovat kokonaisuudessaan."

A1 "Meillä myös saattaa kestää kauemmin jonkun ominaisuuden saaminen valmiiksi, mutta kumminkin saamme aina jotain osia valmiiksi tilauksista ja se ei välttämättä ole niin iso ero. Tähän vaikuttaa oikeastaan väärinarviointi työmäärästä sekä yleisesti vaan tiimin kokemattomuus, kun Scrum on meille aika uutta. Ei osata vielä pilkkoa tekemistä riittävän pieniksi osiksi."

A2 "Mutta puutteita on enemmän tiimillä käytännönasioissa, koska ei ole kokemusta käytännöstä ja koulutuksista vaan tarjotaan pohja Scrum perusteille."

Palvelumuotoilutiimissä ilmeni kumminkin se eroavaisuus ohjelmistokehittäjiin, että tuoteomistaja ei ole yksittäisen tuotteen omistaja vaan usean. Vaikka tuoteomistajan läsnäolo vaikuttaa merkittävästi projektityöskentelyn selkeyteen, niin henkilökohtaisesti palvelumuotoilutiimin tuoteomistajalla on kehitettävää omassa toiminnassaan.

A1 ”Tuoteomistajana haluan kehittyä koska palvelumuotoiluryhmän tuoteomistajana asettaa erilaisia haasteita kuin yhden tuotteen tuoteomistajana toimiminen.”

Myös Scrum Masterin käytännön kokemus Scrumista on erilainen verrattuna ohjelmistokehittäjiin. Scrum Master palvelumuotoilutiimissä ei pysty tarjoamaan sellaista tukea tiimilensä, mitä Scrum- viitekehityksessä on ajateltu. Tämä kokemus kumminkin kasvaa sprintiltä sprintille ja ovat nähneet valtavaa kehitystä hänessä.

A1 ” Scrum Masterilla ei ole ollut aikaisempaa kokemusta Scrum Masterin töistä sekä kehitystiimin jäsenille kokemus scrumista on ollut uutta. Eli kaikki ollaan tavallaan uuden haasteen edessä, mutta on toiminut hyvin ja on lähtenyt yllättävän hyvin liikkeelle. Pahemista sudenkuopista on päästy yli ja ehkä jopa ollaan väistetty ne kokonaan. ”

6 Pohdinta

Tutkimus alkoi haastattelemalla erilaisia palvelumuotoilualan ammattilaisia ja heidän kokemuksiaan Scrum- viitekehyksen soveltamisesta palvelumuotoilutyöhön. Tutkimukseen myöhemmin otettiin ohjelmistokehittäjän ajatuksia siitä, kuinka heille suunniteltu Scrum- viitekehys on toiminut. Tämän vertailun apuna pystyimme arvioimaan sitä, että onko oikeasti Scrum- viitekehys suunniteltu pelkästään ohjelmistokehittäjille vai voidaanko sitä soveltaa muillekin aloille.

Kokemukset Scrum- viitekehuksesta palvelumuotoilijoille ovat olleet kaikin puolin hyvät. Haastateltavat toivat yhteisesti esille, että siirtyminen Scrum- viitekehykseen on edistänyt merkittävästi suunnittelutiimin avoimuutta sekä läpinäkyvyys tiimin sisällä on parantunut huomattavasti. Suunnittelutiimi on kyennyt parantamaan kehitettävien palveluiden ja tuotteiden tuottavuutta kuin myös ominaisuutta sekä laatua. Tehtävien henkilöityminen on vähentynyt kehitystiimin sisällä läpinäkyvyyden myötä. Scrum- tiimin yksi kriittisistä menestystekijöistä on läpinäkyvyys. Läpinäkyvyyden hyödyt voidaan havaita sekä tuotekehityksessä että toiminnan kehittämisessä. Lyhyiden sekä intensiivisten sprinttien myötä tiimin toimintaa voidaan arvioida ja mukauttaa helposti sekä ketterästi tarpeiden mukaan. Sprintin lopussa järjestettävässä retrospektiivissä (Retrospective) arvioidaan menneen sprintin onnistumiset ja kehityskohteet tulevaa sprinttiä varten. Retrospektiivi auttaa kehittämään tiimin toimintatapoja ja organisaatiota tunnistaa heidän potentiaaliset kykynsä

Scrum- viitekehyksen myötä on myös havaittu paljon haasteita palvelumuotoilutiimissä. Nämä haasteet rajoittuvat resursseihin, Scrum- tiimin haasteisiin sekä Scrum- tiimin kokemukseen Scrum- viitekehuksesta. Näihin haasteisiin on kyllä olemassa ratkaisuja välttääkseen suurempaa vahinkoa.

Resurssipula eli ajanhallinta on aika yleinen ongelma Scrumissa. Scrum- viitekehyksen myötä työkalenterit täytyvät pakollisista Scrum- seremonioista ja näitä pakottavia seremonioita ovat esimerkiksi kehitysjonontyöstö, dailyt ja ylimääräiset katselmoinnit. Seremoniat vievät selkeästi paljon tärkeätä aikaa itse työltä ja ajan niukkuus voi aiheuttaa valtavaa turhautumista tiimien sisällä. Tätä ongelmaa voidaan pienentää yhdistelemällä eri seremonioita yhteen.

Palvelumuotoilijoilla osaaminen on erittäin monipuolista, mutta osaaminen voi henkilöityä erittäin nopeasti. Projektien tekeminen ja valmiiksi saaminen on riippuvainen työntekijän osaamisesta. Myös Scrum- tiimin suuruus voi haitata projektien etenemistä tai aiheuttaa jopa resurssipulaa ajanhallinnan näkökulmasta. Suuressa Scrum- tiimissä kokoukset on

mahdottomia pitää alle 15 minuutin pituisina. Kokousten venyminen voi myös vaikeuttaa tiimiläisten keskittymistä. Tähän ratkaisuna voisi olla Scrum- tiimin sisälle kehitettäisiin pienryhmiä, minkä avulla voidaan taata työtilausten valmistumista ajallaan ja kehittäjät antavat tukea toisilleen työtilausten tehtäessä. Tällä tavalla saadaan myös osaamista levitettyä ja tiimiläisten ei tarvitse yksin huolehtia tilausten suorittamisesta.

Viimeisenä esteenä oli Scrum- tiimin käytännön kokemus itse Scrumista. Scrumin täydellinen soveltaminen ei onnistu, vaikka kaikille Scrum- tiimiläisille olisi tarjottu koulutusta Scrumin perusteille. Perusteet voivat olla kaikilla hallinnassa ja ymmärrystäkin Scrumista löytyy, mutta kokemus vaikuttaa siihen, kuinka Scrum toimii käytännössä. Käytännön kokemuksen avulla osataan hahmottamaan minkä verran sprintillä saadaan työtä aikaiseksi. Ja vaikka kuinka hyväksi koetaan tehtävien pieniksi pilkkominen niin välillä tehtävien arviointi voi olla haasteellista. Myös Scrum- roolien omaksuminen riippuu kokemuksen kasvattamisella.

Lopulta voidaan päätellä, että Scrum- viitekehys on toiminut juuri niin kuin se on suunniteltu. Tämän menetelmän soveltamisessa ei ole väliä kuulutko ohjelmistokehitykseen vaan, mikä tahansa tuotteita kehittävät alat voivat omaksu Scrum-viitekehyyksen omaan työhönsä. Scrum on kokeilevaa ja tehokasta, ja maksimaaliseen tehokkuuteen vaikuttaa käytännön kokemus Scrumista.

Tämän opinnäytetyön avulla voidaan varmistaa, että Scrum-viitekehys soveltuu palvelumuotoilutyöhön vaikka, jotainkin hidasteita on olemassa. Jatkokehityksen kannalta olisi mielenkiintoista soveltaa muita ketteriä menetelmiä palvelumuotoilutyöhön ja tutkia, mikä ketteristä menetelmistä voisivat sopeutua paremmin palvelukokemuksen käyttäjälähtöiseen suunnitteluun. Voidaanko jokaisesta menetelmästä ottaa parhaat palaset ylös ja luoda jotain täysin uutta?

Kokemukseni opinnäyteprosessista oli hyvät ja koen oppineeni paljon. Vaikka palvelumuotoilu sekä Scrum- viitekehys olivat minulle jo entuudestaan tuttuja, en koskaan ajatellut kuinka paljon syvällisemmin pääsen näitä tarkastelemaan teorian puolesta. Haastatteluiden toteutus tuotti omia haasteitaan. Nykyinen COVID-tautitilanne tuotti haasteita haastateltavien rekrytoinnissa sekä haastatteluiden toteuttamisessa. Tämän kautta opin sopeutumaan täysin uudella tapaa haastatteluiden toteutukseen, vaikka itse haastatteluiden tekeminen on minulle melko tuttua. Haastatteluihin kohdistuva kirjallisuus antoi minulle uudenlaista tietämystä haastatteluiden toteuttamiseen ja analysoimiseen. Opinnäytetyö oli kaiken kaikkiaan antoisa sekä itsensä kehittävä kokemus.

6.1 Tutkimuksen pätevyys ja luotettavuus

Tuomi ja Sarajärvi kertovat aineistossaan, että luotettavan tutkimuksen arvioinnissa hyviä huomioitavia asioita ovat tutkittava kohde ja sen tarkoitus, tutkijan omistautuminen tutkimukseen, miten haastateltavat valittiin ja aineisto kerättiin, tutkimuksen kesto, aineiston analyysimenetelmä ja tutkimuksen raportointi. (Tuomi & Sarajärvi 2018, 163–164.)

Tutkimuksen kohteeksi valittiin palvelumuotoilijoiden kokemukset Scrum-viitekehystä ja kuinka se on soveltunut palvelumuotoilutyöhön. Tutkimuksessa tietoperustan pohjalta helpotettiin ymmärtämään, mitä palvelumuotoilu on ja mitä Scrum-viitekehys tarjoaa ketterässä kehityksessä.

Tutkimus toteutettiin usean kuukauden aikana, jolloin tutkimuksen laatu kasvoi ajan myötä. Aineisto kerättiin haastatteluiden avulla, jotka nauhoitettiin ja litteroitiin tutkimuksen analysointia varten. Tutkimuksen analysointiin hyödynnettiin teoriaohjaavaa sisällönanalyysiä ja tutkimuksen raportointi noudatti yleisiä tutkimusmenetelmiä. Tutkimukseen valitut haastateltavat valittiin heidän ammattitaidon monipuolisuuden ja työtehtävien erolalaisuus vuoksi. Haastateltavilta saatiin paljon hyödyllistä dataa, mikä helpotti tutkimuksen kokonaisuuden toteutusta.

Lähdeluettelo

- Abraham, P. 2019. A Guide to Building Agile Culture. Luettavissa: <https://medium.com/building-the-agile-business/a-guide-to-building-agile-culture229abc3bb943>. Luettu: 15.5.2020.
- Ahonen, M. 2010. Tapaustutkimus: Soveltuuko Scrum vesiputousmallin korvaajaksi yrityksen sovelluskehitysprojekteihin? Elektroniikan, tietoliikenteen ja automaation tiedekunta. Automaatio- ja systeemitekniikan laitos. Diplomityö
- Atlassian.com. What is a kanban board. Luettavissa: <https://www.atlassian.com/agile/kanban/boards> Luettu: 29.10.2020
- Ashmore, S., Runyan K. 2014. Introduction to Agile methods. Addison-Wesley Professional.
- Bhavsar K, Shah, V & Gopalan, S. 2020. Scrum: An Agile Process Reengineering in Software Engineering. Blue Eyes Intelligence Engineering & Sciences Publication. Luettavissa: https://www.researchgate.net/profile/Krunal_Bhavsar5/publication/338527215_Scrum_An_Agile_Process_Reengineering_in_Software_Engineering/links/5e19706f299bf10bc3a353e7/Scrum-An-Agile-Process-Reengineering-in-Software-Engineering.pdf. Luettu: 16.5.2020.
- Business Technology Standard. 2020. Luettavissa: <https://btmalli.fi/> Luettu: 27.10.2020.
- Business Technology Standard. 2020. Kehittäminen. Luettavissa: <https://btmalli.fi/book/development/introduction-to-development-discipline/> Luettu: 27.10.2020.
- CGI 2016. The Agile Cultural Shift: Why Agile Isn't Always Agile. Luettavissa: <https://www.cgi.com/sites/default/files/white-papers/agile-culture-white-paper.pdf>. Luettu: 15.4.2020.
- Design Council. Luettavissa: <https://www.designcouncil.org.uk/> Luettu: 04.10.2020.
- Forsey, C. (2018) UX vs. UI: What's the Difference? Luettavissa: <https://blog.hubspot.com/marketing/ux-vs-ui> Luettu: 29.10.2020.

Heinonen, T. 2018. Kaikki muuttuu muotoiluksi – Suojaudu asianmukaisesti (Osa 1). Luettavissa: <https://medium.com/@TaneliHeinonen/kaikki-muuttuu-muotoiluksi-suojaudu-asianmukaisesti-osa-1-addbc05e6422>. Luettu: 14.4.2020.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2004. Tutki ja kirjoita. Helsinki: Tammi.

palvelub.blogspot.com 2018. Palvelupolku- kuvituskuva. Luettavissa: <http://palvelub.blogspot.com/2018/09/palvelupolku.html> Luettu: 29.10.2020.

Uxdesign.cc. Why UX and UI should remain separate. Luettavissa: <https://uxdesign.cc/why-ux-and-ui-should-remain-separate-7d6e3addb46f> Luettu: 29.10.2020.

Hirsjärvi, S. & Hurme, H. 2015. Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Helsinki:Gaudeamus University Press Oy.

Johnson E. 2015. Why is Scrum so popular? Why is Scrum so successful? Luettavissa: <https://content.intland.com/blog/agile/scrum/why-is-scrum-so-popular-why-is-scrum-so-successful> Luettu: 20.4.2020.

Koivisto, M., Säynäjäkangas, J. & Forsberg, S. 2019. Palvelumuotoilun bisneskirja. Alma Talent. Helsinki.

Knapp J., Zeratsky J., Kowitz B., 2016. Sprint. How to solve big problems and test new ideas in just five days. Bantam Press.

Lindström, J. (2010) 'Scrum parantaa näkyvyyttä ja ohjattavuutta', Reaktor, 1 November. Luettavissa: <https://www.reaktor.com/blog/scrum-parantaa-nakyvyytta-ja-ohjattavuutta> Luettu: 20.4.2020

Reason B., Lovlie L., Flu M. 2016. Service Design for Business. Wiley

Schwaber, K & Sutherland, J. 2017. The Scrum Guide. The Definite Guide to Scrum: The Rules of the Game. Luettavissa: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. Luettu: 13.4.2020.

Scrum.org 2020. The Scrum Framework- kuvituskuva. Luettavissa:
<https://www.scrum.org/resources/what-is-scrum> Luettu: 12.9.2020.

Stickdorn, M., Lawrence, A., Hormess, M., Schneider, J. 2018. This is service design doing. Sebastopol, USA: O'Reilly Media.

Tilastokeskus 2020. Kvalitatiivinen tutkimus. Luettavissa:
https://www.stat.fi/meta/kas/kvalit_tutkimus.html#tab1. Luettu: 18.4.2020.

Tuomi, J. & Sarajärvi, A. 2018. Laadullinen tutkimus ja sisällönanalyysi. Tammi.

Tuulaniemi, J. 2011. Palvelumuotoilu. Talentum Oyj.

Virtanen, J. (2016). UX-design ja UI-design: Mitä eroa niillä on? <https://contrast.fi/ux-design-ja-ui-design-mita-eroa-niilla-on/> Luettu: 29.10.2020.

Liitteet

Liite 1. Haastattelukysymykset

Palvelumuotoilu/ohjelmistokehitys

- Mikä on sinun roolisi/asema?
- Kauanko olet tehnyt palvelumuotoilua/ohjelmistokehitystä, kauanko olet ollut osallisena tässä tiimissä?
- Mikä on sinusta palvelumuotoilun/ohjelmistokehityksen merkitys?
- Minkälaisia töitä teet palvelumuotoilu- / ohjelmistokehitystiimissäsi?
- Mitä erilaisia palvelumuotoilun/ohjelmistokehityksen työkaluja käytät työssäsi?
- Minkälaisia palvelumuotoilun/ohjelmistokehityksen menetelmiä käytät työssäsi?
- Onko muita palvelumuotoilun/ohjelmistokehityksen työkaluja tai menetelmiä mitä tulee mieleesi, mutta et ole käyttänyt niitä?
- Miten sinusta palvelumuotoilu eroaa ohjelmistokehityksestä?

Ketterä kehitys

- Kuinka tärkeänä koet ketterän kehityksen ja siihen siirtymisen?
- Miten olette tehneet töitä ennen, mitä ennen ketterään siirtymistä?
- Minkälaiseen ketteryyteen kannattaa pyrkiä?
- Kauanko olet ollut osa Scrum tiimiä?
- Eroaako palvelumuotoilijat ohjelmistokehittäjistä?
- Onko sinulla ollut aikaisempaa kokemusta Scrumista?
- Mikä merkitys on Scrumin eri rooleilla ja roolituksilla?
- Miten aikataulun jaksottaminen ja aikarajat näkyvät työssäsi (sprintit jne)?
- Mitä hyötyä Scrumista on ollut suunnittelijan näkökulmasta?
- Mitä olet oppinut ja minkälaisia kokemuksia sinulla on Scrumista?
- Millaisia haasteita olet huomannut Scrumissa, onko jotain parannusehdotuksia?
- Tiedätkö muista ketteristä menetelmistä ja voiko jokin muu menetelmä tarjota parempia vaihtoehtoja?