

PALVELUMUOTOILUN HYÖDYNTÄMINEN KETTERÄSSÄ OHJELMISTOKEHITYKSESSÄ

Toimintatutkimus IT-alan konsultointiyritykselle

Tiivistelmä

Tekijä(t) Kaartinaho, Juha-Pekka	Julkaisun laji Opinnäytetyö, AMK Sivumäärä (35+1)	Valmistumisaika Syksy 2019
Työn nimi Palvelumuotoilun hyödyntäminen ketterässä ohjelmistokehityksessä.		
Tutkinto Tradenomi (AMK)		
Tiivistelmä Opinnäytetyö tehtiin käsittelemään palvelumuotoilun mahdollisuuksia ohjelmistokehityksessä. Toimeksiantajana oli suuri kansainvälinen It-alan konsultointiyritys. Tutkimus rajattiin koskemaan vain tietyn yksikön ohjelmistoalan asiantuntijoita. Tavoitteena oli kehittää kohdeyrityksen toimintamalleja sekä tämän myötä saada laadukkaampia ohjelmistokehitysohjelmia. Tutkimus toteutettiin laadullisena toimintatutkimuksena. Tutkimusmenetelminä käytettiin puolistrukturoitua yksilöhaastattelua. Haastatteluiden tuloksia analysoitiin eri tavoin, jotta niiden pohjalta pystyttiin luomaan ohjelmistokehitykseen liittyviä kehitysehdotuksia kohdeyritykselle. Kehitysehdotukset suunniteltiin niin, että niitä voi hyödyntää myös muissa yrityksissä. Työn tietoperustaan kerättiin monipuolisesti tietoa suomen- ja englanninkielisistä lähteistä. Teoreettinen viitekehys sisältää osa-alueita niin ohjelmistokehityksestä kuin palvelumuotoilustakin. Tutkimuksen avulla saatiin paljon tietoa ohjelmistokehittämisen kehittämiskohteista sekä ainakin yhdestä menetelmästä, jolla niitä voitaisiin parantaa.		
Asiasanat käyttäjälähtöinen suunnittelu, palvelumuotoilu, design sprint, ketterä, ohjelmistokehitys, Agile, SAFe, Scaled Agile Framework		

Abstract

Author(s) Kaartinaho, Juha-Pekka	Type of publication Bachelor's thesis	Published Autumn 2019
	Number of pages (35+1)	
Title of publication Title Utilizing service design in agile software development		
Name of Degree Bachelor of Business Administration		
Abstract <p>The thesis deals with the possibilities of service design in software development. The client for this thesis was large international IT consulting company. The study was limited to software experts in a specific unit.</p> <p>The aim was to develop the operating model of the target company and, as a result, to obtain higher quality software development topics. The study was conducted as a qualitative operational study. Semi-structured individual interviews were used as research methods.</p> <p>The results of the interviews were analyzed in different ways in order to generate software development suggestions for the target company. The development proposals were designed so that they can be used by other companies as well.</p> <p>The knowledge base of the thesis consisted of versatile information from Finnish and English sources. The theoretical framework includes aspects of software development as well as service design.</p> <p>The study provided a wealth of information on the areas for software development and at least one method to improve them.</p>		
Keywords User-oriented design, service design, design sprint, software development, Agile, SAFe, Scaled Agile Framework.		

SISÄLLYS

1	JOHDANTO	1
2	TUTKIMUKSEN LÄHTÖKOHDAT	2
2.1	Tutkimuksen taustaa	2
2.2	Tutkimustehtäväni ja tutkimuskysymys	2
2.3	Tutkimuksen tavoitteet	3
2.4	Tutkimusmenetelmän valinta	3
2.5	Tutkimukset riskit	4
3	KETTERÄN OHJELMISTOKEHITYKSEN ESITTELY	5
3.1	Ohjelmistokehityksen synty ja kasvu	5
3.2	Agile Manifeston arvot ja periaatteet	6
3.3	Yleisimmät ketterät kehittämisen- menetelmät (Agile) ja käsitteet	8
4	PALVELUMUOTOILUN ESITTELY	13
4.1	Palvelumuotoilu	13
4.2	Käyttäjälähtöinen vs. Käyttäjäkeskeinen suunnittelu	14
4.3	Yleisimmät palvelumuotoilun menetelmät ja käsitteet	15
4.4	Design Sprintin synty	20
5	TUTKIMUKSEN TOTEUTTAMINEN	21
5.1	Aineiston kerääminen ja haastatteluiden suunnittelu ja toteutus	21
5.2	Tutkimusaineiston analyysi	21
6	TUTKIMUSTULOKSET	23
6.1	Yrityksen ohjelmistokehityksen tämänhetkinen tila	23
6.2	Nykytilan kartoituksen tulosten pohdinta ja uuden toimintamallin valinta	25
6.3	Kehitetty konstruktio	26
6.4	Tulokset kehitetyn konstruktion jälkeen	29
7	JOHTOPÄÄTÖKSET JA POHDINTA	31
7.1	Kehitysehdotus	32
7.2	Tutkimuksen pätevyys ja luotettavuus	32
	LÄHTEET	33
	LIITTEET	36

1 JOHDANTO

Suoritin kesällä 2018 Scaled Agile Frameworkin (SAFe) ketterän ohjelmistokehityksen kurssin työpaikallani it-alan konsultointiyrityksessä. Eitellessäni itseni kurssilaisille, tulin maininneeksi opiskelevani palvelumuotoilua erikoistumisopintoina ja tämä herätti yksiköni ohjelmistokehityksen asiantuntijoissa kiinnostusta. Heitä kiinnosti erityisesti, kuinka tätä suhteellisen uutta osaamisalaa voisi hyödyntää paremmin yksikkömme toiminnassa. Opinnäytetyöni aihe oli syntynyt.

Ohjelmistokehitys-organisaatioissa ongelmana on usein, että ohjelmistoja kehitettäessä ei asiakasta oteta huomioon kokonaisvaltaisesti. Samalle asiakkaalle saattaa olla organisaation sisällä useampi projekti käynnissä ja tämä usein aiheuttaa sekaannusta ja väärinymmärrystä asiakkaan ja ohjelmisto-organisaation välillä. Projekteissa valittavan usein myös tavoitteet ovat epäselviä. Vaikka huomattaisiinkin, että projektissa kokonaisuuden kannalta jokin asia kannattaisi tehdä toisella tavalla kuin mitä alun perin suunniteltu, suunnitella harvemmin lähdetään enää muuttamaan. (Auer, Heinäsmäki, Hölttä, Kalliala, Laanti, Laine, Lekman 2013, 26)

Ketterän ohjelmistokehityksen perimmäiset arvot ja periaatteet ovat hyvät. Ketteryys mahdollistaa iteroinnin, eli ohjelmiston toiminnallisuuksien toistamisen, kunnes se palvelee tarkoitustaan sekä nopean palautteen saamista käyttäjiltä. Loppukäyttäjien tulisi siis olla mukana kehitystyössä jo varhaisessa vaiheessa, mutta näin ei usein kuitenkaan ole. Ei siis riitä, että loppukäyttäjää edustaa yrityksen liiketoiminnasta vastaava henkilö, vaan todellinen ymmärrys asiakkaista ja heidän tarpeistaan tarvitaan. (Auer, ym. 2013, 27)

Ketterän ohjelmistokehityksen ja palvelumuotoilun periaatteet ovat mielestäni hyvin samankaltaisia. Palvelumuotoilu yhdistää muun muassa teollisessa muotoilussa ja käyttäytymistieteissäkin käytettyjä ketteriä tutkimusmenetelmiä. Sen tarkoituksena on käyttäjäymmärryksen laajentaminen, uusien palveluiden innovoiminen ja nopeat kokeilut käytännössä.

Opinnäytetyössäni perehdyn aluksi ketterän ohjelmistokehityksen syntyyn, toimintamalleihin ja menetelmiin sivuten myös mallin hyviä ja kehitettäviä puolia. Luvussa 4 käsittelen palvelumuotoilua, käyttäjälähtöisen ja käyttäjäkeskeisen suunnittelun eroja sekä tutkimusmenetelmiä. Nämä aiheet sidon tutkimustyöni tueksi, jossa tutkin yksikkömme ohjelmistokehittämisen nykytilaa ja kuinka palvelumuotoilun menetelmiä voisi hyödyntää paremmin työn laadun parantamiseksi sekä ohjelmistojen laadukkaamman lopputuloksen saamiseksi.

2 TUTKIMUKSEN LÄHTÖKOHDAT

2.1 Tutkimuksen taustaa

Ohjelmistokehityksessä ongelmana usein on, että ohjelmistot suunnitellaan kiireessä pienen kehitystiimin kesken. Mikäli tulevan ohjelmiston tilaajana toimii toinen yritys, on vaativuusmäärittelijöinä yleensä projektipäällikkö ja yrityksen liiketoimintajohtaja. Näiden henkilöiden näkemykset ovat yleensä samankaltaisia, varsinkin jos kyseessä on esimies ja alaissuhde. Harmillisen usein jää pimentoon loppukäyttäjän näkemykset järjestelmävaatimuksista. Tämä kostautuu viimeistään siinä vaiheessa, kun ohjelmistoa otetaan käyttöön. (Juvonen 2018, 58-59.) Tästä esimerkkinä voidaan mainita Helsingin Seudun Liikenteen (HSL) matkakortinlukijoiden uudistus pääkaupunkiseudulla vuonna 2016. Kuukausi käyttöönoton jälkeen Yle teki tutkimuksen laitteen käytöstä, johon vastasi 140 henkilöä ja palaute oli tyrmäävää. Kaikissa vastauksissa matkakortinlukijaa arvosteltiin hitaana ja vaikeana käyttää. Ongelmia oli paljon; se ei toiminut hanskat kädessä, tarvitsi jossain tapauksissa käyttäjänsä molempia käsiä, jolloin ruokakassit piti laskea liikkuvan bussin lattialle. Myös matkakortin arvon tarkistaminenkin oli monen painalluksen päässä. Vaikka vuotta myöhemmin HSL järjesti uusia käyttöttestauksia paremman käyttömukavuuden lisäämiseksi, tuli matkakortinlukijoiden uudistamisprojektin hintalapuksi yli 60 miljoonaa euroa ja ainoa asia minkä käyttäjät muistivat, oli huono käyttökokemus. (Yle Uutiset 2019.)

2.2 Tutkimustehtäväni ja tutkimuskysymys

Tutkimustehtäväni on kehittää kohdeyrityksessä toimivan yksikön ohjelmistokehityksen laatua palvelumuotoilun keinoin sekä analysoida osallistujien kokemuksia ja käsityksiä siitä. Tutkimuskysymykseni on, olisiko palvelumuotoilun menetelmillä mahdollista parantaa kohdeyrityksen ohjelmistokehitysaihioiden tämänhetkistä laatua?

Tällä hetkellä ohjelmistokehitystä kohdeyrityksessä on toteutettu pääsääntöisesti ns. vesiputous- mallilla, joka juurtaa juurensa 1970 -luvulle saakka. Mallissa aluksi määritellään tavoitteet ja tämän jälkeen kehitetään ohjelmistoa tietystä ennalta sovitussa, vesiputouksen lailla valuvassa järjestyksessä. Tämä on haasteellinen malli toimia nykyisessä ympäristössä, jossa käyttäjien tarpeet ja odotukset muuttuvat koko ajan ja näihin muutoksiin ei pystytä tarpeeksi nopeasti reagoimaan. Lisäksi ohjelmistokehityksen kehitysaihiot ovat syntyneet pääsääntöisesti tilaavan yrityksen liiketoimintajohdon tarpeesta tai jonkin toiminnallisuuden puutteesta ohjelmistossa.

2.3 Tutkimuksen tavoitteet

Tutkimuksellani tavoittelen palvelumuotoilun prosessien liittämistä osaksi nykyajan ketterää ohjelmistokehittämistä kohdeyrityksessä ja parantamaan kehittäjien työskentelykäytäntöjä. Lisäksi tuottamaan jatkossa uusia laadukkaampia kehitysaihoita huomioiden liiketoiminnan, loppukäyttäjien sekä teknisen kehityksen näkökulmat.

Tutkimuksella haetaan vastauksia nykytilan haasteisiin ja siihen, onko palvelumuotoilu menetelmillä mahdollisuutta kirkastaa ohjelmistokehittäjien suunnittelemaa kehitysaihoita sekä saada laaja-alaisempi ja laadukkaampi lopputulos kuin aikaisemmin.

2.4 Tutkimusmenetelmän valinta

Uutta toimintamallia ja kehitysehdotuksia laadittaessa käytettiin kvalitatiivista sekä konstruktiiivista tutkimusmenetelmää. Kvalitatiivisessa osassa hyödynnettiin induktiivista päättelyä, jolla pyrittiin keskittymään annettuun asiayhteyteen keräämällä tietoa tietyltä kohderyhmältä ja tekemään tämän pohjalta yleistyksiä. Lähestymistapoina kvalitatiivisessa tutkimuksessa on mm. teorian luonti sekä toimintatapojen kuvaileminen. (Gorman, Clayton, Shep 2005, 4.)

Kvalitatiivisia menetelmiä käytetään usein haastatteluista tai kohdetapauksista muodostuvissa tutkimuksissa keinona vahvistaa ja arvioida tutkimustuloksia laajemmalla mittakaavalla. (Shuttleworth 2008.)

Konstruktiiivinen tutkimus tarkoittaa suunnittelua, mallintamista ja mallien toteutusta sekä testaamista. Konstruktiiivinen tutkimus muistuttaa paljon innovaatioiden tuottamista ja palvelumuotoilua, ja se soveltuu konkreettisiin tuotoksiin kuten mittareiden, mallin tai suunnitelman tekoon. Tavanomainen prosessi toiminta- ja konstruktiiivisessa tutkimuksessa: Ongelma, syvällisen teoreettisen ja käytännöllisen tiedon hankkiminen tutkimuksen ja kehittämisen kohteesta. Ratkaisujen laatiminen, ratkaisun toimivuuden testaus ja konstruktion oikeudellisuuden osoittaminen. Ratkaisussa käytettyjen teoriakytkentöjen näyttäminen ja ratkaisun uutuusarvon osoittaminen. Ratkaisun soveltuvuusalueen laajuuden tarkastelu. Kvalitatiivinen tutkimusmenetelmä soveltuu tähän opinnäytteeseen siinä määrin, että liike-toimintamallin sisältö ja kehitysehdotukset ovat osittain johdettu taustamateriaalista ja kohdeyrityksen edustajien haastattelutuloksista. Konstruktiiivisen tutkimusmenetelmän kriteerit työ saavuttaa esittelemällä opinnäytetyön tekijän toteutetun konseptin, joka kuitenkin pohjautuu suurilta osin teoriaan. (Ojasalo, Moilanen, Ritalahti, 2015.)

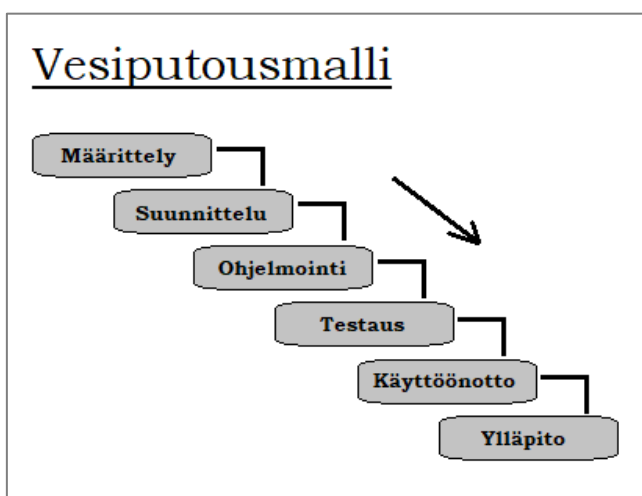
2.5 Tutkimukset riskit

Tutkimuksen riskejä ovat, että ohjelmistokehittäjät eivät innostu uusista menetelmistä ja eivät halua lähteä kokeilemaan uutta tapaa tehdä asioita. Lisäksi riskinä on, että yksittäisten ohjelmistotuotteiden syväosaajat keskittyvät ainoastaan nykyiseen työhönsä ja täten eivät pysty ideoimaan ja käyttämään luovaa ajattelua kyseisen tuotteen ulkopuolelta. Tämän lisäksi työntekijöiden kiire nykyisessä työssä luovat riskin tutkimuksen onnistumiselle.

3 KETTERÄN OHJELMISTOKEHITYKSEN ESITTELY

3.1 Ohjelmistokehityksen synty ja kasvu

1970-luvulla Winston W. Royce julkaisi artikkelin "Managing the development of large software systems". Tällä artikkelilla hän tuli luoneeksi samalla niin sanotun vesiputousmallin (kuva 1), jota ohjelmistokehittäjät käyttävät vielä tänäkin päivänä. Vesiputousmallin tavoitteena on lineaarisesti edetä ohjelmiston kehittämisen eri osa-alueilla. Aloittaen ohjelmistojen vaatimusten määrittämisestä ja analysoinnista. Edeten ohjelmiston suunnitteluun, ohjelmointiin, testauksiin ja toiminnallisuuksien kartoittamiseen sekä dokumentointiin. Ikävä kyllä, Roycen artikkelista jäi vähemmälle huomiolle kohta otsikolla "Riskialtis ja mahdollisuus epäonnistumiselle". Hän ehdottikin artikkelissaan kehitystyön toistojen eli iteraatioiden lisäämistä epäonnistumisriskien pienentämiseksi tai kokonaan poistamiseksi. Näitä ei kuitenkaan koskaan maailmanlaajuisen käyttöönoton jälkeen otettu huomioon. (Girvan, Paul 2017,12.)



Kuva 1. Vesiputousmalli, jossa kehitysprosessi valuu vesiputouksen lailla ylhäältä alaspäin. (Mukailtu Thinkingportfolio 2016)

Ohjelmistojen kehittämisen vaatimusten kasvaessa, lisääntyivät myös erilaiset haasteet vesiputousmallin käyttämisessä. Uusia malleja alkoi syntyä. Yksi suosituimmista oli Rapid Application Development eli RAD. Se syntyi Yhdysvalloissa, alun perin New Yorkin puhelinlaitoksella ja lopulta IBM otti sen käyttöönsä hieman suoraviivaistetumpana versiona. Sen ideana oli ensin tehdä prototyyppi ohjelmistosta, testata sitä ja tämän pohjalta määrittää vaatimukset ja muotoilu sekä kehittää lopulta käytettävä tuote asiakkaille. RAD-mallin käytöstä syntyi myös lisää uusia vaihtoehtoisia malleja, kuten Spiral Model-malli, joka

syntyi Barry Boehmin ja James Martinin yhteistyön tuloksena. Siinä nimensä mukaisesti pyrittiin, jokaisen kehittämisvaiheen kohdalla tekemään ns. spiraalia, kunnes riskit olivat tarpeeksi hyvin selvillä ja ennakoitavissa. Koko 1990-luvun syntyi edellä mainittujen lisäksi monia eri malleja paremman ohjelmistokehittämisen tueksi. Lopulta vuosituhatluvun vaihteessa James Rumbaugh, Grady Brooch ja Ivar Jacobson kehittivät Rational Unified Process (RUP) -mallin, joka koostui useammasta ohjelmistokehitysmallista, mukaan lukien RAD ja Spiral model- mallit. Tämä muodosti rungon tulevalle ketterälle ohjelmistokehitykselle. (Girvan ym. 2017.13.)

Ketterälle kehitykselle ominaisesti projektin aikana tehdään jatkuvaa määrittelyä, kehitystyötä sekä testausta asiakkaan kanssa, mikä takaa huolitellun tuotteen sekä halutun lopputuloksen. Työ tapahtuu iteraatioissa, joiden lopuksi valmis toiminnallisuus lisätään ohjelmistoon. Muutokset ovat tervetulleita ja niitä pystytään ottamaan työn alle nopeasti. Tärkeintä on asiakkaan palaute, jonka mukaan jatkuvaa kehitystä voidaan tehdä tuotteelle. (Kiiskinen 2017.)

Asiakas on aina tervetullut kehittämispalaveriinkin keskustelemaan kehittäjien kanssa ja testaamaan toiminnallisuuksia tuotteessa. Jotkut ovat vieneet asiaa jo niin pitkälle, että ovat liittäneet asiakkaan pysyväksi osaksi kehitystiimiä. Puhuttaessa asiakkaasta yksilönä, piileekin vaara kehitystyön onnistumisen suhteen. Kun ei katsota käyttäjäryhmää vaan ainoastaan yhden käyttäjän tarvetta, pahimmillaan rakennetaan lopulta ohjelmisto, joka ei vastaa tulevien käyttäjiensä todellisia tarpeita. Tehdään siis yhden asiakkaan näköinen ohjelmisto, joka ratkaisee vain kyseisen asiakkaan yksittäisen ongelman laajemman käyttäjäkunnan sijaan. (Meyer 2014. 52-53.)

Kent Beck oli ensimmäisiä ketterän ohjelmistokehityksen pioneereja. Julkaistessaan kirjansa vuonna 1999 ja kutsumalla osapuolet kokoukseen Yhdysvaltain Oregoniin, saivat he yhteisen käsityksen eri ohjelmistokehitysmallien yhdenvertaisuuksista. Samalla he päättivät perustaa Agile Alliance -järjestön, edistämään ketterien ohjelmistokehitysmallien tulevaisuutta. Muutama vuosi myöhemmin, he tapasivat uudestaan. Tämä tapaaminen tapahtui Snowbird -hiihtokeskuksessa Yhdysvaltain Utahissa helmikuussa 2001. Suurin saavutus kokouksella oli niin sanotun ”Agile Manifeston- julistus”. Julistus loi arvot ja pääperiaatteet ketterälle ohjelmistokehittämiselle. (Girvan ym. 2017, 14-15.)

3.2 Agile Manifeston arvot ja periaatteet

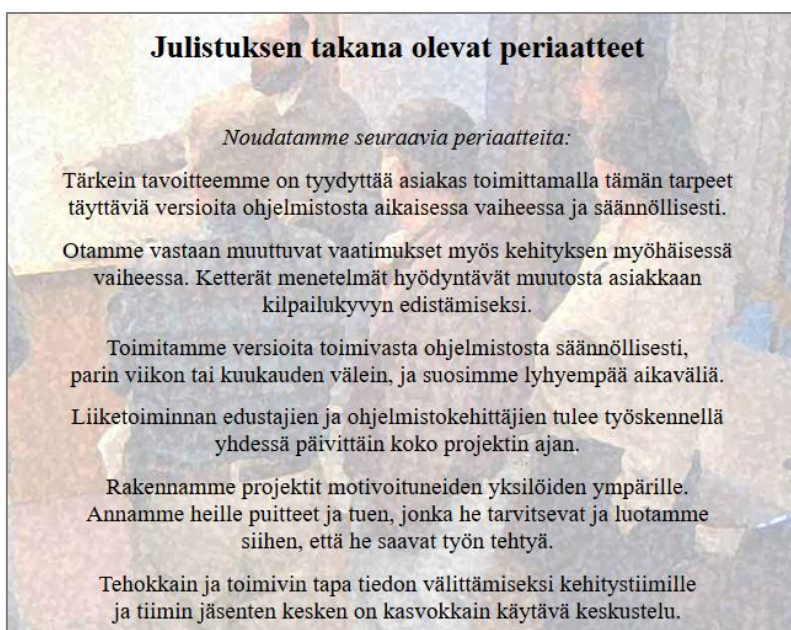
Mitä sitten tarkoitetaan Agilella? Usein sen ajatellaan olevan prosessi, iso määrä periaatteita tai ohjeita, jotkut jopa uskovat sen olevan työkalupakki parempaan ohjelmistokehitykseen. Se ei kuitenkaan ole mitään näistä. Agile on mielentila ja apuväline, joka hyödyntää

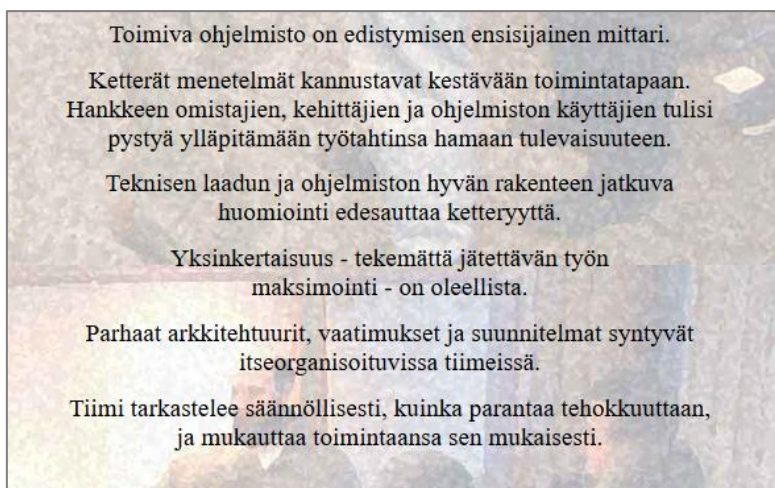
työntekijöiden voimaa ja asiakkaiden palautetta saavuttaen nopeammin menestyksekkään lopputuloksen. Jotta tähän mielentilaan pääsee, on tärkeää sisäistää Agile Manifeston sanoma. (kuva2) (Moreira 2017. 22-23.)



Kuva 2. Agile Manifesto- julistuksen neljä tyypillistä arvoa. (Agilemanifesto.org 2001)

Kuvassa 3 periaatteiden tarkoitus ei ole ainoastaan kuvata kuinka Agile-tiimin pitäisi toimia, vaan niiden tarkoitus on myös vaikuttaa siihen, kuinka jokaisen tiimin jäsenen pitäisi muuttaa ajatustaan sekä käyttäytymistään. (Girvan ym 2017. 20)





Kuva 3. Manifeston kaksitoista periaatetta, jotka luovat viitekehyksen koko tekemiselle. (Agilemanifesto.org 2001)

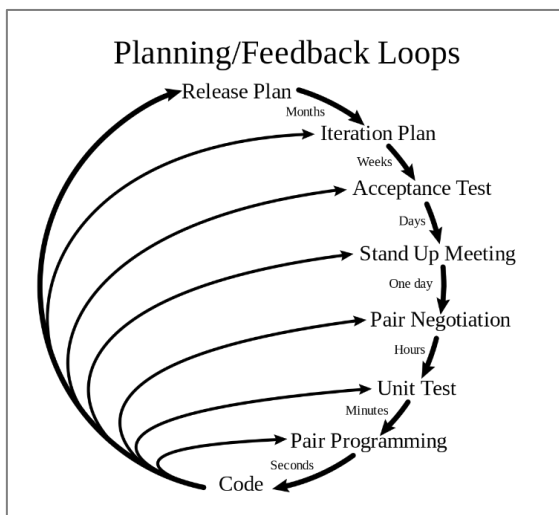
Aikaisempaan ohjelmistokehitykseen verrattaessa, Agile tuo organisaatioihin itseohjautuvat tiimit, joissa kaikki asiantuntijat tiiminä ajavat kohti yhteistä päämäärää. He suunnittelevat myös itse säännöt, kuinka työt saadaan tehdyksi. Valitsevat itse työnsä, jonka haluaa tehdä sekä osallistuvat lopullisiin testauksiin ja muutosten tekemisiin. Agilen ytimessä on myös jatkuva kehittäminen, jossa jokainen tiimiläinen on vastuussa iteraatioiden riittävästä tekemisestä ja virheiden korjaamisesta. Aikaisemmin virheistä vastasi esimies. Tämä jatkuva kehittäminen ja yksilöiden vastuun lisääminen luo työntekijöiden sitoutumista työhönsä ja lopputulokseen. Myös tärkeää on tiheämpi palautteen saanti. Aikaisemmin ohjelmistoa kehitettiin kuukausia, jonka päätteeksi kysyttiin palaute. Agilessa palaute on jatkuvaa alusta loppuun, jolloin korjaaminen matkan varrella on helpompaa ja nopeampaa. Agile pyrkii toiminnallaan myös poistamaan silot organisaation sisältä. Esimerkiksi ohjelmistojen kehittäjät ja testaajat ovat yhtä samaa tiimiä. Tällöin ei synny ”me vastaan nuo”-ongelmaa ja kaikki vastaavat sekä aikataulusta kuin ohjelmiston virheettömyydestä. Lisäksi kun kaikki asiantuntijat työskentelevät samassa fyysisessä tilassa, on heidän helpompaa kommunikoida kasvotusten, tehdä tarvittavia korjauksia sekä luoda uusia ideoita nopeasti. (Ashmore, Runyan 2015. 17-21.)

3.3 Yleisimmät ketterät kehittämisen- menetelmät (Agile) ja käsitteet

Extreme Programming (XP)

XP on ohjelmistokehitys-malli (kuva 4), joka sai alkunsa Chrysler Motorsin palkkahallinnon järjestelmä uudistuksen yhteydessä vuonna 1996. Kent Beck pyrki löytämään menetelmän tiimilleen, joka aikaansaisi laadukasta koodia normaalia nopeammin ja tehokkaammin sekä tuottaisi laadukkaamman lopputuloksen. Mallissa hyödynnetään pariohjelmointia,

jossa nimensä mukaisesti ohjelmoidaan pareittain. Tämä lisää ohjelmoijien parempaa keskinäistä kommunikointia ja motivaatiota. Lisäksi malli hyödyntää testivetoista kehittämistä, jossa vähimmäiskoodia testataan, kunnes se vastaa tarpeeseen ja tämän jälkeen vasta aloitetaan laajempi ohjelmointiprosessi. (Ashmore ym 2015. 50-51.)

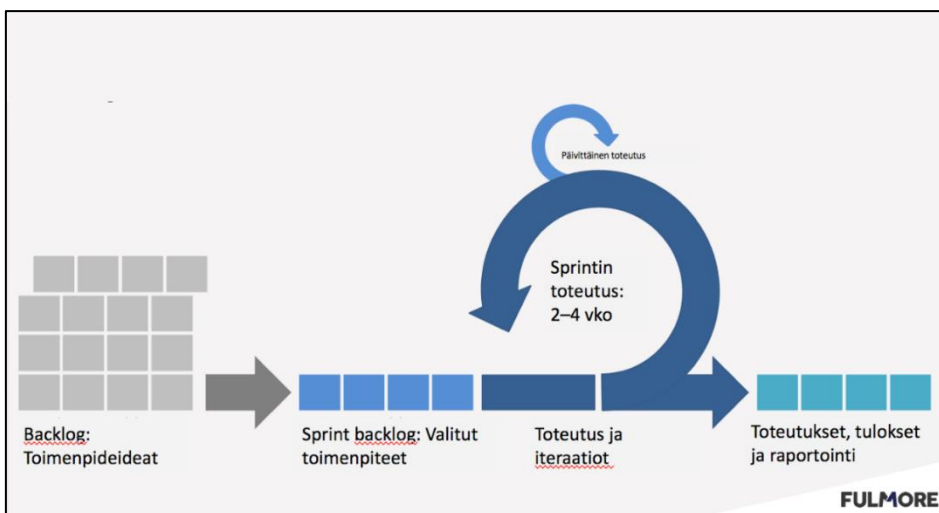


Kuva 4. XP toiminnan kuvaus (Wells 2001)

Mallissa käytetään myös työmäärää vähentävää KISS-periaatetta (Keep It Simple, Stupid!), jossa ohjelmisto pyritään pitämään mahdollisimman yksinkertaisena, jolloin kaikki ylimääräinen työ vähenee ja kehittämisestä tulee tehokkaampaa. (Ashmore ym 2015. 50-51.)

Scrum-malli

Scrum on Jeff Sutherlandin kehittämä malli vuodelta 1993 mutta virallisesti julkaistiin vasta vuonna 1995. XP:n ollessa enemmän tekninen ja kehittäjille suunnattu – Scrum on enemmän projektien ja tiimin johtamiseen sekä aikatauluttamiseen suunnattu malli (kuva 5). Scrum tulee Rugby-termistä, jossa pelin aloituksessa molemmat joukkueet nojaavat tiiviisti vastakkain toisiinsa. He ovat siis tiiviisti yhdessä, koko ajan kommunikoiden. Tiimissä ei voi olla yhtä supertähteä vaan kaikkia tarvitaan parhaan lopputuloksen saavuttamiseksi. (Ashmore ym. 2015.54-55.)

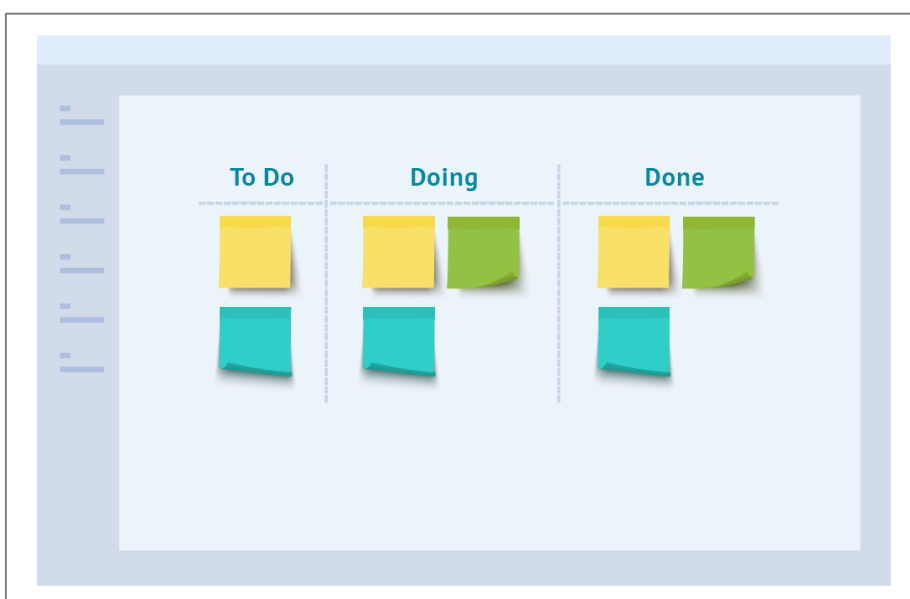


Kuva 5. Scrum- malli. (Fulmore 2019)

Scrum koostuu 2-4 viikon Sprinteistä, jonka sisällä tuotetaan sovittu ohjelmiston toiminnallisuus. Kehitystyö koostuu suunnittelu- ja päiväpalavereista, joissa suunnitellaan kehitystyön sisältö. Palavereissa on tarkka kaava ja ne pidetään hyvin tehokkaina. Tuotteen kehitysjonoihin lisätään työmäärää, kun edellinen työ on valmis. Sprinttikatselmus ja Sprintin retrospektiivi ovat palavereita, jossa tarkastellaan tuotteen ja tiimin työskentelyn onnistumista sekä tarvittaessa muokataan tuotetta ennen seuraavaa sprinttiä. (Ashmore ym. 2015.54-55.)

Kanban-malli

Kanban-mallin kehitti David J. Anderson, (kuva 6) joka piti tärkeänä asiana saada ohjelmistotuotteet valmiiksi sovitussa aikataulussa. (Ashmore ym. 2015. 60-61)

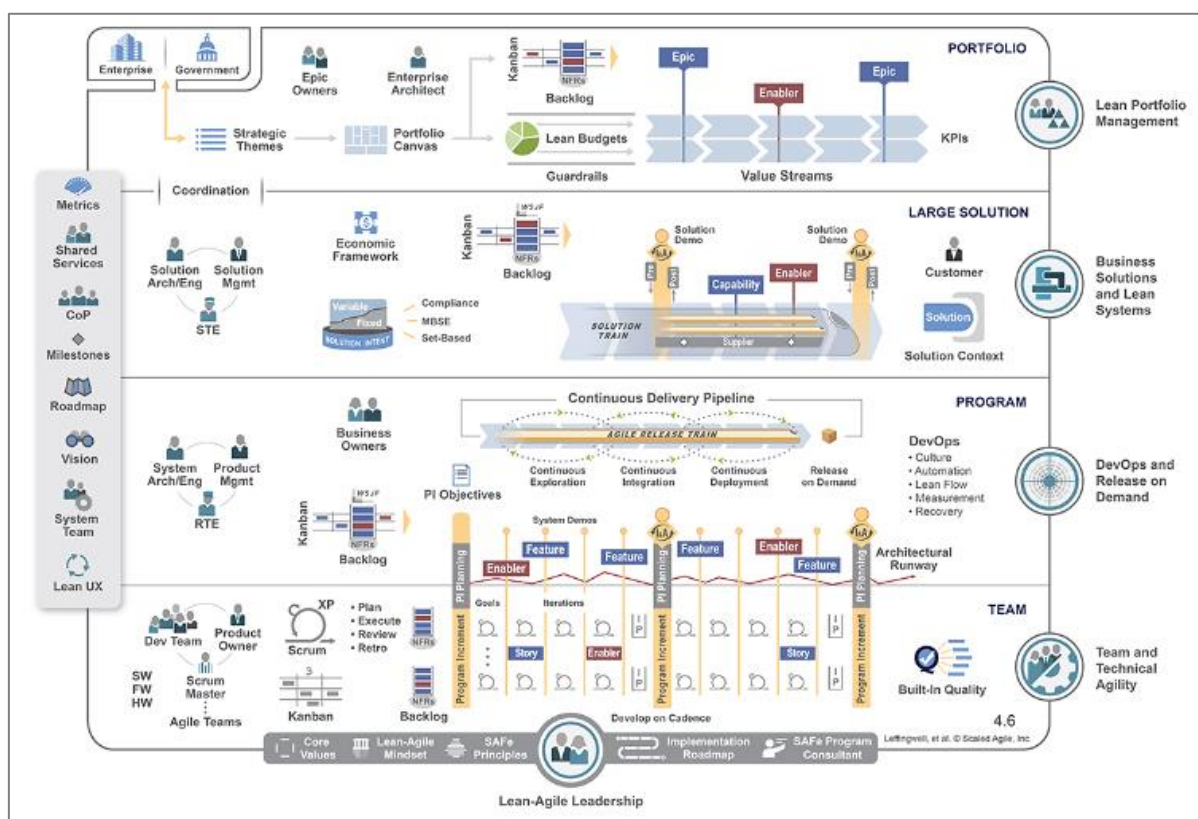


Kuva 6. Kanban-taulu. (Digite 2019)

Mallissa hyödynnetään taulua, johon merkitään tehtävät, joita pitää tehdä (eng. To Do). Tehtävät, jotka ovat työn alla (eng. Doing) ja tehtävät, jotka on suoritettu (eng. Done). (Meyer 2014, 136.) Tämä helpottaa ohjelmistoprojektien hallintaa ja prosessien seuraamista aikataulussaan. (Ashmore ym. 2015, 60-61.)

SAFe- malli (Scaled Agile Framework)

SAFe on skaalautuva, ohjelmistokehityksessä käytetty viitekehys -malli, joka on suunniteltu suurienkin organisaatioiden tarpeisiin. Sen kehitti Dean Leffingwell vuonna 2008 ja ensimmäinen versio siitä julkaistiin vuonna 2011. SAFe yhdistää Agile-ajattelun, XP:n, Scrumin ja Kanbanin sekä muut ketterän kehityksen periaatteet. Viitekehys kertoo ohjailevasti sitä käyttävälle organisaatiolle, mitä pitää tehdä. (kuva 7) Erityisesti mallissa halutaan tuoda esille, että ihmiset, heidän osaamisensa sekä tieto on tärkeämpää kuin käytetty teknologia. (Powell-Morse 2017) Safen ydinarvoja on sisäänrakennettu laadun varmistaminen, tuloshakuisuus, läpinäkyvyys sekä tekemisen linjaaminen liiketoiminnallisiin tavoitteisiin. (Harle 2019)



Kuva 7. SAFe – viitekehys. (Scaled Agile Framework 2019)

Kehyksen ylimmällä Portfolio- tasolla organisaatio määrittää *strategiset teemansa*, jotka ohjaavat koko ohjelmistokehitystyön tekemistä. Portfoliotaso määrittää myös tarvittavat roolit, periaatteet ja käytännöt kehitysaihioiden määrittelyyn, budjetointiin ja hallintaan.

Kehitysaihiot (eng. Epic) ovat merkittäviä aloitteita, jotka kulkevat arvovirroissa ja joiden pohjalta lähdetään kehittämään tulevia ohjelmistoja. Ne vaativat usein isoja investointeja ja vaikuttavat pitkälle tulevaisuuteen. Kehitysaihiot vaativat määrittelyn sekä kustannuksen, vaikutuksen ja mahdollisuuksien analysoimista, joka tehdään kevyellä liiketoimintakuvausella. Ne vaativat myös yrityksen johdon taloudellisen hyväksynnän ennen toimeenpanoa. Kehitysaihoita on kahdenlaisia, liiketoiminnallisia tai näiden mahdollistajia.

Safen ytimessä kulkee *Toimitusjunat* (eng. Agile Release Train, ART), joissa rakentuu itse ohjelmistot. Yhdessä junassa on tyypillisesti 5-12 kehitystiimiä eli noin 50 -120 henkilöä. (Nitor 2018.) Juna lähtee liikkeelle ohjelmistolisäyksen eli inkrementin suunnittelutapahtumalla eli Program Increment Planningillä. Tapahtumassa toimitusjunan tiimit linjaavat yhteisen vision ja suunnitelman ohjelmiston kehitystyölle. Junassa he testaavat iteraatioissa 2 viikon välein ohjelmiston eri toiminnallisuuksia sekä kehittävät lopulta valmiin ohjelmiston noin 10-12 viikon aikana. (Nitor 2018.)

4 PALVELUMUOTOILUN ESITTELY

4.1 Palvelumuotoilu

Palvelumuotoilun (eng. Service Design) tehtävänä on auttaa organisaatioita havaitsemaan palveluiden mahdollisuudet omassa liiketoiminnassaan, innovoimaan uusia palveluita ja kehittämään jo olemassa olevaa. Se on osaamisala, joka tuo teollisesta muotoilusta tutut menetelmät palveluiden kehitykseen ja yhdistää ne perinteisiin palvelun kehityksen menetelmiin. Palvelumuotoilu siis yhdistää vanhoja asioita uudella tavalla. Se ei ole mitään abstraktia vaan konkreettista tekemistä, joka tuo yhteen käyttäjien tarpeet ja odotukset sekä palveluntuottajan liiketoiminnalliset tavoitteet toimivina palveluina. (Tuulaniemi 2011, 24-25.)

Palvelumuotoilussa tarvitaan analyyttistä ja loogista päättelykykyä, keskittymällä konkreettisiin asioihin, käyttäjätutkimukseen ja dataan. Lisäksi tulee olla intuitiivinen taito nähdä, mikä tulevaisuudessa voisi olla mahdollista, mitä ei vielä ole olemassakaan. (Tuulaniemi 2011, 10-11.)

Ihminen on keskeinen osa palveluketjua. Palvelumuotoilun termein se tarkoittaa asiakkaan palvelukokemusta. Kokemusta sinällään ei pystytä suunnittelemaan, koska se on aina subjektiivinen. Palvelumuotoilulla pyritäänkin optimoimaan tuo kokemus keskittymällä kokemuksen kriittisiin pisteisiin; palveluprosesseihin, työtappoihin, tiloihin ja vuorovaikutukseen sekä poistamalla kokemusta häiritsevät elementit. (Tuulaniemi 2011, 26-27.)

Palvelumuotoilun periaatteita on kuusi;

- 1.) Käyttäjälähtöisyys. Palvelut pitää kokea aina käyttäjän näkökulmasta. Käyttäjälähtöisyyden sijaan voidaan puhua myös ihmislähtöisestä suunnittelusta. Tällöin huomio ei kiinnity vain käyttäjiin tai asiakkaisiin vaan kaikkiin, jotka ovat mukana; palveluntuottajat työntekijöineen, sidosryhmät ja vaikkapa ei-asiakkaat, joihin palvelun käyttäminen kuitenkin välillisesti vaikuttaa.
- 2.) Yhdessä tekeminen. Kaikki osapuolet osallistuvat palvelumuotoilun prosessiin. Koska palvelut ovat tehty käyttäjiään varten, täytyy käyttäjä osallistuttaa myös suunnittelutyöhön. Tiimien tulee olla yhteistyökykyisiä, jossa on monialaisesti eri työ- ja koulutustaustoiltaan olevia henkilöitä. Palvelumuotoilun tarkoitus on hajottaa hierarkiat ja siilot.
- 3.) Jatkuva kehittäminen. Palvelumuotoilu on tutkiva, mukautuva ja kokeellinen lähestymistapa, jossa jatkuva kehittäminen vielä lopulta tavoitteeseen.

- 4.) Havainnollistaminen. Palvelukokemus pitää pystyä visualisoimaan. Palvelupolku on yleisin tapa visualisoida koko palvelukokemuksen elinkaari alkupisteestä päätökseen.
- 5.) Todentaminen. Käyttäjän tarpeet olisi löydettävä todellisista kokemuksista ja ideat pystyttävä mallintamaan. Aineettomatkin palvelut pitää pystyä visualisoimaan fyysisten tavoin.
- 6.) Kokonaisvaltainen suunnitteluprosessi. Palveluiden olisi kestävästi vastattava kaikkien sidosryhmien tarpeisiin, ottaen huomioon palvelun koko elinkaari sekä liiketoiminnalliset tavoitteet. (Stickdorn, Hormess, Lowrence, Schneider 2018, 25-27.)

On havaittu, että tuotteiden erilaistuminen vähenee toimialojen kypsyessä, jolloin yritykset pystyvät erottautumaan massasta vain luomalla parempia palvelukokemuksia asiakkailleen. Palveluiden parantamisen etuina on myös asiakkaiden saama parempi palvelukokemus ja sitä kautta kasvava uskollisuus yritystä kohtaan. (Reason, Lovlie, Brand, Flu 2016, 2.) Sosiaalisissa trendeissä näkyvät kuluttajien muutokset. Aikaisemmin kuluttajat hyväksyivät sen mitä on tarjolla, nykyään he vaativat koko ajan parempaa palvelukokemusta. Myös työpaikoilla on tapahtunut muutos. Työntekijät ovat olleet tottuneet tähän asti siihen, että heidän työntekijäkokemuksensa, niin hyvässä kuin pahassakin on sitä mitä on ja kuuluu osaksi työtä. Jatkossa työntekijöiden odotukset paremmasta kokemuksesta ovat kasvussa myös työpaikoilla. (Reason ym. 2016, 3.)

Digitalisaation murroksen trendit ovat jo näkyneet vuosia. Esimerkiksi finanssialalla, kasvotusten tapahtuvien asiakastapaamisten määrä pienenee koko ajan, asiakkaiden siirtyessä verkko- ja mobiilipankin käyttäjiksi. Palvelun muuttuminen ihmisen toimittamasta palvelusta täysin digitaaliseksi, luo tarpeen palvelumuotoilulle. Palvelumuotoilu tarjoaa työkalut teknologian inhimillistämiseen luoden sitä kautta paremman palvelukokemuksen. (Reason ym. 2016, 4.)

4.2 Käyttäjälähtöinen vs. Käyttäjäkeskeinen suunnittelu

Monessa yhteydessä sekoitetaan usein keskenään, ohjelmistokehityksessä jo pidemmän aikaa käytössä ollut *käyttäjäkeskeinen suunnittelu* ja palvelumuotoilun terminologiassa käytetty *käyttäjälähtöinen suunnittelu*.

Käyttäjäkeskeinen suunnittelu lähtee teknologian ja kaupallisuuden ehdoista. Jos palvelua on vaikea käyttää, sitä on vaikea myydä. Käyttäjäkeskeisessä suunnittelussa käyttäjät osallistutetaan palvelun testaajiksi, kun tuote on jo prototyyppi asteella. Käyttäjät

aiheuttavat toimillaan häiriötä ja näin synnyttävät virhetiloja palvelussa. Näitä virheitä pyritään eliminoimaan mahdollisimman tehokkaasti. (Keinonen 2018.)

Käyttäjälähtöisellä suunnittelulla tarkoitetaan tuotteen tai palvelun tekemistä vuorovaikutuksessa käyttäjien kanssa koko projektin elinkaaren yli, asiakkaiden tarpeen kautta syntyneestä ideasta, sen kehittämiseen, nopeaan testaukseen ja lopulta tuotteen lanseeraukseen. Kehitystyötä voidaan tehdä monella eri tapaa, esimerkiksi perinteisesti haastatteleamalla tai havainnoimalla käyttäjiä. Kun on jo vastaavanlaisia palveluita käytössä, käyttäjät nopeasti huomaavat asioita, jotka eivät toimi. Jos lähdetään tekemään jotain aivan uutta, mennään sinne missä ihmiset käyttävät palveluita ja selvittämään mitkä ovat heidän piilevät tarpeensa ja mitä he oikeasti näkevät, tekevät ja aistivat. (Keinonen 2018.)

4.3 Yleisimmät palvelumuotoilun menetelmät ja käsitteet

Palvelumuotoilussa kaikki alkaa projektin tavoitteen tai toimeksiannon määrittelyllä eli briiffillä. Yleensä briiffin toimittaa palvelumuotoilutyön tilaajana oleva yritys, mutta joskus uutta innovoitaessa, voi se muodostua vasta tarkempien tutkimusten ja havainnointien tuloksena. Sitä voidaan siis tarvittaessa muokata koko projektin ajan. Hyvässä briiffissä tulee olla ajateltuna kohderyhmä, mihin tarpeeseen palvelun halutaan vastaavan ja mitkä ovat yrityksen liiketoiminnalliset tavoitteet. (Tuulaniemi 2011,132.)

Esitutkimuksissa suunnitteluryhmällä on myös hyvä olla myös selkeä kuva yrityksestä, jolle työ tehdään. Selvitettäviä asioita on yrityksen strategia ja visio sekä tuotteet, toimiala sekä kilpailutilanne. Tutkimusvaiheessa suunnittelijaryhmä hyödyntää useasti Benchmark-menetelmää, tutkien jo samalla alalla olemassa olevia yrityksiä, heidän strategiaansa, palveluita ja toimintatapojaan. Näin he saavat muodostettua paremman kuvan toimintaympäristöstään sekä oppivat parhaita käytäntöjä alalla. (Tuulaniemi 2011, 136-138.)

Koska palvelumuotoilussa hyödynnetään hyvin paljon visuaalista viestintää, käsitekartta eli Mindmap (kuva 8) on oiva keino jo heti alussa tuoda esille projektiin liittyviä käsitteitä ja aiheita. (Tuulaniemi 2011, 140.)



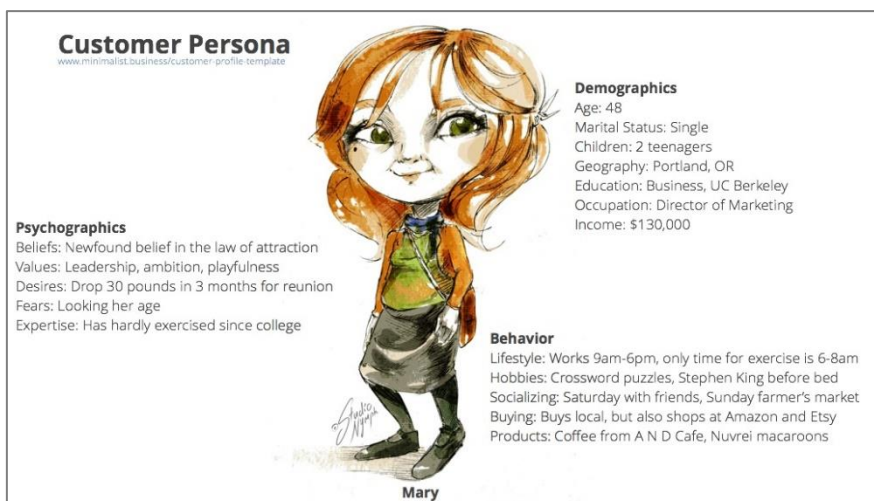
Kuva 8. Mind map eli käsitekartta. (Gröpl 2009)

Palvelumuotoilussa hyödynnetään sekä laadullista, että määrällistä tutkimusmenetelmää. Laadulliset menetelmät sopivat paremmin tuotetta tai palvelua suunniteltaessa ja määrälliset lopullisen tuotteen tai palvelun arvioimista varten. (Tuulaniemi 2011, 144.)

”Empatia eli aito ja syvä kiinnostus kohderyhmään on yksi palvelumuotoilun keskeisempiä asioita” (Tuulaniemi 2011, 144.)

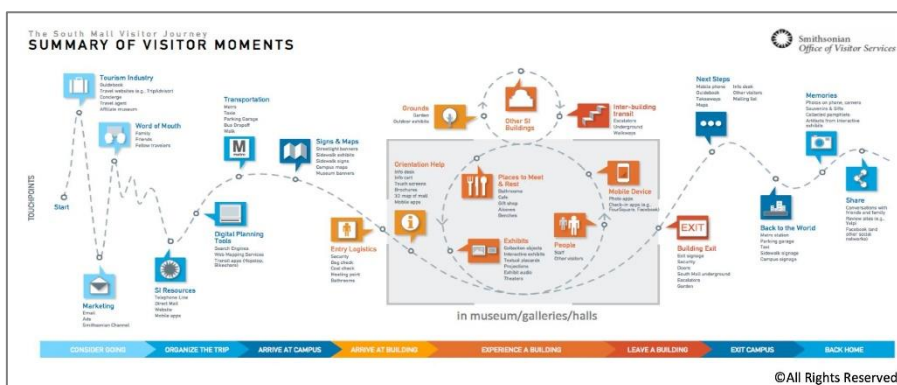
Asiakasymmärrystä lisätään tutkimalla jo olemassa olevia taustatietoja sekä haastattelemalla ja havainnoimalla käyttäjiä. Myös Luotain-menetelmä toimii hyvin ymmärryksen lisäämisessä. Siinä annetaan käyttäjille itsedokumentointivälineet, ja näiden avulla saadaan esiin tärkeää hiljaista tietoa käyttäjästä. Näitä ihmisen kuvaamiseen tarkoitettuja menetelmiä kutsutaan etnografiseksi tyyliksi ja sen perinteet löytyvät antropologiasta ja sosiologiaaiteista. (Tuulaniemi 2011, 146.)

Kun tietoa käyttäjistä on kertynyt riittävästi, on helpointa kiteyttää tietoa rakentamalla erilaisia käyttäjäpersoonia, käyttäjätarinoita ja palvelupolkuja palvelupisteineen. (Kuvat 9 ja 10)



Kuva 9. Käyttäjäpersoonaa, joka edustaa tiettyä asiakaskuntaa. (Minimalist 2019)

Nämä profiilit, tarinat tai palvelupolut ovat kuvitteellisia, jotka rakentuvat tietystä kohderyhmästä, tietyssä ajanjaksona kerätyn tiedon pohjalta. Tämä on hyvin tärkeää tietoa, koska yritys, joka pyrkii tuottamaan paremman asiakaskokemuksen, on ymmärrettävä asiakaidensa arvomuodostus ja käyttäytyminen. (Tuulaniemi 2011, 154-155)



Kuva 10. Museossa vierailevan henkilön Palvelupolku. (Toptal 2019)

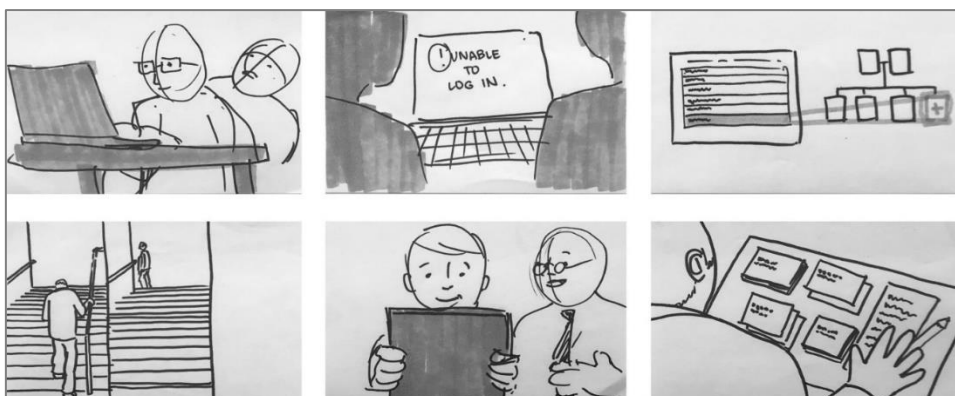
Kun profiileista, tarinoista ja poluista on löydetty jokin käyttäjän piilevä tarve, voidaan jatkossa määrittää projektille suunnitteluajurit. Ne nimensä mukaisesti ajavat suunnittelu-työtä. Esimerkiksi tutkimuksissa nousee esille, että käyttäjät eivät osaa käyttää uutta järjestelmää ensimmäisellä kerralla ja siihen kuluu normaalia enemmän aikaa. Suunnittelu-ajurina voi silloin olla; järjestä perehdytystilaisuus uusille käyttäjille palveluun tutustumiseksi. (Tuulaniemi 2011, 156-157.)

Tämän jälkeen on ideoimisen aika. Ideoinnin tarkoitus on luoda pohjaa tuleville palvelukonsepteille. Ideointi on laajenevaa, jolloin ideoita syntyy iso määrä ja hulluimmatkin ideat otetaan huomioon. Ideoinnin päätteeksi lähdetään supistamaan, jolloin luovutaan tai

yhdistellään ideoita saaden toteuttamiskelpoisia palveluaihioita. (Tuulaniemi 2011, 180.) Tätä menetelmää kutsutaan Double Diamond- menetelmäksi. Ideoinnin aloitusmenetelmiä on monia; Voidaan tehdä nykyisestä palvelupolusta tulevaisuuden palvelupolku. Miettiä, kuinka voisimme tehdä tämän toisin ja tällä tavoin löytää vaihtoehtoisia polkuja nykyiseen palvelupolkuun. (Stickdorn ym 2018, 178-179.)

Jotta ideoita saadaan syntymään paljon, on hyvä hyödyntää ryhmätyöskentelyä työpaikoissa (eng. Workshop). Pajoissa voidaan pitää aivoriihiä ja kirjoitusriihiä tai vaikkapa kehoriihiä, luoden nimensä mukaisesti uusia ideoita palveluun kirjoittamalla tai vaikkapa kehoa hyväksi käyttäen. (Tuulaniemi 2011, 189.)

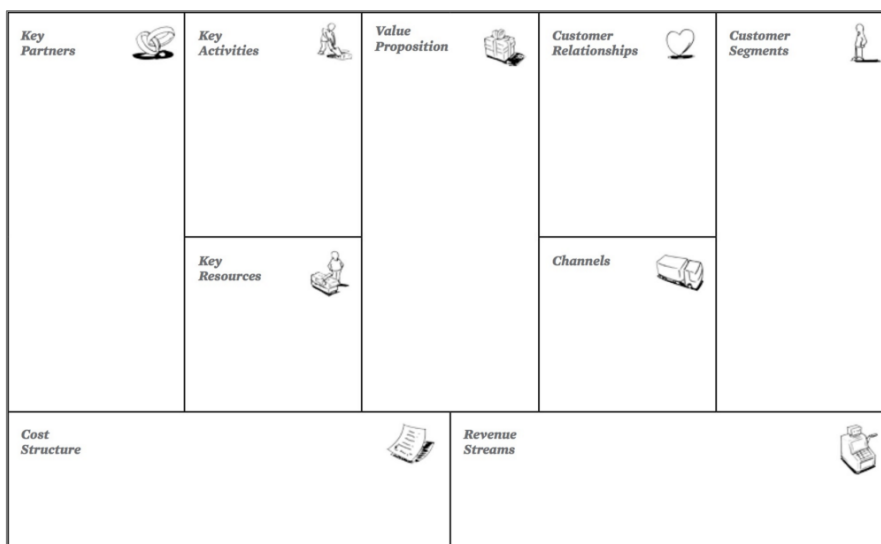
Myös abstraktien asioiden konkretisoiminen on yksi tärkeä osaamisalue palvelumuotoilussa. Vaikea asia voidaan pukea visuaaliseen asuun käyttämällä vaikkapa legoja palvelutilan hahmottamiseen, kuvia tarinan kerrontaan (kuva 11) tai mielikuvien ilmaisemiseen. Myös animaatiot ja videot ovat suosittuja. (Tuulaniemi 2011, 189.)



Kuva 11. Tarinan kertomista sarjakuvin. (Hodson 2018)

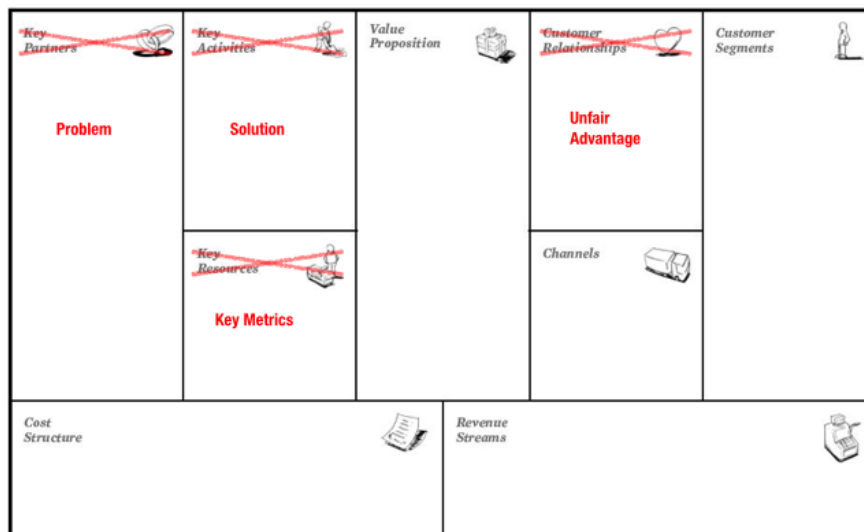
Kun palvelun aihio on muodostunut, siirrytään konseptoinnin vaiheeseen. Konsepti on iso kokonaiskuva, johon sisällytetään palvelun keskeinen idea. Se ei ole yksityiskohtainen kuvaus, vaan tarina palvelun suurista linjoista. (Tuulaniemi 2011, 189.)

Konseptoinnissa hyödynnetään usein isomman organisaatioille suunniteltua Business Model Canvasia (kuva 12) tai Startup-yrityksille ja yksittäisille tuotteille tai palveluille tarkoitettua Lean Model Canvasia (kuva 13). Business Model Canvas on alkujaan Alexander Osterwalderin kehittämä liiketoimintamallin kuvaustapa. Lean Model Canvas sisältää myös 9 elementtiä, mutta on suunniteltu enemmän pienemmille yrityksille tai yksittäisen tuotteen konseptointiin. Len Model Canvasin kehitti Business Model Canvasin pohjalta Ash Maurya.



Kuva 12. Business Model Canvas, jossa 9 eri liiketoiminnallista aluetta. (Smartinsights 2019)

Business Model Canvas kuvaa lyhyesti ja selkeästi, mitä organisaatio tekee ja millaisia palveluita se tuottaa antamansa arvolupauksen mukaisesti. Mallissa kuvatut 9 elementtiä ovat hyödyllinen lisä kokonaisarkkitehtuurin strategialähtöiseen kuvaamiseen ja suunnitteluun.



Kuva 13. Lean Model Canvas, jossa muutama alue korvattu vastaamaan paremmin pienempien yritysten ja yksittäisten tuotteiden ja palveluiden tarpeeseen. (Maurya 2012)

Kun konsepti on suunniteltu, voidaan testata sen toimivuutta prototyypillä. Prototyypinnillä tarkoitetaan tuotteen tai palvelun nopeaa testausta. Se on ensimmäinen versio, joka on nopeasti ja edullisesti mahdollista toteuttaa ja testata toimiiko tuote tai palvelu niin kuin oli

ajateltu ja suunniteltu. Mukaan osallistutetaan käyttäjiä ja palveluntuottajia. (Tuulaniemi 2011. 194-195.)

Prototyypin jälkeen palvelu tuotteistetaan ja lanseerataan käyttöön. Palvelussa tai tuotteessa otetaan vielä huomioon kaikki liiketoiminnalliset asiat kuten, hinnoittelu, kannattavuus, kilpailutilanne sekä markkinointi. Luodun palvelun kehittäminen on jatkuvaa ja sitä seurataan erilaisin mittarein. (Tuulaniemi 2011. 226-227.)

4.4 Design Sprintin synty

Jake Knapp kehitti palvelumuotoilun menetelmistä nopean Design Sprintin Yhdysvalloissa vuonna 2011 työskennellessään Googlella. Idea tähän syntyi hänen pitäessään innovoivia työpajoja kollegoilleen. Ensimmäinen havainto oli, että kaikilla oli kiire ja tästä syystä eivät kerkeä osallistumaan innovointiin ja suunnitteluun tarpeeksi. Lopulta työkollegan kysyessä häneltä:” Kuinka tiedät, että nämä innovoivat työpajat edes toimivat?” Heräsi ajatus kehittää malli, jonka lopputuloksena olisi jotain konkreettista. (Knapp,Zeratsky,Kowitz 2016, 1-2.)

Syntyi Design Sprint, mikä on viiden päivän prosessi, jolla haetaan vastauksia kriittisiin kysymyksiin nopealla prototypoinnilla ja testaamalla ideaa asiakkaiden kanssa. Prosessi sisältää parhaat käytännöt liiketoiminnan suunnittelusta, innovoinnista, käyttäytymistieteistä ja palveluiden muotoilusta, joka on konseptoitu selkeäksi prosessiksi. (Knapp ym. 2016. 9.)

Design Sprintin on tarkoitus auttaa tilanteissa, jossa halutaan testata nopeasti ja halvalla palvelua ennen kuin investoidaan isompi summa rahaa pidempikieliseen kehitystyöhön. Se auttaa nopeutensa takia myös tilanteissa, jossa aikaa on vähän käytettävissä. Myös tiettyjen projektien ollessa jumissa, Design Sprint auttaa selkeyttämään yhteistä visiota ja suuntaa sekä viemään projektia eteenpäin. (Knapp ym. 2016. 26.)

5 TUTKIMUKSEN TOTEUTTAMINEN

5.1 Aineiston kerääminen ja haastatteluiden suunnittelu ja toteutus

Tutkimuksella haetaan vastauksia kohdeyrityksen nykytilan haasteisiin ja siihen, onko palvelumuotoilumenetelmillä mahdollisuutta kirkastaa ohjelmistokehittäjien suunnittelemaa kehitysaihiota sekä saada laaja-alaisempi ja laadukkaampi lopputulos kuin aikaisemmin?

Olennaista oli saada ymmärrys kohdeyrityksen nykytilasta sekä toimintamallin muutoksen jälkeisestä tilasta. Aineisto kerättiin haastattelemalla teemaan liittyen ketterän ohjelmistokehittämisen kolmea asiantuntijaa. Käytetään heistä haastattelussa nimitystä Hlö A, Hlö B ja Hlö C. Osa heistä toimii ohjelmistokehittäjinä ja osa menetelmän parissa työskentelevinä tuoteomistajina. Haastattelut olivat puolistrukturoituja ja ne tehtiin kasvotusten toukuussa 2019 kohdeyrityksen omissa tiloissa.

Kysymykset nykytilasta:

- 1) Minkälaisena näet liiketoiminnan – palvelumuotoilun – teknisen ohjelmistokehityksen yhteistyön tällä hetkellä?
- 2) Minkälaisena näet ohjelmistojen kehitysaihioiden laadun tällä hetkellä?
- 3) Hyödynnetäänkö loppukäyttäjiä kehitystyössä mielestäsi tarpeeksi?
- 4) Kuinka tätä voitaisiin kehittää?

Uuden toimintamallin hyödyntämisen jälkeen:

- 1.) Saatiinko uudella toimintamallilla aikaisempaa parempia lopputuloksia? Jos saatiin niin minkälaisia?
- 2.) Paraniko yhteistyö mielestänne liiketoiminnan– palvelumuotoilun ja teknisen ohjelmistokehityksen osalta?
- 3.) Kuinka tätä voitaisiin vielä kehittää?

5.2 Tutkimusaineiston analyysi

Laadullisen aineiston analyysi alkaa useimmiten haastattelulla, jossa tehdään havaintoja vastausten toistuvuuden, jakautumisen ja erityisvastausten perusteella. Aineistoa analysoidaan lähellä aineistoa ja sen kontekstia. Kvalitatiivinen tutkimus eroaa kvantitatiivisesta tutkimuksesta siinä, että laadullinen tutkimus säilyttää aineistonsa yleensä sanallisessa muodossa. (Hirsijärvi & Hurme 2000, 136).

Siihen miten laadullista aineistoa analysoidaan, ei ole mitään valmista kaavaa. Tutkijalla ei ole käytössään valmista luokittelurunkoa tai teoreettista käsitteistöä. Tämän vuoksi olennaisia asioita pyritään etsimään aineistosta käsin (Hirsijärvi & Hurme 2000, 157). Kun tekstiä on luettu ja tutkittu useampaan kertaan, vastauksia aletaan ryhmittelemään sopiviin kategorioihin. Tässä tutkimuksessa tutkimusaineiston pääkategoriat ovat:

1. Yrityksen ohjelmistokehitysaihioiden tämänhetkinen tila.
2. Tulokset kehitetyn konstruktion jälkeen.

6 TUTKIMUSTULOKSET

6.1 Yrityksen ohjelmistokehityksen tämänhetkinen tila

Ensimmäinen kategoria käsittelee yrityksen ohjelmistokehityksessä syntyvien kehitysaihioiden tämänhetkistä tilaa. Kyseisten asioiden selvittäminen auttoi kirkastamaan ja saamaan kokonaiskuvaa yrityksen kehittämismallien toiminnasta ja siitä millaisessa tilassa ne tällä hetkellä ovat. Yrityksessä haastatellut henkilöt kuvailivat kehittämisen nykytilaa hyvin yleisluonnollisesti ja oman työnsä näkökulmista.

Haastateltava Hlö A on työskennellyt IT-arkkitehtinä yrityksessä noin kuusi vuotta ja ketterän ohjelmistokehittämisen parissa noin kolme vuotta. Haastateltava Hlö B on työskennellyt yrityksessä IT-arkkitehtinä noin viisi vuotta ja ketterän ohjelmistokehityksen parissa melkein saman verran. Hän ei ole suoraan ole liiketoiminnan edustajiin yhteydessä työskennellessään asiakasyritykselle. Asiakkaalla on oma IT-yksikkönsä, joka kommunikoi oman liiketoimintansa kanssa. Haastateltava Hlö C on työskennellyt ketterien ohjelmistokehitysmallien (SAFe) kouluttajana koko yrityksessä työskentelyaikansa, eli noin viisi vuotta.

1) Minkälaisena näet Liiketoiminnan – Palvelumuotoilun – Teknisen ohjelmistokehityksen yhteistyön tällä hetkellä?

Hlö A näkee liiketoiminnan – palvelumuotoilun ja teknisen ohjelmistokehityksen janana, jossa palvelumuotoilu voisi toimia ns. liimana tai tulkkina liiketoiminnan ja teknisen ohjelmistokehityksen välillä. Tulkkia hänen mielestään tarvitaan, koska tällä hetkellä yhteisymmärrystä ei löydy liiketoiminnan ja ohjelmistokehittäjien välillä.

Hlö B:n mukaan palvelumuotoilun rooli on luoda tuotteesta haluttava ja liiketoiminnallisesti elinkelpoinen yhdessä liiketoimintajohdon kanssa. Hän näkee myös palvelumuotoilun tarpeen käyttöliittymän suunnitteluvaiheessa, jossa suunnitellaan, kuinka ohjelmisto toimii ja mitä toiminnallisuuksia se tarjoaa.

Hlö C näkee yhteyden liiketoiminnan, palvelumuotoilun ja teknisen ohjelmistokehityksen välillä tärkeänä. Suurimmaksi osaksi tällä hetkellä ohjelmistoja kehitetään vesiputousmallin tai joissakin harvoissa yksiköissä Scrum-mallin mukaisesti. Häntä kiinnostaisikin nähdä jonkinlainen konsepti, kuinka palvelumuotoilu saataisiin kytkettyä paremmin osaksi ketterää ohjelmistokehitystä. Tällä hetkellä SAFe-viitekehys sisältää käyttökokemussuunnittelua, mutta tämä on haastateltavan mukaan enemmän ohjelmistoa tilaavan yrityksen liiketoimintaedustajien toiveiden mukaan tehtävää työtä.

2) Minkälaisena näet ohjelmistojen kehitysaihioiden laadun tällä hetkellä?

Haasteena hlö A:n mukaan on ollut ohjelmistokehityksen alussa julkaistavat kehitysaihiot (eng. Epic), jotka voivat olla hyvinkin erilaisia lopulta. Alussa syntyy liiketoiminnan luomaa strategiaa ja visiota tukevia kehitysaihioita, mutta tämän jälkeen voi ilmestyä kehityslis-toille muita aihioita, joilla halutaan nopeasti paikata tai korjata aikaisempaa kehitysideaa. Tällöin ns. punainen lanka kehittämisestä saattaa kadota ja laatu kärsii. Haastateltava A on myös törmännyt työssään tilanteisiin, jossa liiketoiminta on määritellyt heidän näkökulmastaan tarpeellisia muutoksia ohjelmistossa. Kiireen takia kehitysaihioita ei myöskään keretä testauttamaan loppukäyttäjillä.

Hlö B ei ole varma, mistä liiketoiminnalliset kehitysaihiot asiakkaalla syntyvät ja tästä syystä ei osaa ottaa kantaa myöskään laatuun. Hän on kuitenkin työskennellyt alalla pitkään ja uskoo, että ideat syntyvät yleensä yleisiä trendejä seuraamalla. Hän ei kuitenkaan usko, että ideat tulisivat loppukäyttäjiltä itseltään.

Hlö C näkee monesti tilanteita, joissa aluksi on ollut hyvä kehitysaihio. Kun alkuperäinen kehitystyö on saatu kiireessä päätökseen, nousee työlistoilte alkuperäisen kehitystyötä korjaavia toimenpiteitä. Tämän pohjalta laatua voisi parantaa.

3) Hyödynnetäänkö loppukäyttäjää kehitystyössä mielestäsi tarpeeksi?

Hlö A:n mukaan järjestelmän loppukäyttäjää ei hyödynnetä siinä määrin kuin pitäisi. Suurimmaksi osaksi kehitystyö perustuu liiketoiminnan strategiseen näkemykseen.

Hlö B on huomannut uransa aikana se, että välillä hyödynnetään loppukäyttäjää enemmän ja välillä vähemmän. Ongelma hänen mielestään kuitenkin monesti syntyy siinä, että annetaan loppukäyttäjien määrittää tekniset vaatimukset järjestelmille, heidän arkisiin ongelmiinsa. Eli toisin sanoen asiakasyrityksessä oleva henkilö, jolla on aivan arkinen haaste, pyrkii määrittelemään järjestelmävaatimusluettelon yritykselle. Tämä yleensä johtaa siihen, että luettelosta tulee niin laaja, että on mahdotonta toteuttaa kaikkea.

Hlö C ei ole koskaan nähnyt, että tuotteen loppukäyttäjää olisi osallistunut kehitysjuniin. Haasteena haastateltava C mukaan on yleensä kiire ja se, ettei uskota, että loppukäyttäjä osaisi antaa vastauksia. Lisähaastetta tuo myös se, että liiketoimintajohto ei usko SAFE-kouluttajia tai muita asiantuntijoita tietyissä ketterän ohjelmistokehityksen asioissa. Esimerkiksi kehitystiimien koot pitäisi hänen mielestään olla maksimissaan 5-6 henkilön kokoisia. Monesti niistä tehdään liian suuria 9-10 henkilöä ja tällöin ketteryys häviää, kun päivittäiset yhteistapaamiset ovat haasteellisia isolla porukalla ja tärkeää dialogia kehittäjien kesken on vaikea käydä.

4) Kuinka tätä voitaisiin kehittää?

Hlö A:lla olisi kehitysehdotus. Yhteistyötä liiketoiminnan ja teknisten asiantuntijoiden välillä saataisiin helpotettua sillä, että asioita käsiteltäisiin liiketoiminnan kanssa käyttöliittymä-näkymien kautta. Näin olisi helpompi viestiä, miten liiketoiminnan vaatimukset heijastuisivat teknisiin asioihin.

Haastateltava A, It-arkkitehti: ”Liiketoimintaa ei niinkään kiinnosta mitä ”pellin alla” tapahtuu vaan heitä kiinnostaa miten se tuottaa liiketoiminnallista arvoa.”

Hlä B uskoo, että kehitysaihioiden laatua voitaisiin saada vahvemmiksi, jos loppukäyttäjää osallistutettaisiin ideointiin ja kehitykseen nykyistä enemmän.

Hlö C:n mielestä pidettäessä tiimit oikean kokoisina ja dialogin jatkuvana liiketoiminnan, asiakkaiden ja teknisten osaajien välillä, saataisiin parempia lopputuloksia.

6.2 Nykytilan kartoituksen tulosten pohdinta ja uuden toimintamallin valinta

Kaikkien haastateltavien mukaan dialogia liiketoiminnan edustajien ja teknisten asiantuntijoiden välillä on vaikea käydä. Tämä voidaan nähdä johtuvan siitä, että kummallakin on oma visionsa ja tarpeensa palvelukehityksen suhteen. Kun näitä asioita ei käydä ja avata tarpeeksi, ymmärrys puuttuu ja yhteen törmäyksiä syntyy. Ratkaisuna tähän voitaisiin nähdä yhteiset työpajat, jossa keskityttäisiin rauhassa yhteisen vision ja tahtotilan vahvistamiseen. Haastateltavien mukaan myös kehitysaihioiden taso on vaihtelevaa. Asiakkailta testauttaminen jää hyvin vähäiseksi ja jälkikäteen tapahtuvia korjauksia asiakkaiden pyynnöstä tehdään paljon. Ratkaisuna voisi nähdä loppukäyttäjien mukaan ottamisen varhaisessa vaiheessa sekä laadukkaamman nykytilan kartoituksen, jossa enemmän paneuduttaisiin asiakkaiden arkisiin haasteisiin ja toiveisiin eikä niinkään teknisiin asioihin vaatimusluettelon muodossa.

Haastateltavat esittivät myös ehdotuksen, jossa uusia kehitysaihoita kehitettäessä, teknisen puolen asiantuntijat esittäisivätkin asioita prosessikaavioiden sijaan, enemmän prototyyppien ja demojen käyttöliittymä-näkymien kautta. Tämä voisi avata keskustelua molemmin puolin paremmin.

Nykytilan koskevien haastattelutulosten perusteella yksikössä on siis selkeä tarve laajalaisempaan dialogiin Liiketoiminnan- Teknisten asiantuntijoiden – asiakkaiden välillä. Asiakkaat pitäisi saada paremmin ääneen jo kehityksen alkumetreillä. Lisäksi selkeyden lisäämiseksi ehdotettiin, että liiketoiminnan edustajille ja asiakkaille esitettäisiin asioita prosessikaavioiden sijaan käyttöliittymä-näkyminä. Koska aikaa ei ole käytettävissä

kovinkaan paljon, pitäisi nämä asiat saada kompaktisti ja tehokkaasti suoritettua kaiken muun työn keskellä.



Kuva 14. Uuden toimintamallin kehittämisen prosessi.

Tämän pohjalta päädyttiin suunnittelemaan uusi konstruktio (kuva 14), johon haetaan parhaita käytäntöjä palvelumuotoilusta ja sen pohjalta kehitetystä nopeasta Design Sprintistä. Tavoitteena on luoda kokonaan uusi toimintamalli kohdeyritykseen ja ottaa se osaksi yksikön toimintaa ja selvittää kuinka se vaikuttaa kehitystyöhön.

6.3 Kehitetty konstruktio

Kehitettävän konstruktion suunnittelussa vahvimpina ajureina toimivat haastatteluiden pohjalta nousseet asiakaskeskeisyys, yhdessä suunnittelu sekä nopea testattava prototyyppi palvelusta. Asiakaskeskeisyydessä tärkeintä oli, että testattava palvelu tai tuote olisi asiakkaan tarpeesta lähtöisin. Tämä tarkoitti sitä, että asiakkaan ongelma oli jo ennen yhteistä kehittämistyöpajaa selvillä. Toimintamallin ensimmäiseksi osaksi kehitettiin Nykytilan kartoituslomake (kuva 15), jolla voitiin paremmin selvittää asiakkaiden tämän hetkiset haasteet ja toiveet palvelun suhteen.



KEHITYSKOHTTEEN RAJAUS

Mikä on kehitystyon kohde?
Uusi palvelu

Olemassa oleva palvelu (mikä?)

Mikä ovat kehitystyon tavoitteet? Miksi halutaan parantaa palvelua?
Miltä halutaan säilyttää? Miksi on tarpeen parantaa palvelua?

Millä onnistumista mitataan?
Mittausmenetelmä, jota onnistumista mitataan.
Palvelun käyttökäytännön mittarit, jota onnistumista mitataan.

Kelisi ovat palvelun loppukäyttäjät ja mistä heitä tavoittaa?
Käyttäjät, jotka ovat palvelun loppukäyttäjät. Käyttäjät, jotka ovat palvelun loppukäyttäjät.

Mitä et vielä tiedä palvelun käyttäjistä?
Tutkimusmenetelmä, jota onnistumista mitataan. Miltä halutaan säilyttää palvelun käyttäjät?

Asiakasympäristöä laajentamisen menetelmä:
Asiakasympäristöä laajentamisen menetelmä.

Kuva 15. Nykytilan kartoitus- lomake.

Kartoituksessa haetaan vastauksia, mikä on kehitystyon kohde ja tavoitteet, miltä palvelu näyttää loppukäyttäjän silmin, missä kohtaa palvelua ovat pullonkaulat, mistä arvo syntyy ja mitkä ovat akuuteimmat kehityskohteet. Tämän pohjalta oli myös helpompi koota oikea tiimi työpajaan ratkaisemaan asiakkaan ongelmaa.

Yhdessä suunnittelu sekä nopea prototyyppiin kokeiltavuus huomioitiin työpajakutsussa. Työpajaan kutsuttiin asiakkaan edustajan lisäksi tuotteen liiketoiminnallisesta ja teknisestä toteutuksesta vastaavia Tuotejohtajia, Käyttöliittymä -suunnittelija, jolla oli mahdollisuus tehdä nopean prototyyppi testattavaksi ja pöytäkirjanpitäjä sekä fasilitaattori, joka vastasi ja ohjasi työpajan kulkua.

Kutsut lähetettiin 2 viikkoa ennen työpajan alkua. Kutsussa pyydettiin myös osallistujia tekemään oman aikansa puitteissa vertailuanalyysia jo markkinoilla olevista palveluista ja miettimään mitä lisäarvoa halusivat tuottaa asiakkaille luomallaan uudella palvelulla?

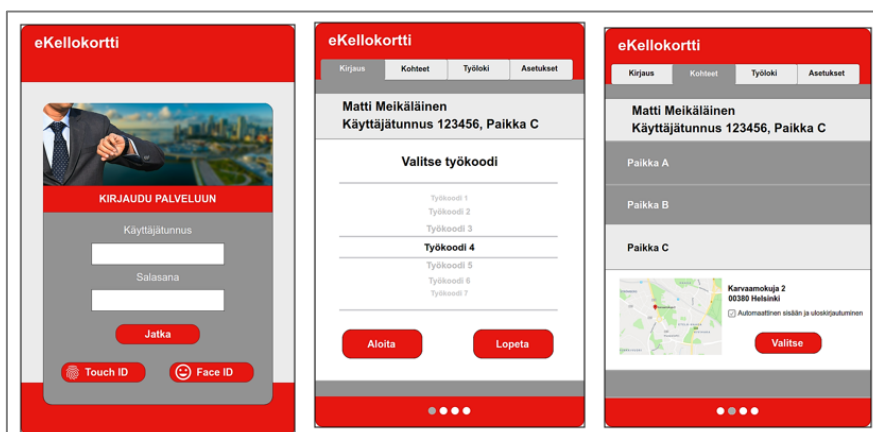
Työpaja avattiin keskustelulla ja osallistujien esittelyllä (kuva 16). Tämän jälkeen pureudimme nykytilaan ja sen pohjalta syntyneeseen tarpeeseen. Nykytilasta syntyneen tarpeen pohjalta kirkastettiin tahtotilaa, minkälainen palvelu halutaan ja minkälaista lisäarvoa halutaan sen tuottavan? Tähän hyödynsimme myös kaikkien ennakkoon saamaa vertailuanalyysia markkinoilla jo olevista palveluista, joissa jokainen erikseen esitteli havaintonsa.



Kuva 16. Työpajatoimintaa.

Koska tarve oli myös nähdä palvelu niin tekniseltä kuin liiketoiminnalliseltakin kantilta, työpajassa täydennettiin yhdessä Lean Model Canvas- mallia, joka auttoi yhteistä dialogia kaikkien osallistujien kesken sekä lisäsi huomiota palvelun kriittisiin kohtiin.

Vertailuanalyysin sekä paperille tehtyjen luonnosteluiden pohjalta käyttöliittymäsuunnittelija loi kaksi nopeaa prototyyppiä asiakkaan edustajan kokeiltavaksi (kuva 17). Palautteiden pohjalta korjattiin vielä esimerkiksi toiminnallisuuksien paikkaa ja nimiä sekä typografiaa ja värejä.



Kuva 17. Yksi työpajassa syntynyt prototyyppi palvelusta.

Ajansäästämiseksi kehittämisen työpaja pidettiin Design Sprintin viisipäiväisen suunnittelun sijaan kolme päiväisenä, jossa kaksi ensimmäistä päivää kehitettiin ja suunniteltiin yhdessä palvelua ja kolmantena päivänä testattiin ja kerättiin palaute.

Kehittämisen työpajasta syntyneitä dokumentteja olivat:

- 1) Nykytilan kartoitus, jossa selvisi asiakkaiden tämänhetkiset haasteet ja joihin haluttiin ratkaisua.
- 2) Yhdessä täydennetty Lean Model Canvas, jonka avulla saatiin vastauksia tuotteen ominaisuuksiin kuin liiketoiminnallisiin edellytyksiin.
- 3) Kaksi erilaista prototyyppiä käyttöliittymäprototyyppiä asiakkaiden testattavaksi.
- 4) Loppuraportti tuotoksineen.

6.4 Tulokset kehitetyn konstruktion jälkeen

Toinen kategoria käsittelee ohjelmistokehitysaihioiden laadun muutosta uuden toimintamallin jälkeen. Uuden toimintamallin mukainen työpaja järjestettiin elokuussa 2019.

- 1.) Saatiinko uudella menetelmällä aikaisempaa parempia lopputuloksia? Jos saatiin niin minkälaisia?

Hlö A:n mielestä työpaja auttoi näkemään alustavan kehitysaihion uudessa valossa. Lean Model Canvas haastoi pohtimaan alusta lähtien kriittisemmin hyvää ideaa sekä näkemään tuotteen liiketoiminnallisenkin puolen paremmin. Vaikeinta haastateltavan mukaan oli löytää vastaus siihen, mikä palvelussa olisi etulyöntiasemassa kilpailijoihin nähden? Mikä on erottautumistekijämme muista kilpailijoista? Myös vertailuanalyysi oli mielenkiintoista ja avarsi näkemystä kilpailijoiden vastaavista tuotteista.

Hlö B:n mukaan saatiin parempia tuloksia. Hänelle oli jo ennestään tuttuja Lean Model Canvas ja Vertailuanalyysi- malli.

Haastateltava C:n mukaan tulokset olivat lupaavia vaikkakin olisi kaivannut jonkinlaista ”Check Pointia”, jossa ideaa olisi esitelty myös johdolle ja saatu selvyys onko tämä suunta, johon halutaan mennä.

- 2.) Paraniko yhteistyö mielestänne liiketoiminnan– palvelumuotoilun ja teknisen ohjelmistokehityksen osalta?

Hlö A:n mukaan parani. Mukana oli joka osa-alueen edustajia, keskustelu oli hyvää ja antoi vastauksia heti asioihin, joita ei olisi osannut muuten ottaa huomioon.

Hlö B:n mukaan yhteistyö oli hyvää, mutta olisi hyvä jatkossa miettiä kokoonpanoa päivi-ville. Esimerkiksi tietyn osa-alueen syväosaajat, eli tuoteomistajat olisi hyvä olla heti alusta

lähtien mukana sekä tekninen arkkitehti. Alussa ei vielä haastateltavan mukaan tarvittaisi UI tai UX-suunnittelijoita.

Hlö C:n mukaan Yhteistyö paranee aina kun ollaan yhdessä samassa paikassa ja keskustellaan kasvotusten asioista.

3.) Kuinka tätä voitaisiin vielä kehittää?

Hlö A:n mielestä haasteena on, että resurssit ovat rajalliset ja osapuolet työskentelivät eri työpisteissä. Kehitysehdotuksena olisi, että kahden ensimmäisen intensiivipäivän jälkeen, voitaisiin jatkaa prototyypin tekemistä pienemmässä porukassa ja testaamista etäkokouksina videoyhteydellä. Tämän helpottaisi mukana olleiden muun työn hallintaa.

Hlö B:n mielestä olisi hyvä jatkossa saada selkeä kokonaiskuva, mitä työpajoihin tuotaisiin kehitettäväksi, koska aikataulut ja kiire määrittävät työn tekemistä. Kaikkia ideoita, kun ei vaan ole mahdollista tuoda, jolloin tarvittaisiin jokin kiireellisyysjärjestys kehittämiseksi.

Hlö C:n mukaan olisi hyvä selvittää asiakkaan tarpeita ennen työpajaa vielä hieman paremmin. Tällöin olisi myös ennakkoon miettimään sidosryhmiä eri järjestelmien välillä. Työpajassa olisi hyvä olla mukana näiden riippuvuuksien edustajat esim. analytiikka.

7 JOHTOPÄÄTÖKSET JA POHDINTA

Tutkimus aloitettiin haastattelemalla jo alalla pitkään olleita asiantuntijoita ja kuinka he näkivät dialogin ja yhteistyön liiketoiminnan, palvelumuotoilun ja teknisten osaajien välillä. Tästä saatiin erittäin hyvä pohja sille, kuinka saisimme palvelumuotoilun menetelmiä paremmin osaksi kehitystyötä.

Palvelumuotoilusta eli käyttäjälähtöisestä suunnittelusta keskustelu hyvin teknisesti rajoituneessa työyhteisössä oli hyvin haastava mutta avartava kokemus. Monet sekoittivat tämän suoraan käyttöliittymäsuunnitteluun ja käsitykset asiakkaiden syvempien tuntemusten ja toiveiden keräämisestä haastatteluiden ja havainnointien avulla olivat hyvin kaukaisia.

Teoriaosuudessa ja asiantuntijoiden haastatteluissa tuli ilmi, haasteita laadukkaassa kehittämistyössä oli paljon. Yksi oli tarvittavan ajan ja asiakkaan puuttuminen kehitystyön alkuvaiheesta. Näiden kahden asian takia usein tuotannosta valmistunutta kehityshanketta jouduttiin jälkikäteen korjaamaan ja luottamus asiakkaan sekä ohjelmistokehitysyhtiön välillä heikkeni. Lisäksi käsitys asiakkaan tarpeiden nykytilan kartoituksesta eroaa paljon käyttäjälähtöisen suunnittelun ja ohjelmistokehityksessä käytettävän vaatimusluettelon välillä. Vaatimusluettelossa usein asiakas ei päässyt kertomaan todellista syytään vaatimuksille, vaan yritti parhaansa mukaan kertoa ohjelmistoyhtiölle teknisiä yksityiskohtia, jotka olisivat ratkaisu heidän ongelmaansa. Tämä monesti aiheutti sen, että ohjelmistoja kehittävä yritys ei tiennyt tarkkaa juurisyytä ongelmaan, eikä asiakas tiennyt tarkkaa teknistä ratkaisua samaiseen ongelmaan.

Kolmas huomio tutkimuksessa oli dialogin puute. Kohdeyrityksessä dialogi asiantuntijoiden välillä oli hyvin vähäistä ja kulttuuriin tuntui kuuluvan tämän ylläpitäminen. Oli hämmäntävää huomata, kuinka oletukset asioista ohjasivat ihmisten tekemistä, niin sisäisesti työntekijöiden kesken kuin asiakkaidenkin suuntaan.

Opinnäytetyön lopputuloksena syntyi uusi toimintamalli osaksi ohjelmistokehitystyötä ja siitä oli myös selkeää hyötyä. Asiakkaan tarpeita edustavan henkilön paikalla olo helpotti viemään kehitystyötä eteenpäin nopeassa aikataulussa. Lean Model Canvas auttoi kohdeyritystä näkemään yksittäisen tuotteen kokonaispalveluna, jossa oli hyvä tuotteen teknisten asioiden lisäksi ottaa huomioon liiketoiminnalliset asiat sekä minkälaista lisäarvoa ja palvelukokemusta halutaan asiakkaalle tarjota. Vertailuanalyysi avasi myös kehittäjien näkemystä kilpailijoista. Yksi tärkeimmistä havainnoista toimintamallin muutoksen jälkeen oli tiimin sisäinen muuttuminen itseohjautuvaksi asiantuntijatiimiksi. Kaikki

osallistuivat, kaikki sitoutuivat ja kaikki tekivät parhaansa laadukkaan ohjelmiston aikaansaamiseksi.

Työtä helpotti myös kohdeyrityksen ja haastateltavien positiivinen suhtautuminen tutkimuksen tekemiseen. Haastattelut saatiin sujuvasti aikataulutettua ja ennalta lähetetyt kysymykset auttoivat henkilöitä saamaan kiinni aiheesta hieman paremmin. Kohdeyrityksen puolelta tutkimukseen annettiin vapaat kädet ja tuettiin tarvittaessa. Haluan kiittää kohdeyritystä mahdollisuudesta tehdä opinnäytetyön heille ja Lahden ammattikorkeakoulun ohjaavaa opettajaa Pasi Tiihosta auttamisesta ja tukemisesta.

7.1 Kehitysehdotus

Toimintamallin muutoksen jälkeisissä tuloksissa nousi esiin entistä tarkempi nykytilan selvityksen tarve, jolloin olisi mahdollista saada myös sidosryhmien edustajia kuten analytiikan asiantuntijoita työpajoihin mukaan. Näitä olisi hyvä lähteä jatkokehittämään tarkentamalla kartoituslomakkeen kysymyksiä sekä laajentamalla sidosryhmiä. Myös uusien kehittämiskohteiden priorisointijärjestyksen tarve nousi esille. Tätä voitaisiin kehittää esimerkiksi ottamalla tuotehallinnasta vastaavat henkilöt paremmin mukaan toimintamalliin alusta asti ja päättämään organisaatiotasolla mihin suuntaan kehitystä halutaan viedä.

7.2 Tutkimuksen pätevyys ja luotettavuus

Tutkimuksen luotettavuutta arvioidaan validiteetin ja reliabiliteetin avulla. Validiteetillä pyritään selvittämään, tutkitaanko oikeita asioita ja reliabiliteetin avulla tutkimuksen luotettavuutta. Kvalitatiivisessa tutkimuksessa luotettavuutta mitataan tutkimuksen luotettavuudella, siirrettävyydellä ja tutkimuksen vahvistettavuudella (Kananen 2008, 123-126). Validiteetin osalta tutkimuksen lähtökohtana oli selvittää, olisiko palvelumuotoilun menetelmillä mahdollista parantaa kohdeyrityksen ohjelmistokehitysaihioiden tämänhetkistä laatua. Kehittämistehtävänä oli parantaa toimintatapoja ja tätä kautta saada laadukkaampia lopputuloksia. Haastattelutulokset ja niiden pohjalta tehdyt toimintamallin muutokset vastaavat opinnäytetyönprosessin alussa mietittyyn kehittämistehtävään ja tutkimuskysymykseen. Tulokset ja toimintamalli on pyritty esittämään ja suunnittelemaan niin, että myös muilla yrityksillä olisi mahdollisuus hyötyä ja käyttää ohjelmistojen kehittämisessä. Reliabiliteetin osalta haastatteluun valikoituivat henkilöt, joilla on paljon omakohtaista kokemusta kohdeyrityksen kehittämistyöstä ja tutkittavasta aiheesta. Luotettavuutta voi heikentää rajallinen otanta ja havaintojen muodostaminen sekä tämän pohjalta tehdyt yleistykset. Opinnäytetyöhön hankittu tietoperusta tuki tutkimustyötä myös hyvin. Tietoperustan pohjalta oli helpompaa ymmärtää ohjelmistokehityksen elinkaarta ja haasteita sekä palvelumuotoilun perusteita hyötyineen ja haasteineen.

LÄHTEET

Ashmore, S., Runyan K. 2014. Introduction to Agile methods. Addison-Wesley Professional.

Auer, A., Auer, L., Heinäsmäki, M., Hölttä, J., Kalliala, E., Laanti, M., Laine, K., Lekman, L. 2013. Ketterää kehitystä, Finn Lectura.

Bertrand M. 2014. Agile! The good, the hype and the ugly. Springer.

Cunningham W. 2001. Ketterän ohjelmistokehityksen julistus. [Viitattu 4.3.2019] Saatavissa: <https://agilemanifesto.org/iso/fi/manifesto.html>.

Digite. 2019. What is a Kanban Board? [Viitattu 4.3.2019] Saatavissa: <https://www.digite.com/kanban/kanban-board/>.

Girvan L., Debra P. 2017. Agile and Business analysis. BCS, The Chartered Institute for IT.

Gorman, G.E., Clayton, P. Shep S., Clayton A. 2005. Qualitative Research For The Information Professional: A Practical Handbook. 2nd ed. London: Facet Publishing

Gröpl E. 2009. Service Design methods Mind Map. [Viitattu 5.3.2019] Saatavissa: <https://www.flickr.com/photos/groep1/3611532540>.

Harle R. 2019. CGI.fi: Safe tulee ja pelastaa? -blogi. [Viitattu 21.2.2019]. Saatavissa: <https://www.cgi.fi/fi/blogi/safe-tulee-ja-pelastaa>.

Hirsjärvi S., Hurme H. 2000. Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Yliopistopaino.

Hodson B. 2018. UXCollective. [Viitattu 27.2.2019] Saatavissa: <https://uxdesign.cc/how-to-storyboard-experiences-fc051e2bc04d>

Itkonen, J. 2018. SAFe sanasto Suomeksi – Ymmärrystä äidinkielellä -blogissa. [Viitattu 13.6.2019] Saatavissa: <https://www.nitor.com/fi/uutiset-ja-blogi/safe-sanasto-suomeksi-ymmarrysta-aidinkielella>.

Junikka J. 2019. Scrum markkinoinnin johtamisen toimintamallina. [Viitattu 4.3.2019] Saatavissa: <https://fulmore.fi/blogi/scrum-markkinoinnin-johtamisen-toimintamallina/>.

Juvonen, R. 2018. Ohjelmistoprojektien sudenkuopat ja miten ne vältetään. BoD – Books On Demand.

Kananen, J. 2008. Kvali Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Jyväskylän yliopistopaino.

Keinonen T. 2018. Muotoilutarinat.fi. [Viitattu 12.2.2019] Saatavissa: <https://www.muotoilutarinat.fi/fi/artikeli/k%C3%A4ytt%C3%A4j%C3%A4l%C3%A4ht%C3%B6inen-suunnitelu/>.

Kiiskinen, J. 2017. Vesiputousmalli vs. Ketterät menetelmät -blogi. [Viitattu 21.2.2019]. Saatavissa: <http://blogi.sysart.fi/moderni-ohjelmistokehitys-pahkinankuoressa-vesiputousmalli-vs.-ketterat-menetelmat>.

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

Maurya A. 2012. Lean Model Canvas. [Viitattu 27.2.2019] Saatavissa: <https://blog.leanstack.com/why-lean-canvas-vs-business-model-canvas-af62c0f250f0>.

Minimalist 2019. Customer Profile template. [Viitattu 27.2.2019] Saatavissa: <https://www.minimalist.business/customer-profile-template/>.

Moreira M., 2017. The Agile Enterprise. Apress.

Nitor. 2018. SAFe 4.5 sanasto. Saatavissa: <https://www.nitor.com/application/files/8415/2524/8051/Nitor-SAFE-4.5-FIN.pdf>.

Ojasalo, K., Moilanen, T., Ritalahti, J. 2015. Kehittämistyön menetelmät –Uudenlaista osaamista liiketoimintaan. Sanoma Pro. Oppariapu. [viitattu 4.3.2019] Saatavissa: <https://oppiapu.wordpress.com/konstruktivinen-tutkimus>.

Powell-Morse A. 2017. Scaled Agile Framework: What Is It and How Do You Use It? – blogi. [Viitattu 21.2.2019] Saatavissa: <https://airbrake.io/blog/sdlc/scaled-agile-framework>

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

SAFe. 2019. Full Safe. [Viitattu 4.3.2019] Saatavissa: <https://www.scaledagileframework.com/#>.

Shuttleworth, M. 2008. Qualitative Research Design. [Viitattu 1.4.2015]. <https://explorable.com/qualitative-research-design>.

Smartinsights.2019. Business Model Canvas. [Viitattu 27.2.2019] Saatavissa: <https://www.smartinsights.com/digital-marketing-strategy/online-business-revenue-models/summarising-business-models-use-the-business-model-canvas-diagram/>.

Stickdorn M., Hormess M., Lawrence A., Schneider J. 2018. This is Service Design Doing. O'Reilly.

Thinkingportfolio 2016. Kuinka valita sopiva menetelmä projektiin? Saatavissa: <https://thinkingportfolio.com/kuinka-valita-sopiva-menetelma-projektiin/>.

Toptal 2019. Designers [Viitattu 27.2.2019] Saatavissa: <https://www.toptal.com/designers/product-design/customer-journey-maps>.

Tuulaniemi J. 2011. Palvelumuotoilu. Talentum.

Wells D. 2001. Planning/Feedback loops. [Viitattu 4.3.2019] Saatavissa: <http://www.extremeprogramming.org/map/loops.html>.

Yle Uutiset 2019. Lukijat vastasivat: "Uusi matkakortinlukija on täysi susi!" – HSL lupaa viilata näyttöä, mutta ei aio muuttaa laitteiden toimintaperiaatetta. [viitattu 4.3.2019]. Saatavissa: <https://yle.fi/uutiset/3-9320151>.

LIITTEET

Kysymykset ohjelmistokehityksen nykytilasta.

- 5) Minkälaisena näet Liiketoiminnan – Palvelumuotoilun – Teknisen ohjelmistokehityksen yhteistyön tällä hetkellä?
- 6) Minkälaisena näet ohjelmistojen kehitysaihioiden laadun tällä hetkellä?
- 7) Hyödynnetäänkö loppukäyttäjiä kehitystyössä mielestäsi tarpeeksi?
- 8) Kuinka tätä voitaisiin kehittää?

Kysymykset uuden palvelumuotoilu- toimintamallin hyödyntämisen jälkeen.

- 4.) Saatiinko uudella toimintamallilla aikaisempaa parempia lopputuloksia? Jos saatiin niin minkälaisia?
- 5.) Paraniko yhteistyö mielestänne Liiketoiminnan– Palvelumuotoilun ja Teknisen ohjelmistokehityksen osalta?
- 6.) Kuinka tätä voitaisiin vielä kehittää?