



Muotoiluajattelun ja ketterän kehittämisen symbioosi: Muotoiluohjautuva ketterä projekti- malli

Carita Savin

2023 Laurea

A decorative horizontal bar at the bottom of the page, divided into three segments of different colors: pink, blue, and teal.



Laurea-ammattikorkeakoulu

**Muotoiluajattelun ja ketterän kehittämisen symbioosi:
Muotoiluohjautuva ketterä projektimalli**

Carita Savin
Tulevaisuuden digitaaliset palvelut
Opinnäytetyö
Elokuu, 2023

Carita Savin

Muotoiluajattelun ja ketterän kehittämisen symbioosi: Muotoiluohjautuva ketterä projektimalli

Vuosi

2023

Sivumäärä

74

Vuoden 2020 Covid-19 Pandemia kiihdytti digitaalisten palveluiden tarvetta. Digitaalisia palveluja tuli pystyä tuottamaan nopeasti. Palveluiden nopea lisääntyminen johti tilanteeseen, jossa käyttäjille oli tarjolla suuri määrä hyvin vaihtelevan tasoisia digitaalisia palveluja. Käyttäjät valitsivat mieluummin sellaiset palvelut, joita oli helppo käyttää ja joilla asiat sai helposti hoidettua. Palveluiden suuri määrä lisäsi käyttökokemuksen merkittävyyttä huomattavasti.

Tämän opinnäytetyön tavoitteena on löytää keinoja tuottaa asiakkaan odotukset täyttäviä ja käyttäjälähtöisiä digitaalisia palveluja entistä tehokkaammin. Käytännön kautta oli vahva mielikuva siitä, että muotoiluajattelun menetelmien yhdistäminen saumattomasti ketterän kehittämisen menetelmien kanssa voisi tuottaa halutun lopputuloksen. Avoimeksi jäi se, miten nämä menetelmät yhdistetään käytännössä ja miten se vaikuttaa tiimin kokoonpanoon sekä projektin myyntiin. Lisäksi haluttiin tutkia digitaalisten palveluiden tulevaisuudennäkymiä käyttäjälähtöisyyden näkökulmasta sekä perehtyä jo olemassa oleviin parhaisiin käytänteisiin tällä saralla. Tutkimuksen toimeksiantaja on kansainvälinen ohjelmisto- ja palveluyritys, jossa toimii toistatuhatta asiantuntijaa.

Opinnäytetyössä käydään läpi muotoiluajattelun ja ketterän kehittämisen ideologiaa ja käytänteitä. Sen jälkeen kuvataan karkealla tasolla toimeksiantajan ohjelmistotuotannon nykytila ja siihen liittyvät haasteet. Tutkimus etenee Design Councilin tuplatimanttimallin mukaisesti: aluksi suoritetaan taustatutkimus ja perehdytään käyttäjälähtöisyyden lisääntymisen syihin sekä siihen, miten muut toimijat ovat pyrkineet uutta tilannetta ratkaisemaan. Ymmärrystä syvennetään haastattelujen avulla. Kootun tiedon pohjalta lähdetään luomaan mallia, joka vastaa asetettuun tavoitteeseen. Lopuksi arvioidaan, onko näin luodulla toimintamallilla mahdollista ratkaista nykytilan haasteet.

Opinnäytetyö esittelee ohjelmistoprosessin rungon, jossa muotoiluajattelun ja ketterän kehittämisen menetelmät luovat yhtenäisen menetelmäkokonaisuuden alkaen ratkaisun määrittelystä ketterän ohjelmistoprojektin toteutukseen. Se kuvaa tehtävien vaiheistamisen ja eri vaiheiden roolit sekä ottaa kantaa ohjelmistoprojektin myyntiin. Malli on tässä kehittämistehtävässä nimetty muotoiluohjautuvaksi ketteräksi projektimalliksi. Uskon, että tässä opinnäytetyössä ehdotetuilla lähestymistavoilla pystytään tuottamaan parempaa käyttäjäkokemusta ja sen myötä vahvempaa käyttäjien sitouttamista.

Asiasanat: palvelumuotoilu, scrum, käyttäjälähtöisyys, digitaaliset palvelut, ohjelmistotuotanto

Carita Savin

The symbiosis of Design Thinking and Agile Development: Design-Driven Agile Project Model

Year 2023 Pages 74

The 2020 Covid-19 Pandemic artificially accelerated the global adoption of digital apps and services, whether end-users were ready not. Consequently, vast numbers of e-products were rushed through development phases and onto the market, without much consideration for end-user comfort, convenience and overall utility/efficacy of product. Before this, user comfort with digital products had mostly been a developer's afterthought (often driven by "Geek-developer" syndrome). As the drivers and developers of digital products we need fresh and innovative ways of product development which place the needs and comfort of our consumer at the very core of product design.

The purpose of this thesis is to find ways to produce user-oriented digital services that meet customer expectations more efficiently. There was already a strong belief that combining both service design and agile development methods to a seamless process could provide the desired result. What remained open was how to do it in practice and how it affects the team structure and the sale of the project. We also wanted to study the prospects of digital services from the user experience point of view and learn more about the best practices that already exist in this field. The client of this research is an international software and service company with more than one thousand experts.

This thesis walks through the ideology and practices behind design thinking and agile development. Then it shortly describes the current state of the client's software development process and related challenges. The research plan is based Design Council's Double Diamond Innovation Framework. The process starts with background research and dives a bit deeper to the root causes why user experience is - and will be - more valued than earlier. It attempts to find the best practices that other companies or researchers have created to solve this.

This thesis introduces a new software project framework, where the methods of design thinking and agile development create a unified process to define and implement a digital product. It describes the phasing of the project, the recommended roles for each phase, and talks a bit about the sales of the software project. In this research, the framework has been named as a Design-Driven Agile Project Model. I believe the approaches proposed in this thesis will achieve hitherto unattainable levels of end-user satisfaction and thus retention.

Keywords: Design Thinking, Scrum, User Experience, Digital Services, Software Process

Sisällys

1	Johdanto ja rakenne	8
1.1	Taustaa	8
1.2	Tavoite	9
1.3	Toimintaympäristö	9
2	Tietoperusta.....	10
2.1	Digitaalisten palveluiden suunnittelu	12
2.2	Muotoiluajattelu.....	13
2.3	Design Sprint	14
2.4	Ohjelmistokehityksen elinkaarimalli	15
2.5	Ketterä kehittäminen (Scrum)	18
3	Kehittämisasetelma	22
3.1	Ohjelmistotuotantoprosessin nykytila.....	22
3.2	Kehittämissuunnitelma	24
3.3	Tutki-vaihe: taustatutkimus ja ymmärryksen luominen.....	26
3.3.1	Kipupisteiden tunnistaminen	27
3.3.2	Sidosryhmäkartoitus ja rajaus	28
3.3.3	Asiakaskeskeisyyden merkitys kiihtyvässä digitalisaatiossa	29
3.3.4	Muotoilumenetelmät ketterän kehittämisen tukena.....	31
3.4	Määritä-vaihe: haasteen tarkentaminen.....	34
3.4.1	Haastateltavana ydinprosessien ohjeistuksesta vastaava henkilö.....	35
3.4.2	Haastateltavana Ohjelmistopalvelut-yksikön vetäjä ja johtava arkkitehti .	36
3.4.3	Haastateltavana määrittely- ja toteutusvaiheen toimijat	37
3.4.4	Haastateltavana digitaalisten ratkaisujen myyjä.....	40
3.4.5	Koonti haastatteluista	41
3.5	Kehitä-vaihe: Ratkaisun kehittäminen	46
3.5.1	Muotoiluohjautuva ketterä projektimalli	46
3.5.2	Konseptointivaihe muotoiluohjautuvassa projektissa.....	48
3.5.3	Ymmärryksen luominen Design Sprintin avulla.....	49
3.5.4	Ratkaisun suunnittelu ja User Story Mapping.....	50
3.5.5	Priorisointi ja kehitysjonon luominen	51
4	Johtopäätökset ja pohdintaa	56
4.1	Muotoiluohjautuva malli ja toimeksiantajan haasteet	56
4.2	Havainnot osittaisista kokeiluista	59
	Kuviot.....	64
	Taulukot.....	65
	Liitteet.....	66

1 Johdanto ja rakenne

Opinnäytetyö rakentuu siten, että aluksi esitellään opinnäytetyön tietoperusta. Tietoperusta koostuu digitaalisten palveluiden erityispiirteistä sekä niiden tuottamiseen liittyvistä menetelmäkehyksistä, joiden tavoitteena on kehittämisprosessin nopeus sekä käyttäjäkokemuksen varmistaminen. Kolmas luku käy läpi kehittämisasetelman sekä kehittämissuunnitelman mukaisen työn toteuttamisen. Neljännessä luvussa pohditaan, miten hyvin tässä kehitetty malli vastaa nykytilan haasteisiin. Lisäksi tuodaan esiin osittaisista kokeiluista saatuja kokemuksia.

Johdannon alaluvuissa esitellään lyhyesti opinnäytetyön taustaa, tavoitteita ja toimintaympäristöä.

1.1 Taustaa

Vuoden 2020 Covid-19 Pandemia ja sen pakottama eristäytyminen pakotti yritykset siirtymään digitaalisiin palvelukanaviin. Sen myötä syntyi tarve tuottaa uusia digitaalisia palveluja ennätysajassa. Päivittäistavaroiden ostaminen siirtyi verkkokauppoihin, tilauksia ja toimituksia hallinnoitiin digitaalisesti ja tuotteet toimitettiin ilman kohtaamista. Terveystuollossa siirryttiin chat- ja videopalveluihin, toimitukset tyhjenivät ja kokoukset muuttuivat etäkokouksiksi. Digitaalisesta tuli uusi normaali. (Deloitte)

Nopea digitaalisten palveluiden kysynnän kasvaminen johti tarpeeseen tuottaa uusia digitaalisia palveluja kiihtyvällä tahdilla. Digitaalisten palveluiden nopea lisääntyminen johti siihen, että käyttäjille on tarjolla suuri määrä digitaalisia palveluja. Syntyi digipalveluähky. Monet palvelut oli suunniteltu nopeasti ja vaikka tekninen toteutus oli hyvälaatuinen, niiden käytettävyyden ja käyttäjäkokemuksen taso oli hyvin vaihteleva. Hankalasti käytettävällä palvelulla oli vaikea saada asiat hoidettua, joten käyttäjät valitsivat mieluummin ne palvelut, joilla asioiden hoitaminen sujui mutkattomasti ja joita oli mukava käyttää. Yritykset ja digitaalisten palveluiden tuottajat ovatkin nyt uuden haasteen edessä: miten tuottaa nopeasti ja tehokkaasti digitaalisia palveluja, joiden käyttökokemus on ensiluokkainen? (Bestman 2019)

Ketterät toimintamallit, kuten Scrum, ovat jo vakiinnuttaneet jalansijansa ohjelmistotuotannossa. Ketterä malli auttaa tiimejä tuottamaan digitaalisia palveluja nopeasti, mutta se ei ota kantaa siihen, pystyykö suunniteltu palvelu ratkaisemaan käyttäjän ongelman. Se ei myöskään ota kantaa siihen, millainen on palvelun tuottama käyttäjäkokemus. (Bestman 2019) Käyttäjäkokemuksen suunnittelu on osa palvelumuotoilua. Tässä yhteydessä keskitytään käyttäjäkokemuksen luomiseen ja siinä yhteydessä usein käytettyihin menetelmiin. Erityisesti tutkitaan Design Sprint -menetelmää, joka on nopeatempoinen tapa luoda yhteinen ymmärrys palvelun tavoitteista ja tuottaa prototyyppi palvelun keskeisimmistä osista.

1.2 Tavoite

Alkuhaastattelussa muodostettu tutkimuskysymys kuului: ”**Kuinka voisimme tuoda muotoilu-toiminnan kiinteäksi osaksi organisaation toimintamallia sekä ohjelmistokehityksen että myynnin näkökulmasta?**” Opinnäytetyön tehtävänä on tutkia laajemmin, onko yhdistämiselle tarvetta ja mistä tarve johtuu. Lopputuotoksena odotetaan syntyvän ”Muotoiluohjautuva ketterä projektimalli”, jossa muotoiluajattelun ja ketterän kehittämisen menetelmän luovat yhtenäisen prosessimallin asiakaslähtöisten digitaalisten palveluiden tuottamiseksi.

Opinnäytetyössä tutkitaan, miten palvelumuotoilun ja ketterän kehittämisen menetelmät voidaan yhdistää yhtenäiseksi kokonaisuudeksi siten, että luotu prosessimalli hyödyntää molempien kehittämismallien vahvuuksia. Tavoitteena on selvittää, voidaanko palvelumuotoilun ja ketterän kehittämisen menetelmien yhdistämisellä luoda lisäarvoa digitaalisten palveluiden tuottamiselle. Lisäksi tutkitaan sitä, miten näiden menetelmäkehysten yhdistäminen muokkaa ketterää kehittämistä ja sen käytänteitä.

Lisäksi luodaan katsaus digitaalisten palveluiden tuottamiseen tulevaisuudessa käyttäjälähtöisyyden näkökulmasta ja uuden markkinatilanteen luomiin vaatimuksiin yrityksen toiminnalle. Vastaavasti perehdytään siihen, millaisia parhaita käytänteitä muotoilun ja ketterän kehittämisen yhdistämiseen on jo muotoutunut muiden toimijoiden toimesta.

1.3 Toimintaympäristö

Tutkimuksessa hyödynnetään kansainvälisen ohjelmisto- ja palveluyrityksen liiketoimintayksiköitä ja henkilökuntaa. Yrityksessä on käytössä useita ohjelmistokehittämisen viitekehyksiä ja yrityksen ydinprosessit on koottu käsikirjaksi, jota hyödynnetään ohjelmistokehityksessä sekä ohjelmistoprojektien työsuunnitelmissa ja myyntitarjouksissa.

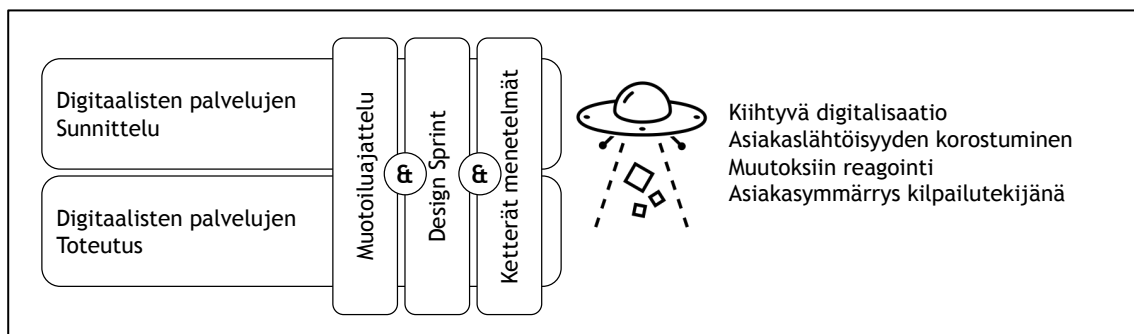
Toimeksiantajan ohjelmistotuotannossa on nykyisin käytössä ohjelmistotuotantoprosessi, jossa tarkka tulevan järjestelmän määrittely tehdään ennen toteutusvaihetta. Määrittely ja toteutus jäävät usein toisistaan irrallisiksi ja sen uskotaan olevan syynä siihen, että valmis ohjelmistoratkaisu ei aina vastaa asiakkaan odotuksia. Muotoilupuolelta on tullut vahva toive siitä, että muotoilu olisi jatkuvaa toimintaa koko projektin ajan.

Tutkimuksen toimeksiantaja on kansainvälinen ohjelmisto- ja palveluyritys, jolla on kattava liiketoimintaa ja digitalisaatiota tukeva palveluvalikoima. Palveluvalikoima koostuu mm. konsultointipalveluista, ratkaisusuunnittelusta, asiakaskokemuksen johtamisesta ja sovelluskehityspalveluista. Lisäksi yrityksellä on laaja tarjoama data- ja integraatiopalveluita, verkkokäyttö- ja asiakkuudenhallintapalveluita, tuotannon- ja toiminnanohjausjärjestelmiä sekä palvelunhallinta-, tietoturva ja pilviratkaisuja. Yrityksessä toimii toistatuhatta asiantuntijaa. Tämä kehittämistehtävä rajataan koskemaan vain uusien digitaalisten palveluiden tuottamista

ja niiden ratkaisusuunnittelua, käyttökokemusta sekä sovelluskehitystä. Organisaatio pidetään anonyyminä, sillä itse ohjelmistoprosessin kehitystyön kannalta kohdeorganisaatiolla ei ole merkitystä.

2 Tietoperusta

Tietoperusta rakentuu digitaalisten palveluiden suunnittelun ja toteutuksen sekä niihin vaikuttavien ilmiöiden ympärille. Taustatutkimuksessa ilmiöinä tulivat esiin kiihtyvä digitalisaatio, asiakaslähtöisyyden korostuminen, muutokseen reagointi ja asiakasymmärryksen nousu kilpailutekijäksi. Kokonaisuutta pyritään ratkaisemaan muotoiluajattelun, Design Sprint -menetelmän ja ketterien menetelmien keinoin.



Kuvio 1: Tietoperustan lähtökohdat

Mukaan on valittu digitaalisten palveluiden tuottamiseen liittyvät menetelmäkehykset, joiden tavoitteet liittyvät käyttökokemuksen luomiseen sekä sen varmistamiseen, ratkaiseeko palvelu käyttäjän ongelman. Ohjelmistokehityksen menetelmistä mukaan on valittu nopeaan kehityssykliin ja käyttäjätarpeiden toteutumisen varmistamiseen keskittyvät menetelmät.

Käymme aluksi läpi digitaalisten palveluiden suunnittelun erityispiirteitä ja sen jälkeen tutustumme ohjelmistokehityksen elinkaarimalliin, mikä kuvaa ohjelmistoprosessin yleiset vaiheet. Tämän jälkeen käymme läpi muotoiluajattelun, Design Sprint -menetelmän ja ketterän kehittämisen mallin.

Sekä palvelumuotoilulle että ketterille menetelmille on yhteistä se, että ne pyrkivät varmistamaan mahdollisimman aikaisessa vaiheessa sen, että tuote vastaa käyttäjän tarpeisiin. Palvelumuotoilussa kerätään käyttäjäpalautetta jo ennen palvelun toteutusta prototyypin avulla. Ketterissä menetelmissä ohjelmistotuotteen julkaisutiheys on muutamia viikkoja ja käyttäjäpalautetta kerätään kunkin julkaisun jälkeen. Design Sprint - jolla nimestään huolimatta ei ole mitään tekemistä ketterän kehittämisen (Scrum) sprinttien kanssa - on uusin menetelmä, joka on syntynyt tarpeesta jalostaa alkuperäinen idea testattavaksi tuotteeksi/palveluksi

yhdessä viikossa. Design Sprintillä on paljon yhtäläisyyksiä niin ketterän kehittämisen kuin muotoiluajattelunkin kanssa ja siksi usein nousee esiin kysymys siitä, miten nämä kolme eroavat toisistaan. Taulukko 1 koostaa tiivistetysti näiden menetelmien keskeisimmät ominaisuudet ja niiden ensisijaisen käyttötarkoituksen.

Menetelmä / Ominaispiirre	Muotoiluajattelu	Design Sprint	Ketterä Sprint
Ongelman määrittely	Kyllä		
Ongelman ratkaisu	Kyllä	Kyllä	
Ratkaisun määrittely		Kyllä	Kyllä
Ratkaisun toteutus			Kyllä
Kesto	Kuukausista vuosiin	Korkeintaan 5 päivää	1-2 viikkoa
Päättymisen	Avoin	Aikariippuvainen	Kiinteä aikataulu
Laajuus	Laaja-alainen	Tarkasti rajattu	Tiimin määrittelemä
Ongelman koko	Iso kuva	Yksi ongelma	Yksi inkrementti
Käyttäjäpalaute	Nopeat prototyypit Välitön palaute, laadullinen tieto	Nopeat prototyypit Välitön palaute, laadullinen tieto	Nopeat julkaisut Myöhäinen palaute, mitattua tietoa
Iteratiivinen	Kyllä	Kyllä	Kyllä
Yhteiskehittäminen	Kyllä	Kyllä	Kyllä
Moniosaajatiimi	Kyllä	Kyllä	Kyllä
Vahva prosessi	Kyllä	Kyllä	Kyllä
Vaatii fasilitointia	Kyllä	Kyllä	Kyllä

Taulukko 1: Keskeisimmät ominaisuudet ja käyttötarkoitus - Muotoiluajattelu, Design Sprint ja Ketterä Sprint (Vetan 2020)

2.1 Digitaalisten palveluiden suunnittelu

Palveluja ajatellaan usein yksittäisinä palvelutapahtumina, esimerkiksi ostosten tekemistä kaupassa tai avun kysymistä myyjältä. Digitalisaation myötä palvelukanavien määrä on kasvanut ja henkilökohtainen kohtaaminen on vain yksi monista tavoista saada palvelua. Tyypillisiä palvelukanavia ovat mm. web-sivustot, mobiilisovellukset, media, mainokset ja asiakaspalvelu. Jos asiakas löytää hyvän tarjouksen web-sivustolta ja menee kivijalkaliikkeeseen, jonka asiakaspalvelija ei ole tietoinen tarjouksesta eikä hänellä ole lupaa myydä tuotetta online-tarjouksen hinnalla, on asiakaskokemus pilalla. (Polaine, Lovlie & Reason 2013, 19-22)



Kuvio 2: Käyttökokemus syntyy useiden palvelukanavien kokonaisuudesta (Polaine, Lovlie & Reason 2013, 23, piirretty uudelleen).

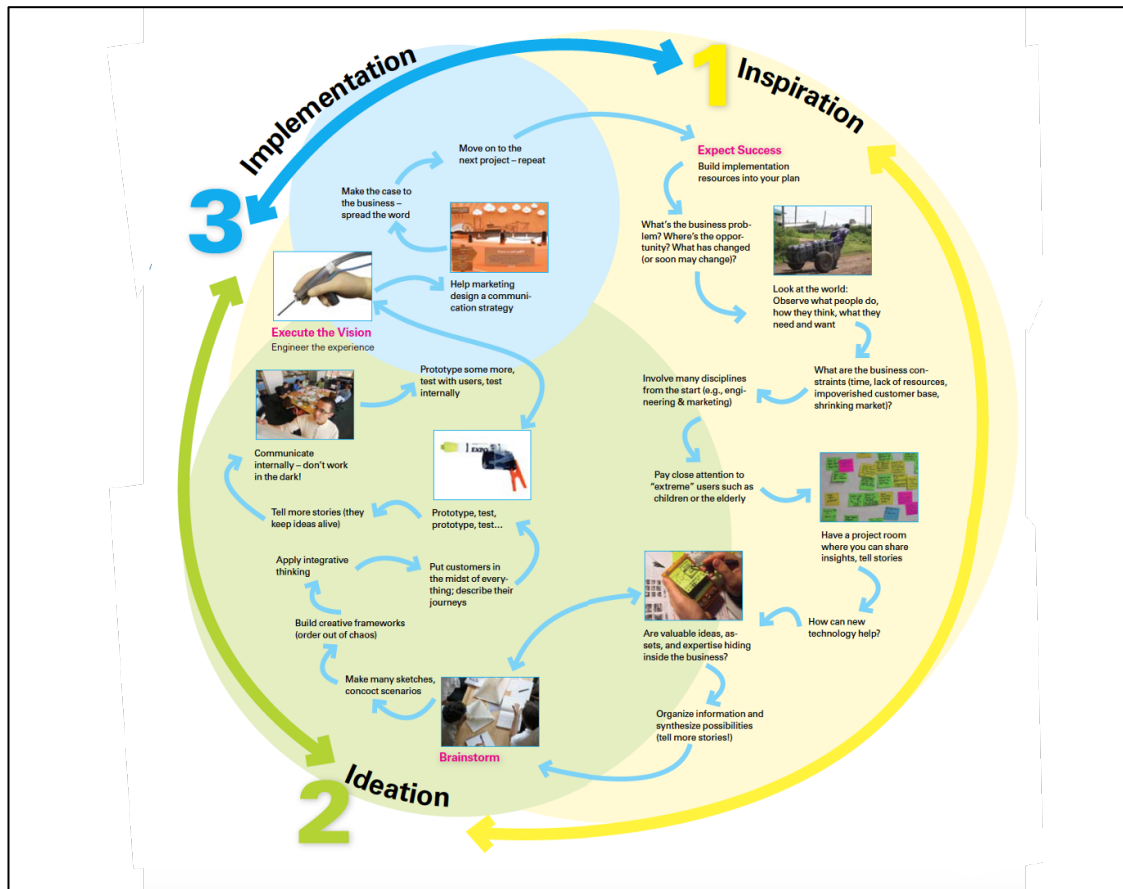
Tämän vuoksi digitaalisten palveluiden suunnittelussa on tärkeää huomioida, että ne ovat usein vain osa asiakkaan palvelukokemusta ja kokonaiskokemus syntyy useiden palvelukanavien kokonaisuudesta. Ei siis riitä, että yksittäinen palvelun osa on hyvin suunniteltu ja toteutettu. Näin ollen digitaalisten palveluiden suunnittelun osalta viitekehyykseksi on valittu muotoiluajattelu (eng. Design Thinking), jonka prosesseista ja menetelmistä usein käytetään myös termiä Palvelumuotoilu (eng. Service Design). Se on valittu sen vuoksi, että sen keskiössä on asiakkaalle tuotettu lisäarvo ja toimintorajojen yli ulottuva käyttökokemus. (Clatworthy 2017, 2)

2.2 Muotoiluajattelu

Muotoiluajattelu on kokonaisvaltainen toimintamalli, jota ohjaa muotoilijan herkkyys sekä valitut menetelmät. Tavoitteena on löytää ratkaisu, joka vastaa ihmisten tarpeisiin tarkoituksenmukaisen teknologian avulla siten, että se on liiketoiminnallisesti kannattavaa ja tuottaa lisäarvoa asiakkaille. (Brown 2008, 4) Se ymmärretään ajattelumallina, joka johtaa muutokseen, jatkuvaan kehittymiseen sekä uusiin tapoihin elää ja harjoittaa liiketoimintaa (Tschimmel 2012, 1). Perinteisesti muotoiluajattelu nojaa muotoilijan kykyyn katsoa maailmaa laajalaisesti ja kokonaisvaltaisesti, mutta sen menetelmät ovat omiaan tukemaan moniosaajatiimejä kaikenlaisissa organisaatioissa. Onnistuminen edellyttää uudenlaisen, abduktiivisen ajattelumallin omaksumista, jossa tunteet ja intuitio ovat yhtä tärkeitä kuin rationaalisuus. Abduktiivinen ajattelija on yhtä aikaa analyttinen ja empaattinen, rationaalinen ja tunteellinen, menetelmäorientoitunut ja intuitiivinen sekä tavoitteellinen ja spontaani. (Tschimmel 2012, 3)

Muotoilun prosessi on vahvasti kokeileva ja pikemminkin erilaisten tilojen muodostama systeemi kuin etukäteen määritelty tietyssä järjestyksessä etenevä sarja toimintoja. Kussakin tilassa suoritettavat, toisiinsa linkittyvät aktiviteetit yhdessä muodostavat innovaatioiden jatkumon. Prosessi voi aluksi tuntua kaoottiselta, mutta kokemus on osoittanut sen tehokkaaksi ja järkeväksi. (Brown 2008, 6-7)

Prosessi muodostuu kolmesta tilasta: Inspiraatio (eng. Inspiration), ideointi (eng. Ideation) ja toteutus (eng. Implementation). Inspiraatiotilassa luodaan ymmärrys asiasta, on sitten kyseessä ongelma, mahdollisuus tai molemmat. Tavoitteena on luoda motivaatio asian ratkaisuun. Ideointi on tila, jossa luodaan, kehitetään ja testataan ideoita, jotka saattavat ratkaista asian. Toteutustilassa keskitytään ratkaisun saattamiseen käyttäjien saataville. Tyypillisesti näihin eri tiloihin, etenkin inspiraatioon ja ideointiin, palataan takaisin ideoiden kehityksessä ja uusia ratkaisusuuntia valittaessa. (Brown 2008, 6-7)



Kuvio 3: Muotoilun prosessi ja sen kolme tilaa (Brown 2008, 6-7)

Muotoilussa käytettäville työkaluille ja menetelmille on ominaista se, että ne antavat tilaa ajatteluprosessille, parantavat ryhmän sisäistä kommunikointia sekä helpottavat asian esittämistä ryhmän ulkopuolisille henkilöille. Lisäksi käytössä on menetelmiä, jotka auttavat osallistujia ajattelemaan normaalia joustavammin ja radikaalisemmin. Mukana on myös käyttäjälähtöisen suunnittelun menetelmiä. Palvelumuotoilun prosessille ja työkaluille onkin tyypillistä visuaalisuus, keskeneräisyys, prototyypit, virheiden hyväksyminen, käyttäjäkeskeisyys, haavainnointi ja osallistaminen. (Tschimmel 2012, 3-4)

2.3 Design Sprint

Jake Knapp kehitti Design Sprint -menetelmä parantaakseen ratkaisujen suunnittelun tehokkuutta. Tarkastellessaan tilannetta, hän huomasi, että vaikka perinteiset työpajamenetelmät olivat tehokkaita yhteisen ymmärryksen luomisessa, ne eivät jättäneet aikaa yksilötyöskentelyyn ja asiaan syventymiseen. Lisäksi ratkaisujen suunnittelu kesti pitkään, koska tavoitteena oli valmis ja ”täydellinen” tuote, joka voidaan lanseerata. Kun tuote lopulta tuli käyttöön, käyttäjät käyttivät aluksi vain murto-osaa tuotteen toiminnoista. Osalle potentiaalisista käyttäjistä tuote tuntui niin monimutkaiselta, että he eivät ottaneet sitä käyttöön ollenkaan. Suuri osa suunnittelutyöstä meni siis aivan hukkaan. Jouduttuaan tilanteisiin, joissa ratkaisun

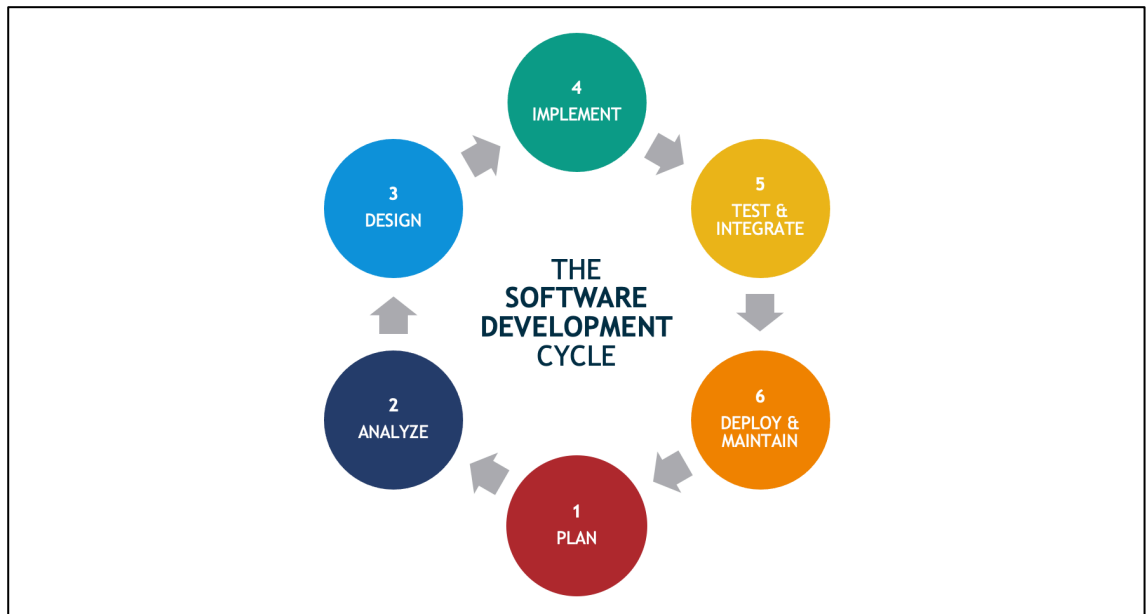
löytämiseen oli vain vähän aikaa käytettävissä, Jake Knapp huomasi, että sopivan kokoinen aikarajoite johti siihen, että kun ratkaistavan ongelman eri osapuolet työskentelivät intensiivisesti ongelman parissa, niin kokeiltava ratkaisuaihio - ei siis vielä valmis ja ”täydellinen” tuote - syntyi huomattavasti aiempaa nopeammin. (Knapp, Zeratsky & Kowitz 2016, 1-6)

Näiden huomioiden pohjalta Jake Knapp kehitti Design Sprint -konseptin, joka tuli aluksi Google Venturesin käyttöön. Design Sprint on viiden päivän mittainen systemaattinen ja intensiivinen työpajamenetelmä, jonka tavoitteena on jalostaa idea prototyyppiksi ja testata sitä käyttäjien kanssa viidessä päivässä. Nykyisin se on tuotekehityksessä ja palvelumuotoilussa yleisesti käytetty menetelmä, jota käyttävät sadat kansainväliset yritykset, kuten Google, Slack, Lego ja IDEO. (Knapp & Zeratsky, 2023)

Design Sprint on ajallisesti rajattu alun perin viiteen päivään. Näin luodaan sopiva aikataulu-paine, jotta tiimin työskentely olisi intensiivistä. Menetelmän neljä peruseriaatetta puolestaan pyrkivät tuotosten laadun parantamiseen. Ensimmäinen periaate on ”yhdessä, yksin” (eng. Together, Alone), jonka mukaisesti jokaisessa vaiheessa on aikaa perehtyä asiaan ja pohdita ratkaisuja itsenäisesti ennen ryhmän yhteistyöskentelyä. Toisen periaatteen ”konkreettinen > keskustelu” (eng. Tangible > Discussion) tavoitteena on konkretisoida asioita. Konkretisoinnin etuina on se, että se pakottaa esittäjän miettimään asiaa tarkasti, vähentää väärinymmärryksen määrää ja tarjoaa tiimin jäsenille mahdollisuuden jatkojalostaa ideaa ja antaa yksityiskohtaista palautetta. Kolmannen periaatteen ”alkuun pääseminen > oikeassa oleminen” (eng. Getting Started > Being Right) mukaisesti lopputuotoksen ei ole tarkoitus olla täydellinen vaan sen sijaan olla ensimmäinen askel kohti sellaista tuotetta, joka ratkaisee käyttäjän ongelman. Tuloksena voi myös olla se, että nyt tehty idea ei olekaan oikeanlainen ratkaisu ongelmaan. Neljäs periaate ”älä luota luovuuteen” (eng. Don't rely on creativity) on tarkoituksellisesti provokatiivinen. Se on muistutus siitä, että paras tapa innovoida on leanmallin mukaisesti testaaminen ja oppinen ja tämän saman syklin toistaminen riittävän pitkään. (Nakamura 2021)

2.4 Ohjelmistokehityksen elinkaarimalli

Ohjelmistokehityksen elinkaarimalli kuvaa uusien, teknologisesti kestävien ja asiakkaan tarpeet täyttävien ohjelmistotuotteiden tuottamismallin organisoidusti. Sen kansainvälisen standardin muodostaa Software Development Life Cycle (SDLC), joka jakaa prosessin useaan vaiheeseen ja tarjoaa selkeän viitekehityksen laadukkaiden ohjelmistotuotteiden tuotantoon, ylläpitoon ja suunnitteluun. (Distinguished 2022)



Kuvio 4: SDLC - ohjelmistokehityksen elinkaarimalli (Distinguished 2022, piirretty uudestaan)

Ohjelmistokehitys alkaa aina suunnitteluvaiheella (eng. Plan). Suunniteltavaa tuotetta koskevat vaatimukset kerätään asiakkailta, asiantuntijoilta sekä liiketoiminnalta ja näistä muodostetaan tuotteen vaatimusmäärittely. Sen pohjalta voidaan analysoida (eng. Analysis), kannattaako projektia aloittaa. Siinä työkaluina on tyypillisesti kustannus-hyötyanalyysi sekä alustavat aikataulu- ja resurssitarvearviot. (AWS)

Määrittelyvaiheessa (eng. Design) pyritään löytämään parhaat ratkaisumallit ja teknologiavaihtoehdot ohjelmistotuotteen toteuttamista varten. Toteutusvaihe (eng. Implement) on nimensä mukaisesti ohjelmistoratkaisun toteutusta ohjelmoimalla ja erilaisia teknologioita hyödyntämällä. Testaus ja integraatio (eng. Test and Integrate) limittyvät usein toteutusvaiheen kanssa. Sen aikana suoritetaan automaatio- tai manuaalitestausta ohjelmiston eri osille. Ohjelmiston eri osat viedään samaan tekniseen ympäristöön, eli integroidaan kokonaiseksi ohjelmistoratkaisuksi ja testataan koko ratkaisun toiminnallisuutta. (AWS)

Ylläpitovaiheessa (eng. Deploy & Maintain) ratkaisu viedään tuotantoympäristöön ja huolehditaan siellä pyörivän ohjelmiston virheettömästä toiminnasta. Siihen sisältyy virhekorjausten lisäksi ohjelmiston suorituskyvyn tarkkailu, turvallisuuden valvonta sekä muu ohjelmistoratkaisun kehittäminen. (AWS)

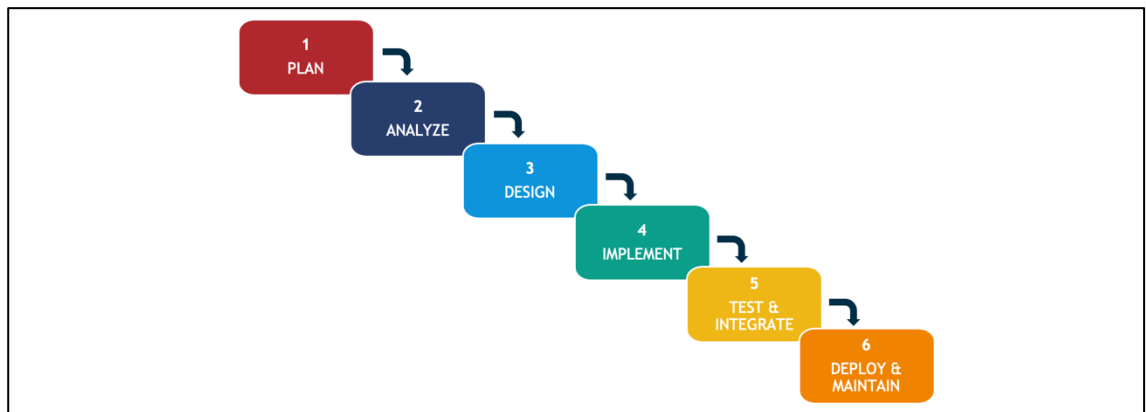
Elinkaarimallin hyötyjä ovat mm.

- ohjelmistokehitysprosessin näkyvyyden lisääntyminen eri sidosryhmille
- tehokas arviointi, suunnittelu ja aikataulutus
- kehittynyt riskien hallinta ja kustannusten arviointi

- systemaattinen ohjelmistotuotteiden toimitusmalli sekä asiakastyytyväisyyden parantuminen (AWS)

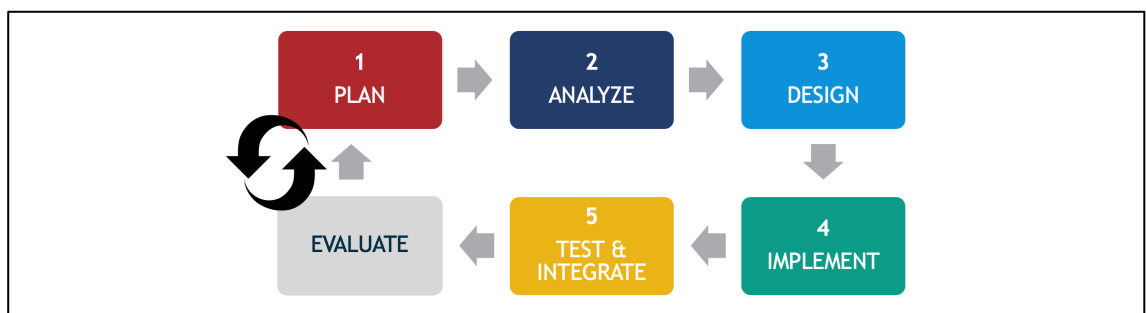
Elinkaarimallissa ohjelmistokehitysprosessin vaiheet ovat vakiot, mutta prosessia voidaan varioida vaiheiden kronologista järjestystä ja julkaistavan kokonaisuuden laajuutta muuntelemalla. Yleisiä malleja ovat vesiputousmalli, iteratiivinen malli ja agile-malli. (AWS)

Vesiputousmallissa elinkaarimallin vaiheet suoritetaan peräkkäin siten, että seuraavan vaiheen tehtävät perustuvat aiemman vaiheen lopputuotoksiin. Työ etenee vesiputousmaisesti vaiheesta toiseen. Se sopii lähinnä pieniin ohjelmistokehitysprojekteihin, joissa ratkaisun vaatimukset tiedetään tarkasti etukäteen. (AWS)



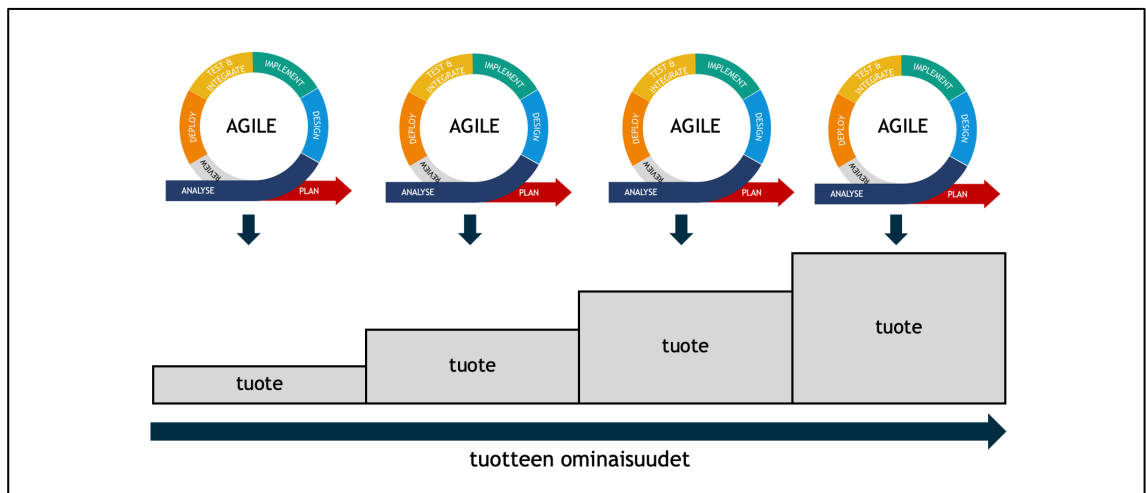
Kuvio 5: SDLC vesiputousmalli (interviewbit waterfall 2022, piirretty uudestaan)

Iteratiivisessa mallissa tiimit aloittavat toteutustyön pienen vaatimusjoukon pohjalta. Uusia ominaisuuksia lisätään kunkin iteraation yhteydessä, kunnes tuote on valmis. Kunkin iteraation päätteeksi syntyy uusi ohjelmistoversio. Tiimit parantavat versioita iteratiivisesti, kunnes koko ohjelmisto on valmis. Malli tehostaa riskien hallintaa, sillä vaatimukset voivat muuttua projektin aikana. Toisaalta toistuvat syklit voivat johtaa alkuperäisen projektin laajuuden mittavaan muutokseen ja resurssien aliarviointiin. (AWS)



Kuvio 6: SDLC iteratiivinen malli (interviewbit iterative 2022, piirretty uudestaan)

Agile-malli, eli ketterän kehittämisen malli, toistaa elinkaarimallin vaiheet useissa kehityssykleissä. Vaiheiden kestot ovat lyhyitä ja tiimi toimittaa pieniä toiminnallisuusjoukkoja kerrallaan. Vaatimuksia, suunnitelmia ja tuloksia arvioidaan jatkuvasti, jotta muutoksiin voidaan reagoida mahdollisimman nopeasti. Agile-malli on sekä iteratiivinen että inkrementaalinen ja siten prosessimalleista tehokkain. Nopeiden kehityssyklien ansiosta ongelmat voidaan tunnistaa ja ratkaista jo varhaisessa vaiheessa. Kunkin kehityssyklin päätteeksi tuotos esitellään asiakkaalle, jolloin voidaan varmistua siitä, että ratkaisu vastaa asiakkaan odotuksia. Malli mahdollistaa asiakkaan sitouttamisen projektiin koko elinkaaren ajaksi. (AWS)



Kuvio 7: SDLC agile-malli (interviewbit agile 2022, piirretty uudestaan)

2.5 Ketterä kehittäminen (Scrum)

Scrum on laajasti käytetty ketterän kehittämisen malli. Viitekehityksen tavoitteena on tarjota tuotekehityksen malli, jonka avulla voidaan varmistaa monimutkaisten tuotteiden tuottaminen mahdollisimman suuri lisäarvo. Käytännössä se tapahtuu jatkuvan syklin avulla, joka mahdollistaa toimintamallin ja tuotteen mukauttamisen tuotekehityksen aikana. (Stefanov 2020)

Taulukko 2 listaa ketterän kehittämisen (Scrum) hyödyt.

Ketterän kehittämisen (Scrum) hyödyt
Tuotteen saaminen nopeammin markkinoille ja tämän mahdollistama nopeampi tuotekehitykseen sijoitetun pääoman tuotto.
Kyky mukautua tuotekehityksen aikana muuttuvaan liiketoimintaympäristöön.
Keskittyminen pieniin lisäarvoa tuottaviin kokonaisuuksiin yksi kerrallaan, vähentää integraatoriskiä ja mahdollistaa aikaisen käyttäjäläpälyksen.

Ketterän kehittämisen (Scrum) hyödyt
Parantaa kommunikaatiota kehitystiimin ja liiketoiminnan välillä.
Parantaa näkyvyyttä ja ennustettavuutta
Haasteet voidaan ratkaista sitä mukaa kun niitä tulee vastaan. Näin voidaan varmistaa, että julkaistava tuote on laadukas ja vastaa käyttäjien tarpeisiin.

Taulukko 2: Ketterän kehittämisen (Scrum) hyödyt (Stefanov 2020)

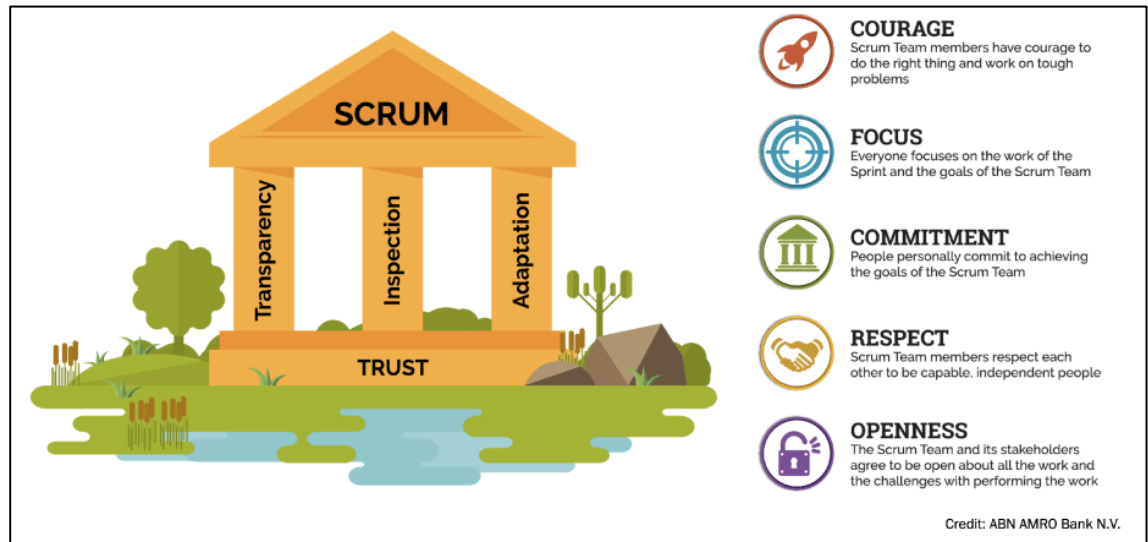
Käytännössä Scrum on viitekehys, jossa tiimi suorittaa kehitysprosessin alusta loppuun siten, että työ on vaiheistettu ajallisesti pieniin kokonaisuuksiin. Prosessia ohjaa jatkuva kokeilu ja palaute, jonka avulla tiimi oppii ja parantaa toimintatapoja työn edetessä. Toimintamallin tavoitteena on auttaa tiimin jäseniä tuottamaan lisäarvoa yhteisvastuullisesti ja inkrementaalisesti. Ketterä kehittäminen on iteratiivinen (asteittain tarkentuva) ja inkrementaalinen (lisäävä) lähestymistapa ennustettavuuden optimointiin ja riskien hallintaan. Ketterän kehityksen viitekehysistä Scrum on yksinkertaisin ja tarjoaa riittävän rakenteen tiimin työskentelyyn ja optimaalisten käytäntöjen luomiseen. (Schwaber & Sutherland 2020, 3)

Scrumin taustalla on lean-ajattelu ja empirismi. Lean-ajattelu keskittyy välttämättömien asioiden tekemiseen ja vähentää hukkaa. Empirismi puolestaan lähtee siitä, että tieto syntyy kokemuksesta ja päätösten tekemisestä havaintojen pohjalta. Tavoitteena on luoda ryhmiä, joilla yhdessä on kaikki osaaminen ja asiantuntemus annetun työn tekemiseksi sekä kyky jakaa tai hankkia tarvittavia taitoja. (Schwaber & Sutherland 2020, 4)

Ketterän kehittämisen mallissa käyttäjäkokemusta varmistetaan lean-ideologian mukaisesti Minimum Viable Product (MVP) -toteutuksen avulla. MVP on sellainen, mahdollisimman nopeasti ja pienellä työmäärällä toteutettava versio tuotteesta, joka voidaan julkaista tuleville käyttäjille palautteen saamiseksi. Käyttäjätestauksen avulla kerätään laadullista (esim. pitävätkö käyttäjät tuotteesta vai eivät) ja määrällistä (esim. kuinka moni ottaa sen käyttöön ja jatkavat sen käyttöä) palautetta. (Ries 2011, 75-77).

Lean-ajattelu ja empirismi luovat Scrumille vahvan ideologisen perustan, jonka empiirisiä peruspilareita ovat läpinäkyvyys (eng. Transparency), tarkastelu (eng. Inspection) ja mukauttaminen (eng. Adaptation). Tiimin onnistumisen kannalta on tärkeää viiden perusarvon omaksuminen: sitoutuminen (eng. Commitment), keskittyminen (eng. Focus), avoimuus (eng. Openness), kunnioitus (eng. Respect) ja rohkeus (eng. Courage). Käytännössä tämä tarkoittaa sitä, että tiimin jäsenet sitoutuvat tiimin tavoitteisiin sekä toistensa tukemiseen. Tiimi keskittyy ensisijaisesti sprintin työhön. Tiimin jäsenet ja sidosryhmät ovat avoimia työhön liittyvissä asioissa ja haasteissa. Tiimin jäsenten ja yhteistyökumppaneiden välillä vallitsee keskinäinen

kunnioitus kyvykkäinä ja itsenäisinä ihmisinä. Kaikilla tiimin jäsenillä on rohkeutta ratkoa hankaliakin ongelmia ja tehdä oikeaksi katsomiaan asioita. (Schwaber & Sutherland 2020, 4-5)



Kuvio 8: Scrumin peruspilarit ja arvot (Scrum.org 2023)

Scrum-tiimi on monialainen ja itseohjautuva tiimi. Siihen kuuluu Scrum Master, tuotemistaja ja kehittäjiä. Sen sisällä ei ole alaryhmiä tai hierarkioita. (Schwaber & Sutherland 2020, 6)

Scrumin tuotokset ovat asioita, joita tiimin tulee toteuttaa. Ne edustavat työtä ja arvoa. Kukin tuotos on sidoksissa tehtävän työn tavoitteisiin siten, että se keskittyy mitattavaan edistymiseen. Sidokset tukevat havaintoihin perustuvaa empiiristä toimintaa sekä Scrum-arvojen toteutumista. Näitä ovat tuotteen kehitysajon, sprintin kehitysajon sekä inkrementti. (Schwaber & Sutherland 2020, 10)

Tuotteen kehitysajon on vähitellen syntyvä, järjestetty lista siitä, mitä tarvitaan tuotteen parantamiseksi. Se on ainoa lähde Scrum-tiimissä tehtävälle työlle. Tuotteen kehitysajon on sidottu tuotteen tavoitteeseen. (Schwaber & Sutherland 2020, 10-11)

Sprintin kehitysajon on kehittäjien itselleen tekemä suunnitelma siitä, miten inkrementti - eli sprintin sisältö - voidaan toteuttaa. Se koostuu sprintin tavoitteiden mukaan valituista tuotteen kehitysajon kohdista. Sprintin kehitysajon on siis sidottu sprintin tavoitteeseen. (Schwaber & Sutherland 2020, 11)

Jokainen inkrementti on lisäys aiemmin tuotettuun sisältöön ja siten aina askel kohti lopullisen tuotteen tavoitteita. Inkrementti syntyy, kun kehitysajon jokainen kohta täyttää valmiin määritelmän. Valmiin määritelmä sisältää tuotteelle asetetut laatuvaatimukset. Inkrementti on näin sidottu valmiin määritelmään. (Schwaber & Sutherland 2020, 11-12)

Scrum-prosessi koostuu tapahtumista. Tapahtumat tarjoavat muodollisen mahdollisuuden tutkia ja mukauttaa tekemistä. Scrumin tärkein tapahtuma on sprintti, joka sisältää muut tapahtumat. Sprintin sisäisiä tapahtumia ovat suunnittelu, päivittäispalaveri, katselmointi ja retrospektiivi. (Schwaber & Sutherland 2020, 7)

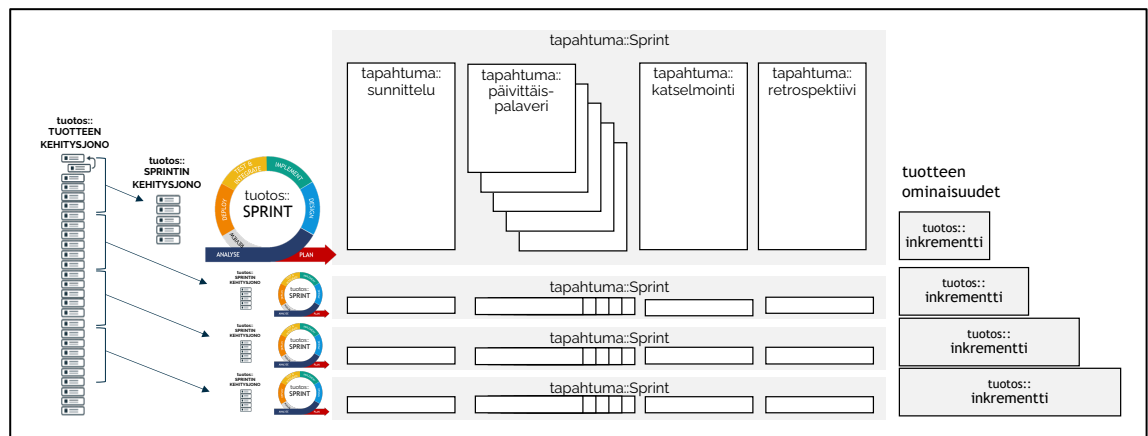
Suunnittelu on kahden tunnin mittainen kokous, joka tapahtuu aina sprintin alussa. Kehitystiimi sopii yhdessä, mitkä toiminnallisuudet seuraavassa voidaan toteuttaa ja sitoutuu niiden toteuttamiseen. Näistä muodostuu kyseisen sprintin kehitysjono. Kokouksen lopussa tiimillä on selkeä suunnitelma siitä, miten sprintin tavoite tullaan saavuttamaan. (Stefanov 2020)

Päivittäispalaveri on kehitystiimin sisäinen kokous ja tapahtuu nimensä mukaisesti joka päivä - mieluiten päivän aluksi - ja on kestoltaan korkeintaan 15 minuuttia. Jokainen tiimin jäsen vastaa kolmeen kysymykseen: Mitä olet tehnyt edellisen kokouksen jälkeen? Mitä aiot tehdä ennen seuraavaa kokousta? Onko olemassa jotain esteitä, jotka haittaavat tavoitteen saavuttamista? (Stefanov 2020)

Katselmointi tapahtuu aina sprintin lopussa. Sen aikana kehitystiimi esittelee sprintin toteutuksen asiakkaalle ja sidosryhmille. Katselmoinnissa kerätty palaute viedään osaksi tuotteen kehitysjonoa ja toivotut muutokset tulevat osaksi jatkototeutusta priorisoinnin mukaisessa järjestyksessä. (Stefanov 2020)

Retrospektiivi päättää sprintin. Sen aikana tiimi käy läpi päättyneen sprintin aikaisia toimintamalleja: mikä toimi, mikä ei toiminut ja mitä halutaan muuttaa. Tämän pohdinnan pohjalta tiimin toimintaa mukautetaan seuraavassa sprintissä. Näin mahdollistetaan jatkuva kehittyminen. (Stefanov 2020)

Sprintit ovat Scrumin pulssi, joissa lisäarvon muodostus tapahtuu. Sprintin pituus on vakio ja se voi olla enintään kuukauden mittainen. Uusi sprintti alkaa heti edeltävän sprintin päättyttyä. Kunkin sprintin lopussa tarkastellaan työn etenemistä kohti tavoitteita ja tarvittaessa mukautetaan toimintamallia. Vähintään kuukausittain tapahtuva tarkastelu ja mukauttaminen parantavat ennustettavuutta. (Schwaber & Sutherland 2020, 7)



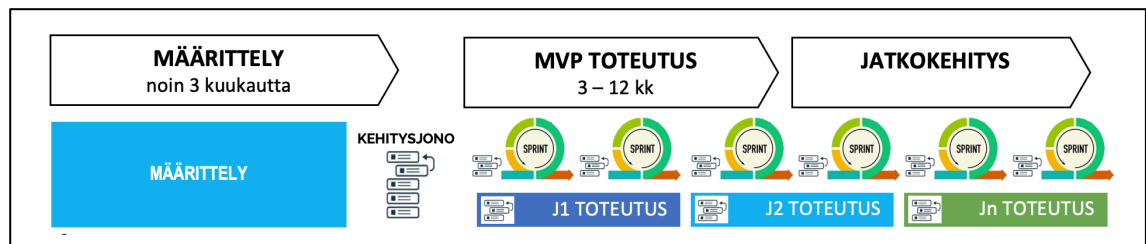
Kuvio 9: Scrum prosessin tapahtumat ja tuotokset

3 Kehittämisasetelma

Tässä kehittämistehtävässä pohditaan sitä, miten palvelumuotoilun ja muotoilumenetelmien avulla voidaan varmistaa asiakkaan tarpeita palvelevan konseptin eheys koko tuotekehityssyklin ajan. Lisäksi tarkastelemme, millaisin edellytyksin voidaan tehdä aidosti ketterää ohjelmistokehitystä. Tällä tarkoitetaan sitä, että myöskään määrittelytyö ei mene hukkaan: määritellään tarkasti vain ne kehitysjonon tehtävät, jotka ovat tulossa toteutukseen. Vaikka ketterän kehittämisen mallien myötä tilaajalta saadaan palautetta toteutetusta järjestelmästä vähintään kerran kuukaudessa, halutaan tätä aikaistaa muotoilun keinoin prototyyppien avulla jo ennen toteutuksen aloittamista. Kehitettävästä mallista käytetään tässä nimeä Muotoiluohjautuva ketterä projektimalli.

3.1 Ohjelmistotuotantoprosessin nykytila

Toimeksiantajan ohjelmistotuotannossa on vallalla ohjelmistotuotantoprosessi, jossa tarkka tulevan järjestelmän määrittely tehdään ennen toteutusvaihetta, riippumatta siitä, ohjaintaanko toteutusvaihetta vesiputouksmallin vai ketterän kehittämisen toimintamallin mukaisesti. Siinä tuotteen kehitysjono muodostetaan määrittelyvaiheen aikana, jolloin määritellään suunniteltavan ratkaisun toiminnallisuudet / ominaisuudet sekä tekninen ratkaisuperusta. Lopputuotoksena on tuotteen tekninen arkkitehtuuri ja kehitysjono, joka tyypillisesti priorisoidaan ainakin kehityspolkutasolla ennen toteutusprojektin aloittamista.

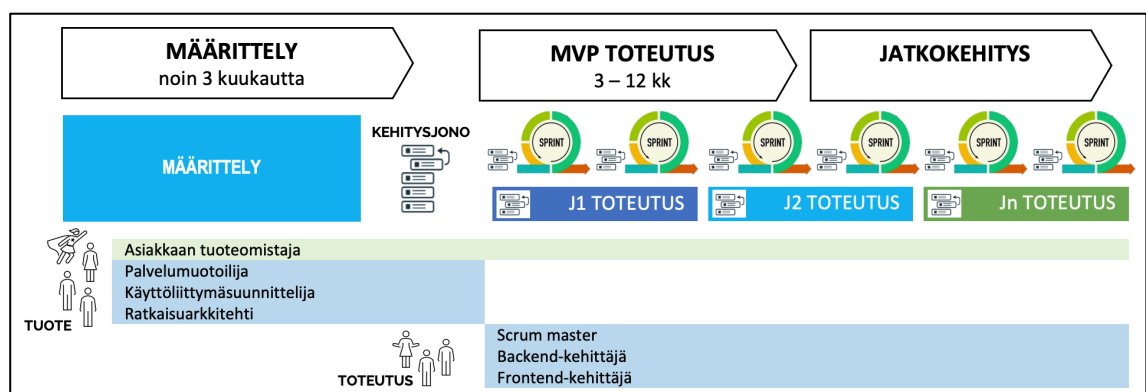


Kuvio 10: Nykyisin vallalla olevan ohjelmistotuotannon vaiheet

Määrittelyvaiheen avainroolit ovat palvelumuotoilija, käyttöliittymäsuunnittelija ja järjestelmäarkkitehti sekä tilaajan edustaja. Toiminnallisen määrittelyn tekee palvelumuotoilija ja käyttöliittymäsuunnittelija ja teknisen määrittelyn arkkitehti. Nämä tapahtuvat usein rinnakkain.

Toteutusvaiheessa ketterässä tiimissä on tyypillisesti arkkitehti, frontend-kehittäjä, backend-kehittäjä, Scrum Master ja asiakkaan tuoteomistaja. Tarvittaessa voidaan käyttää vierailevia asiantuntijoita. Laskutuksesta ja resursoinnista huolehtii hallinnollinen projektipäällikkö. Asiakkaalle myydään yleensä tiimin resurssit heille osoitetulla allokaatiolla. Kaikesta lisätyöstä ja lisäresursseista on sovittava asiakkaan kanssa erikseen. Määrittelyvaiheen ja toteutusvaiheen välinen jatkumo on pääosin asiakkaan tuoteomistajan vastuulla.

Myynti tapahtuu pitkälti kokonaistymääräarvion mukaisesti per rooli ja toteutusvaiheen toimitus on sidottu määriteltyyn tuotteeseen. Tämä edellyttää varsin tarkkaa määrittelyä, jotta toteutusvaiheen kokonaistymääräarviosta voidaan antaa luotettava arvio. Tämä johtaa myös siihen, että projektinaikainen muutostenhallinta on raskasta, eikä se ole lähtökohtaisesti ketterän kehittämisen mukaista. Usein asiakas haluaisi päästä aloittamaan toteutuksen mahdollisimman pian ja muotoilun myyminen osana pitkää määrittelyä on haastavaa.



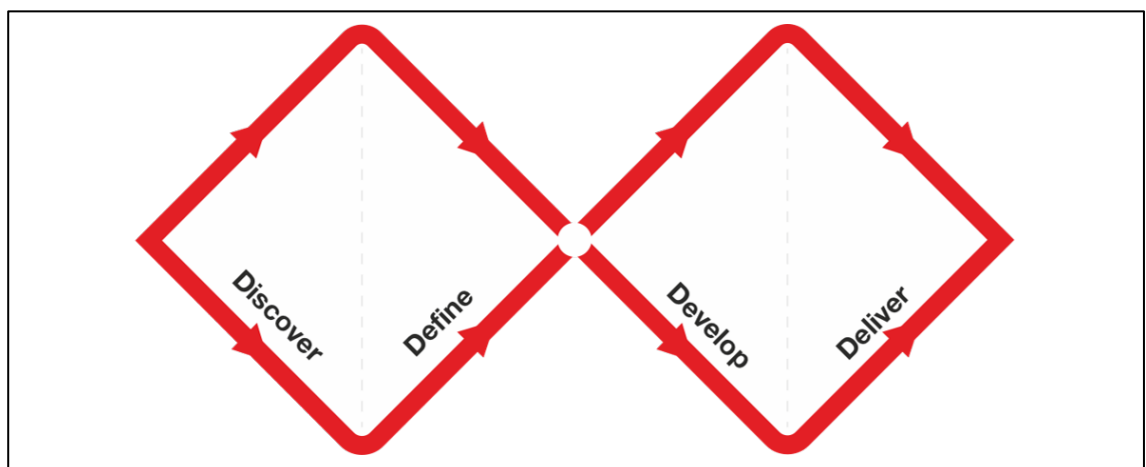
Kuvio 11: Nykyisin vallalla olevan ohjelmistotuotantoprosessin vaiheet ja roolit

Määrittelytyö ei sisälly toteutustiimin ketterän kehittämisen kokouksiin. Menetelmässä tehdään jatkuvaa sisällön priorisointia, mutta toimintaympäristön muutoksiin vastaaminen vaatisi uudelleenmäärittelyä.

Nykyisen toimintamallin, jossa määrittely tapahtuu toteutuksesta erillään, uskotaan olevan juurisyytä siihen, että asiakkaalle toimitettu ohjelmistoratkaisu ei aina vastaa alkuperäiseen ongelmaan eikä siten myöskään vastaa asiakkaan odotuksia. Muotoilupuoella on vahva visio siitä, että muotoilun tulisi olla digitaalisten palveluiden kehitysprojektissa jatkuvaa toimintaa. Tällainen toimintamalli tukisi myös muotoilijoiden kokemuksellista oppimista, kun oman ratkaisun sopivuutta lopulliseen ratkaisuun on mahdollista kehittää koko toteutusvaiheen ajan. Erittäin tärkeänä nähdään myös tilanteet, jossa asiakkaan ja markkinoiden toimintaympäristö muuttuu toteutusprojektin aikana ja lopputuotteeseen kohdistuu uusia vaatimuksia. Nämä tulisi tutkia samalla tavalla kuin alkuperäisetkin vaatimukset ja pohtia niiden ratkaisua osana kokonaiskonseptia. Tällä hetkellä useimmat muutokset jäävät ohjelmistokehittäjien ratkaistavaksi, jolloin ne ovat usein pistemäisiä ja teknologialähtöisiä.

3.2 Kehittämissuunnitelma

Työn tavoitteena on luoda helposti sovellettava malli siitä, miten muotoilumenetelmien tukena voidaan tuottaa käyttäjälähtöisiä ratkaisuja ketterän kehittämisen mallilla. Näin ollen luonteva valinta kehittämissuunnitelman rungoksi on British Design Councilin esittelemä Tuplatimantti-malli, joka kuvaa yleisellä tasolla minkä tahansa suunnittelu- tai innovaatioprojektin vaiheet. Mallin ensimmäinen timantti muodostuu asetetun ongelman ymmärtämisestä ja toinen timantti puolestaan keskittyy ratkaisun kehittämiseen. (Design Council 2023) Molemmat timantit koostuvat aina divergentistä ja konvergentistä vaiheesta. Divergentissä vaiheessa asiaan tutustutaan laaja-alaisesti ja konvergentissa vaiheessa suoritetaan analyttinen valinta.



Kuvio 12: Tuplatimantti-malli (Design Council 2023)

Tutki-vaihe (eng. Discover) keskittyy ongelman ymmärtämiseen olettamisen sijaan. Käytännössä tämä tarkoittaa ajan viettämistä niiden ihmisten kanssa, joihin ongelma vaikuttaa. (Design Council 2023) Näin tunnistetaan seikat, joita tulee tutkia tarkemmin ja kerätään tietoa niiden ympäriltä: millaisia trendejä maailmalla on, miten kilpailijat saman asian ratkaisevat ja mihin suuntaan maailma on menossa (Design Council 2015, Discover).

Ensimmäisessä vaiheessa kerätty tieto tarjoaa uusia näkökulmia ja auttaa määrittelemään ongelman toisella tavalla (Design Council 2023). Määritä-vaihe (eng. Define) keskittyy kerätyn tiedon jäsentämiseen sekä tiedon sovellettavuuden selvittämiseen organisaatioissa. Tavoitteena on laajentaa kokonaiskuvaa haasteesta ja löytää piilevät asiakastarpeet ja/tai haasteen ratkaisemiseen vaikuttavat seikat. Syvempää ymmärrystä ja todellisia tarpeita luodaan laadullisilla ja luovilla menetelmillä. (Innanen 2018)

Kehitä-vaiheen (eng. Develop) tavoitteena on löytää tarkennettuun ongelmaan ratkaisuja. Inspiraatiota haetaan ongelman ulkopuolelta ja uusia ideoita kehitetään erilaisista näkökulmista. (Design Council 2023). Tavoitteena on jalostaa kokeilukelpoisia ratkaisuja ja hakea niihin palautetta mahdollisimman aikaisessa vaiheessa. Voidaan myös sopia osittaisista kokeiluista, mikäli se on mahdollista järjestää. (Innanen 2018)

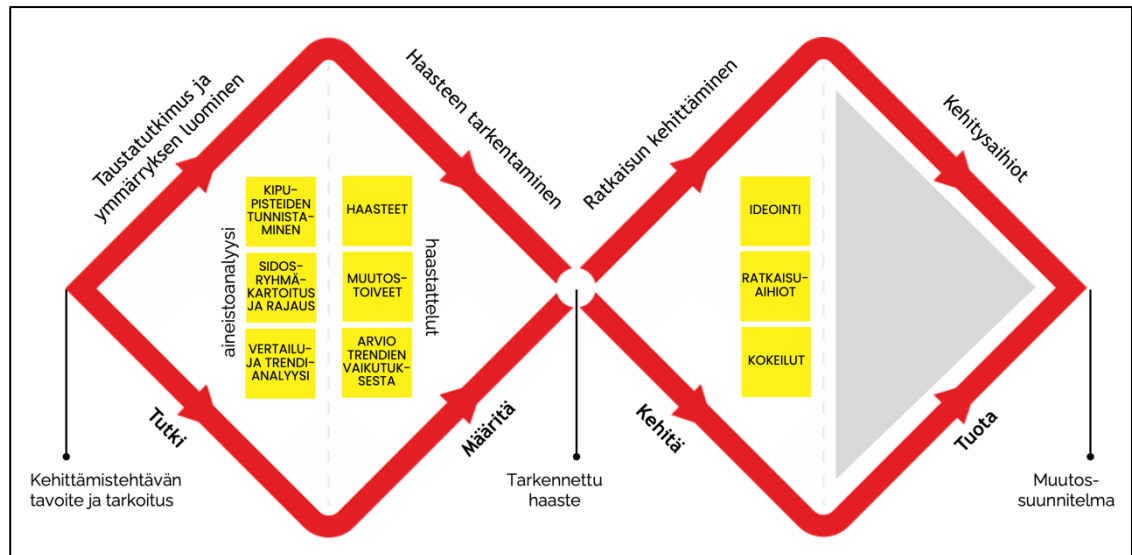
Viimeisessä vaiheessa (eng. Deliver) ratkaisumalleja testataan pienimuotoisesti. Toimimattomista ratkaisuksista luovutaan ja hyviksi todettujen mallien kehittämistä jatketaan. (Design Council 2023)

Tähän kehittämistehtävään tuplatimanttia sovelletaan seuraavasti:

- Taustatutkimus ja ymmärryksen luominen:
Tässä kipupisteet ja sidosryhmät tunnistetaan toimeksiantajan alkuhaastattelun avulla ja samalla sovitaan tehtävän tarkemmasta rajauksesta. Taustatutkimusta tehdään tutustumalla tarkemmin esiin tulleiden haasteiden juurisyihin sekä siihen, miten niitä on ratkottu muissa organisaatioissa.
- Haasteen tarkentaminen:
Menetelmäksi valittiin haastattelut, joissa käytiin läpi ensimmäisessä vaiheessa kootua aineistoa ja pohdittiin sen hyödyntämisestä eri osa-alueilla. Haastateltavat henkilöt valittiin rajauksen mukaisista organisaatioista.
- Ratkaisun kehittäminen:
Ratkaisuehdotukset ja kokeiluaihiot ovat syntyneet pitkälti edellisen vaiheen haastattelujen jatkotyöskentelynä. Toimintamalleja on joiltain osin päästy soveltamaan myös käytäntöön muutamissa projekteissa.

Viimeinen vaihe (eng. Deliver) on rajattu kehittämistehtävän ulkopuolelle. Kehittämistehtävän tuotokset toimitetaan yrityksen toiminnan kehittämisestä vastaavalle organisaatiolle jatkotoimenpiteiden suunnittelun tueksi.

Kuvio 13 kuvaa kootusti, miten tuplatimantti-mallia sovelletaan tähän kehittämistehtävään.



Kuvio 13: Tuplatimantin soveltaminen tässä kehittämistehtävässä

3.3 Tutki-vaihe: taustatutkimus ja ymmärryksen luominen

Vaiheen tavoitteena on saada mahdollisimman hyvä liikkeellelähtö projektille (Innanen 2018). Tässä kehittämistehtävässä päätettiin edetä organisaation haasteiden ja sidosryhmien rajaamisen kautta tutkimaan sitä, miten haasteita on muualla ratkottu. Vaiheet ja tutkimusmenetelmät ovat siten seuraavat:

- Digitaalisten palveluiden tuottamiseen liittyvien kipupisteiden tunnistaminen tehdään haastattelemalla kehittämistehtävän toimeksiantajan edustajaa.
- Sidosryhmäkartoitus ja rajaus suoritetaan pöytätyöskentelyn avulla yhdessä kehittämistehtävän toimeksiantajan edustajan kanssa.
- Vertailu- ja trendianalyysi tehdään pöytätyöskentelyä käyttäen tunnistettujen kipupisteiden ja niiden ratkaisumallien ympäriltä. Myynnin ja asiakkaan ostomotivaation tukemiseksi etsittiin faktatietoa siitä, millainen merkitys asiakaskeskeisyydellä on yrityksen liiketoiminnalle. Muotoilun ja ohjelmistokehitysprojektin osalta selvitettiin alan toimijoiden näkemyksiä muotoilun ja ohjelmistokehityksen kokonaisvaltaisuudesta sekä alalla syntyneitä parhaita käytäntöjä.

3.3.1 Kipupisteiden tunnistaminen

Kehittämistehtävän tavoitteen tarkentamiseksi pidettiin toimeksiantajan kanssa alkuhaastattelu. Menetelmäksi valittiin strukturoimaton, avoin haastattelu. Se on keskustelunomainen tilanne valitun aiheen ympärillä ja tarjoaa mahdollisuuden mennä syvällä aiheen käsittelyssä. (Hirsjärvi & Hurme 2001, 38). Toimeksiantaja edustaa organisaation muotoilutoimintaa, joten kipupisteitä lähestyttiin muotoilutoiminnan näkökulmasta.

Haastattelussa korostui muotoilutoiminnan irrallisuus organisaation muusta toiminnasta. Tämä konkretisoituu uusien projektien myynnissä ja resursoinnissa, projektikäytännöissä sekä organisaation toimintamallia kuvaavissa ydinprosessien kuvauksissa. Liiketoiminnan tavoitteena on, että muotoilu olisi kiinteä osaa yrityksen toimintaa projektien toteutuksen ja resursoinnin, myynnin ja virallisen toimintamallin näkökulmasta.

Tämän pohjalta muodostettu tutkimuskysymys kuuluu: **”Kuinka voisimme tuoda muotoilutoiminnan kiinteäksi osaksi organisaation toimintamallia sekä ohjelmistokehityksen että myynnin näkökulmasta?”**

Alkuhaastattelun aikana kipupisteitä ja niihin liittyviä ilmiöitä syvennettiin seuraavasti:

- Ydinprosesseissa muotoilu kuvataan irralliseksi toiminnaksi.

Ydinprosessit toimii käsikirjana uusien projektien käynnistämisvaiheessa sekä tarjousten tekemisen apuna. Projektin toimintaa myös tarkastellaan ydinprosessien valossa. Tavoitteena on yhdenmukaistaa projektien käytäntöjä ja maastouttaa uusia toimintamalleja. Tällä hetkellä ydinprosesseissa muotoilutoiminta kuvataan erilliseksi toiminnaksi eikä se nivoudu luontevaksi osaksi kokonaisprosessia.

- Muotoilu on pistemäistä ja erillään ketterän tiimin toiminnasta.

Ketterä tiimi keskittyy pitkälti perinteisen vesiputousmallin mukaisesti ratkaisun toteutukseen jättäen muotoilutoiminnan pisteittäiseksi kertasuorituksiksi ennen toteutuksen aloittamista. Muotoilutoiminnalla tarkoitetaan tässä palvelukonseptin luomista sekä käyttöliittymäsuunnittelua.

- Projektin aikana tulevat muutokset rikkovat käyttökokemuksen.

Ohjelmistoprojektit ajautuvat ajoittain tilanteeseen, jossa asiakkaan tarpeet muuttuvat projektin aikana ja uusien tarpeiden sisällyttäminen jo suunniteltuun kokonaisuuteen tapahtuu pääosin teknisestä näkökulmasta asiakas- ja käyttäjäkokemuksen kustannuksella.

- Asiakkaan on vaikea ostaa muotoilua irrallisena.

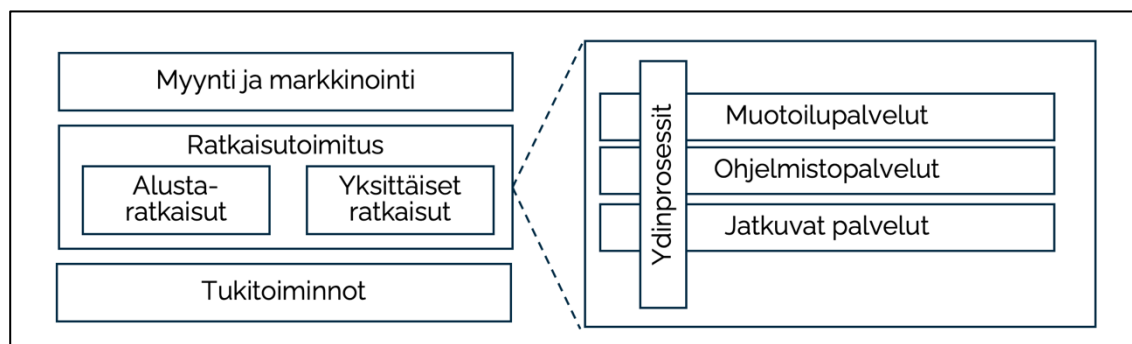
Myynti kokee muotoilutoiminnan myymisen irrallisena lisänä projektiin haasteelliseksi. Muotoilu olisi helpompi myydä kiinteänä osana projektia.

3.3.2 Sidosryhmäkartoitus ja rajaus

Sidosryhmäkartoitus ja rajaus tehtiin alkuhaastattelussa muotoutuneen tutkimuskysymyksen sekä kipupisteistä kerätyn syvemmän tiedon pohjalta.

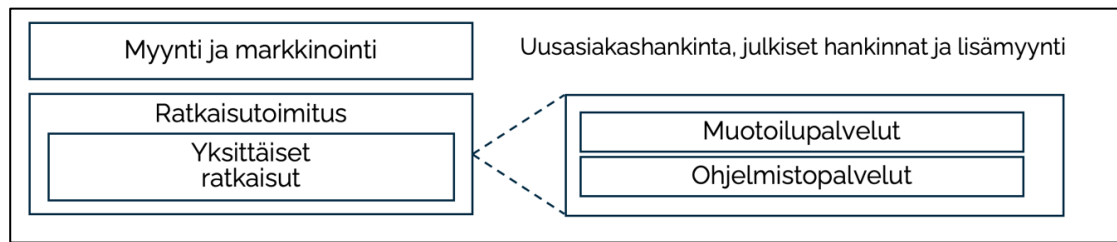
Kehittämistehtävä rajattiin ensin liiketoiminnan näkökulmasta koskemaan uusohjelmistotuotantoa ja sen myyntiä. Sen jälkeen sidosryhmäkartoitus tehtiin niihin liittyvän organisaatiokenteen pohjalta siten, että valittiin ne organisaation osat ja asiakassegmentit, joita alkuhaastattelussa tunnistetut kipupisteet eniten koskettavat.

Toimeksiantajaorganisaation toiminta on laajaa ja organisaation rakenne on tässä esitetty vahvasti yksinkertaistaen. Yrityksen organisaation osat ovat yleistasolla Myynti ja markkinointi, Ratkaisutoimitus ja Tukitoiminnot. Ratkaisutoimitus jakautuu Alustaratkaisuihin ja Yksittäisiin ratkaisuihin. Ratkaisutoimituksen sisällä on Muotoilupalvelut, Ohjelmistopalvelut ja Jatkuvat palvelut, joiden prosessit on kuvattu Ydinprosesseissa.



Kuvio 14: Yrityksen organisaatio yleisellä tasolla

Tämä kehittämistehtävä rajattiin koskemaan Yksittäisten ratkaisujen muotoilu- ja ohjelmistopalveluja sekä niihin liittyvää myyntiä uusasiakashankinnan, julkisten hankintojen ja lisämyynnin osalta. Samalla rajattiin tehtävän ulkopuolelle tuplatimantin viimeinen vaihe eli varsinaisen kehityssuunnitelman valmistelu. Kehittämistehtävän tuotokset toimitetaan yrityksen toiminnan kehittämisestä vastaavalle organisaatiolla, jonka tehtävänä on päättää, ryhdytäänkö tehtyjen havaintojen pohjalta laajempiin kehittämistoimiin.



Kuvio 15: Kehittämistehtävän rajaus

3.3.3 Asiakaskeskeisyyden merkitys kiihtyvässä digitalisaatiossa

Alkuhaastattelussa muotoiltu tutkimuskysymys oli: **”Kuinka voisimme tuoda muotoilutoiminnan kiinteäksi osaksi organisaation toimintamallia sekä ohjelmistokehityksen että myynnin näkökulmasta?”** Vertailu- ja trendianalyysin tarkoituksena on selvittää, onko kilpailijoilla vastaavaa tarvetta ja jos on niin miksi sekä se, onko kyseessä mahdollisesti jokin nouseva trendi, jolla on tulevaisuuden kannalta oleellista merkitystä. Muotoilun tavoitteena on löytää ratkaisuja, jotka vastaavat ihmisten tarpeisiin (Brown 2008, 4). Niinpä lähdimme ensin etsimään vastausta asiakaslähtöisyyden näkökulmasta.

Trendianalyysissä nousi vahvasti esille 2020 Covid-19 pandemian vaikutus digitaalisiin palveluihin ja asiakaslähtöisyyden korostumiseen. Pandemian leviämisen vuoksi asetetut laajat kohtaamis- ja kokoontumiskiellot johtivat siihen, että digitaaliset palvelukanavat jäivät yritysten pääasiallisiksi kanaviksi tavoittaa asiakkaat. Myös fyysisten tuotteiden, kuten ruuan ja päivittäistavaroiden, ostaminen siirtyi digitaalisiin kanaviin. Tuotteiden jakelussa siirryttiin kontaktittomiin toimituksiin, mikä sekä vaati toimiakseen erilaisia digitaalisia palveluja. (Deloitte 2020)

McKinseyn johdon kyselytutkimus osoitti, että pandemia on vauhdittanut digitaalisten teknologioiden käyttöönottoa useilla vuosilla, kansainvälisesti kolmella vuodella kaikkien tuotteiden osalta, ja digitaalisten tuotteiden osalta jopa seitsemällä vuodella (McKinsey & Company 2020).

Vaikka pandemian aikana tehtiin nopeita, väliaikaisia ratkaisuja, pitivät vastaajat pandemian vaikutusta digitaalisiin palveluihin pysyvänä. Siten se on johtanut uudenlaiseen strategiseen ajatteluun, missä teknologia nähdään kriittisenä liiketoiminnan onnistumisen elementtinä, kun sitä aiemmin tarkasteltiin pääosin kustannustehokkuuden näkökulmasta. (McKinsey & Company 2020)

Pandemian myötä on ymmärretty, mitä digitalisuus todella tarkoittaa. Siinä ei ole kyse vain hienoista sovelluksista, vaan sen sijaan kokonaisratkaisusta, jossa yhdistyy prosessit, ihmiset

ja teknologia. Se on vaikuttanut perustavanlaatuisesti siihen, miten me työskentelemme, elämme ja harjoitamme liiketoimintaa. (Deloitte 2020)

Pärjätäkseen pandemian jälkeisessä maailmassa, on yritysten sopeuduttava pandemian aiheuttamiin muutoksiin. Koska digitaaliset palvelut ovat yleistyneet räjähdysmäisesti, korostuu käyttäjien keskuudessa palveluiden merkityksellisyys ja käytettävyys. Tämän vuoksi yritysten tulee menestyäkseen luoda entistä merkityksellisempiä palveluja. Tämä tarkoittaa sitä, että yritysten tulee ymmärtää asiakkaidensa todelliset tarpeet, soveltaa niitä toimintaan ja luoda organisaatioon todellinen ymmärrys asiakkaiden sitouttamisen elementeistä. (Deloitte 2020)

Kilpailuetu digitalisaatiossa syntyy tiedon, teknologian ja organisaation kombinaatiosta. Deloitte listaa kolme haastetta, joihin yritysten tulisi keskittyä asiakkaiden sitouttamiseksi.

- Toimenpiteitä tukevan asiakasymmärryksen luominen asiakastiedosta

Tiedolla ei ole merkitystä, jollei sitä voida hyödyntää tavoitteita tukeviin toimenpiteisiin. Tämä vaatii organisaation sisäisen ja ulkopuolisen tiedon yhdistelyä sekä kurinalaisuutta tiedon laadun ja etiikan suhteen. Näiden avulla tulee johtaa yrityksen markkinointitoimenpiteitä. (Deloitte 2020)

- Asiakkaiden sitouttaminen

Brändin tärkein tavoite on ihmisten auttaminen, vuorovaikutus ja rakastettavien kokemusten tarjoaminen. Tämä edellyttää teknologian ja tiedon tasapainoa inhimillisten arvojen ja identiteetin suhteen. Sen vuoksi pitkäaikaisen sitoutumisen perustan luominen on vaikeaa ja se vaatii kyvykkyyttä muuntaa asiakasymmärrys teoiksi, joilla on merkitystä ja ovat yleismaailmallisia arvoja tukevia. Sillä kaikki mitä brändi tuottaa, koetaan henkilökohtaiseksi. Digitaalisuus on osa uutta normaalia ja markkinoilla tulevat voittamaan ne brändit, joilla on kyky tuottaa pysyvä ja merkityksellinen suhde asiakkaisiin. (Deloitte 2020)

- Toimituskykyisen organisaation rakentaminen

Pelkkä tieto ja teknologia eivät riitä, vaaditaan myös riittävät valmiudet, taidot ja asiakaslähtöinen organisaatio, jotta asiakkaiden sitouttamiseksi suunnitellut toimenpiteet toteutuvat ja tuottavat toivotun tuloksen. Kyse on toimintaketjusta, jossa asiakastieto kerätään ja analysoidaan tavoitteita tukevien toimenpiteiden suunnittelemiseksi sekä asiakaslähtöisen kulttuurin omaavasta palveluorganisaatiosta, joka viime kädessä huolehtii henkilökohtaisen yhteyden rakentamisesta ja ylläpitämisestä. Ilman

syvään juurrutettua asiakaslähtöistä yrityskulttuuria, jäävät panostukset asiakastietoon ja teknologiaan tuloksettomiksi. (Deloitte 2020)

3.3.4 Muotoilumenetelmät ketterän kehittämisen tukena

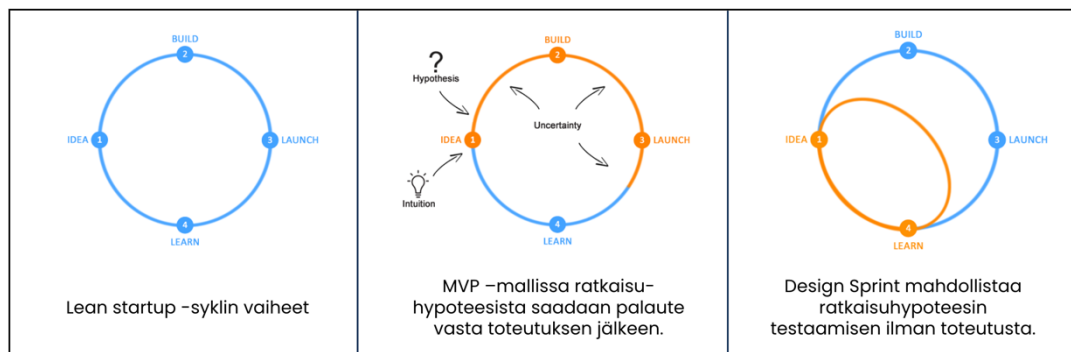
Ensimmäinen analyysin vaihe osoitti selkeästi, että asiakaslähtöisyys on tulevaisuudessa liiketoiminnan onnistumiselle jopa tärkeämpi kuin osasimme alun perin aavistaakaan. Niinpä seuraavaksi tuli selvittää, miten muotoilutoiminta voidaan tuoda kiinteäksi osaksi organisaation toimintamallia. Rajauksen mukaisesti tässä keskityttiin vain uusien digitaalisten palveluiden tuottamiseen. Vertailu- ja trendianalyyseissa selvitettiin sitä, miten muut toimijat ovat ratkaisseet asian ja onko jollekin ohjelmistokehitysprosessin osa-alueelle jo mahdollisesti syntynyt parhaita käytäntöjä.

Ratkaisujen osalta esiin nousi uusien digitaalisten palveluiden luomisen nopeus ja käyttäjälähtöisyys. Ketterät menetelmät ovat vakiinnuttaneet asemansa ohjelmistokehityksessä. Ne mahdollistavat nopean ohjelmistotuotannon ja ajankohtaisia tarpeita vastaavien tuotteiden kehittämisen. Mutta se ei enää riitä. Digitalisaation kiihdyttyä entisestään pandemian jälkeisillä kompleksisilla markkinoilla on paljon digitaalisia palveluja, ja käyttökokemuksesta on tullut yksi tärkeimmistä menestystekijöistä ohjelmistoliiketoiminnassa. (Bestman 2019)

Digitaalisten palveluiden yleistyttyä, käyttäjät eivät enää ole halukkaita käyttämään sellaisia digitaalisia palveluja, joiden ymmärtäminen on hankalaa ja käyttäminen vaikeaa. Käyttäjät vaativat palveluja, jotka palvelevat heidän tarpeitaan ja nopeuttavat tehtävien suorittamista. Tämän myötä muotoiluajattelun ja siihen liittyvien viitekehyksien soveltamisesta on muodostunut merkittävä kilpailuetu ohjelmistotuotteissa. Monet yritykset palkkaavat yhä enenevässä määrin käyttökokemusasiantuntijoita ja palvelumuotoilijoita, jotta voivat kehittää tuotteidensa käyttökokemusta. (Bestman 2019)

Tarkoitustaan palvelevien ja hyvän käyttökokemuksen tarjoavien ohjelmistotuotteiden nopea tuottaminen vaatii ketterien menetelmien muokkaamista muotoilulähtöisiksi. Lähtökohtaisesti ketterät menetelmät keskittyvät ongelman ratkaisuun ja ratkaisun toteuttamiseen, kun taas Design Sprint keskittyy ongelman määrittelyyn ja sen ratkaisemiseen. Design Sprintin johtavana ajatuksena on ideoiden testaaminen ilman niiden toteutusta ja käyttöönottoa. MVP (Minimum Viable Product) toteutuksen sijaan rakennetaan interaktiivinen prototyyppi. Prototyyppi testataan oikeilla käyttäjillä, jolloin saadaan paitsi palautetta ratkaisuidean toimivuudesta, myös arvokasta lisätietoa ratkaisun jatkokehittämiseen. (Bestman 2019)

Kuvio 16 havainnollistaa hypoteesista saatavan palautteen vaikutusta Lean Startup -syklin nopeuteen MVP mallissa ja Design Sprintissä.



Kuvio 16: Hypoteesin testaaminen Lean mallissa (Serafinelli 2016, piirretty uudelleen)

Erityisen paljon hyötyä Design sprintistä saadaan, kun sen avulla muodostetaan tuotteen kehitys-jono (eng. Backlog). Prototyypin avulla käyttäjätarinoita (eng. User Story) voidaan tarkas-tella käyttäjän näkökulmasta: miksi kyseistä toimintoa tarvitaan? kuka sitä käyttää? mitä lisä-arvoa toiminto tuottaa käyttäjälle? (Bestman 2019)

Käytännössä kehitysjonon muodostaminen Design Sprintin avulla toimii seuraavasti:

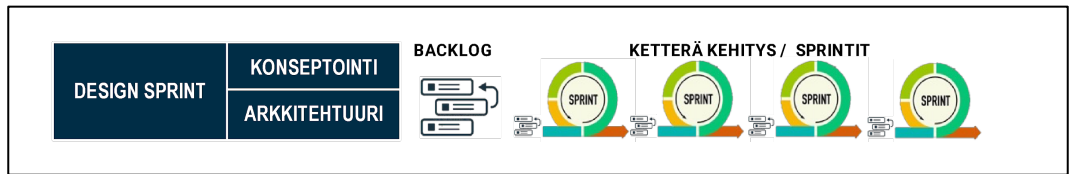
1. Asetetaan tavoite, kootaan tiimi ja suoritetaan Design Sprint.
2. Design Sprintin jälkeen luodaan käyttäjätarinat prototyypin pohjalta.
3. Tuotteen toteutusprojektia ohjataan Scrum-prosessin mukaisesti. Toteutusvaiheen ai-kana tiimi tukeutuu prototyyppiin käyttöliittymän ja käyttäjäinteraktioiden toteutuk- sessa. (Bestman 2019)

Muotoilumenetelmien ja ketterien menetelmien yhdistämisen edellytyksenä on, että muotoili-jat osallistuvat kehitystyöhön osana ketterää tiimiä (Bestman 2019).

Erityisen hyödyllistä Design Sprintin yhdistäminen ketteriin menetelmiin on Bestmanin (Best-man 2019) mukaan seuraavissa tapauksissa:

1. Uuden projektin alussa

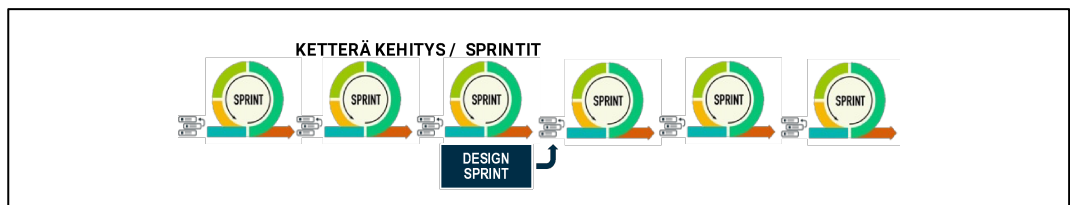
Design Sprintin avulla tuotetaan yhteinen ymmärrys suunniteltavan tuotteen merki-tyksestä asiakkaan liiketoiminnalle sekä käyttäjille. Tällöin Design Sprint voidaan tar-vittaessa iteroida vaativimpien osuuksien täsmentämiseksi. Kun yhteinen ymmärrys on luotu, voidaan määrittelytyö eriyttää toiminnalliseen määrittelyyn (konseptointi) ja tekniseen määrittelyyn (arkkitehtuuri). Tämän työskentelyn lopputuotoksena syntyy tuotteen kehitys-jono (backlog), jonka pohjalta ketterä toteutus voi alkaa. (Bestman 2019)



Kuvio 17: Design Sprint uuden projektin alussa (Bestman 2019, muokattu)

2. Tietyn osa-alueen merkittävässä muuttamisessa tai uuden lisäämisessä

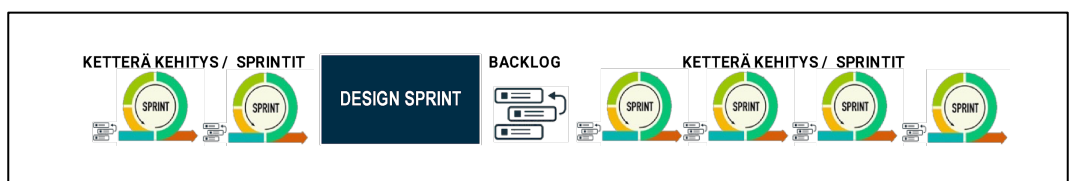
Tällöin toimitaan kuten kohdassa 1, mutta Design Sprintissä käsitellään vain uutta tai muutettavaa toiminnallisuutta. Tällöin on hyvä tarkastella myös valittua arkkitehtuuriratkaisu uusien toiminnallisten vaatimusten näkökulmasta. (Bestman 2019)



Kuvio 18: Design Sprint vaatimusten muuttuessa (Bestman 2019, muokattu)

3. Tuotteen vision, kehityspolun tai kehitysjonon selkiyttämässä tai silloin, kun tiimiin tulee useita uusia jäseniä

Tällöin normaali tuotekehityssykli keskeytetään Design Sprintin ajaksi. Design Sprintin tuotoksena on selkeytetty tuotteen kehitysjono ja kehityspolku. (Bestman 2019)



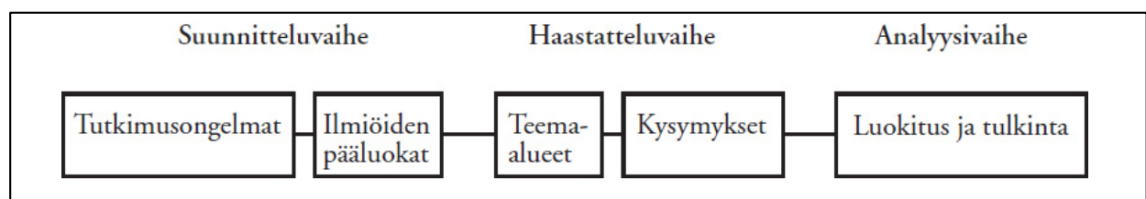
Kuvio 19: Design Sprint selkiyttämässä ja tiimin kokoonpanon muutoksessa (Bestman 2019, muokattu)

Kaiken kaikkiaan muotoilumenetelmien yhdistäminen ketteriin menetelmiin nähdään keinona muovata ajattelua perinteisestä suunnittelusta ja arvaamisesta kohti nopeita kokeiluja ja nopeaa oppimista. (Bestman 2019)

3.4 Määritä-vaihe: haasteen tarkentaminen

Toimeksiantajan kuvaamaa haastetta ja sen pohjalta muokattua tutkimuskysymystä ”**Kuinka voisimme tuoda muotoilutoiminnan kiinteäksi osaksi organisaation toimintamallia sekä ohjelmistokehityksen että myynnin näkökulmasta?**” tarkennettiin haastattelujen avulla. Syynä tähän menetelmävalintaan oli mm. se, että aihepiiri on kohdeorganisaatiossa vielä jäsentymätön, minkä vuoksi vastausten suuntaa on vaikea arvioida etukäteen. Lisäksi haastattelu mahdollistaa tiedonhankinnan suuntaamisen itse haastattelutilanteessa ja antaa mahdollisuuden vastausten taustalla olevien motiivien kartoittamiseen. (Hirsijärvi & Hurme 2001, 28) Haastattelut päätettiin toteuttaa teemahaastattelun rakenteella pienryhmissä. Teemahaastattelu on puolistrukturoitu haastattelu, eikä siinä ole tarkkaan määriteltyjä kysymyksiä. Haastattelun teemat on kuitenkin valittu etukäteen alkututkimuksen ja tutkimuskysymyksen pohjalta. (Hirsijärvi & Hurme 2001, 39)

Teema-alueet muodostettiin suunnitteluvaiheen pohjalta ja haastattelujen jälkeen suoritettiin analyysi. Kuvio 20 osoittaa teema-alueiden paikan tutkimuskokonaisuudessa.



Kuvio 20: Teema-alueet tutkimuskokonaisuudessa (Hirsijärvi & Hurme 2001, 56)

Ryhmähaastattelussa osanottajat keskustelevat ja ottavat kantaa melko spontaanisti, tekevät huomioita ja tuottavat monipuolista tietoa tutkittavasta ilmiöstä (Hirsijärvi & Hurme, 56). Näin haastattelun tuloksena syntyy varsin laaja tutkimusaineisto. Haastateltavat valittiin organisaatorajauksen mukaisesti siten, että he edustivat oman organisaationsa ajankohtaisinta osaamista. Ryhmät muodostettiin siten, että ryhmäläisten näkökulmissa oli selkeää synergiaa, mutta kuitenkin siten, ettei ryhmän sisällä ollut organisaatorakenteen luomia eturistiriitoja.

Haastatteluja varten tehtiin tukimateriaali (Liite 1: Haastattelun tukimateriaali), jonka rakenne oli seuraavanlainen:

1. Kehittämistehtävän esittely
2. Esittely alkututkimuksen löydösten pohjalta:
 - Asiakaslähtöisyyden merkityksen korostuminen
3. Keskustelu siitä miten edellä esitelty on mahdollisesti jo näkynyt
4. Pohdintaa asian merkityksestä
5. Esittely alkututkimuksen löydösten pohjalta:
 - Muotoilumenetelmien ja ketterän kehittämisen yhdistäminen

6. Pohdintaa siitä, miten uutta tietoa voisi organisaatiossa hyödyntää, sekä niihin liittyvistä odotuksista ja mahdollisista uhkista

Haastatteluja varten koottu tukimateriaali on opinnäytetyön liitteenä. Haastattelukysymykset rakentuivat avoimista kysymyksistä seuraavien aiheiden ympärille:

- Onko ilmiö näkynyt omassa toimintakentässäsi? Miten?
- Millaisia uhkia / haasteita ilmiö on luonut? Millaisia mahdollisuuksia se on luonut?
- Millaisia ajatuksia menetelmien yhdistäminen herättää?
- Millaisia haasteita tämä luo / voi luoda? Näetkö että siitä voisi olla hyötyä? Millaista?

3.4.1 Haastateltavana ydinprosessien ohjeistuksesta vastaava henkilö

Aluksi käsiteltiin ydinprosessien ohjeistuksen nykytilaa. Ohjeistus on laajentunut ajan myötä ja tätä nykyä se on kasvanut liian laajaksi. Käyttötapa perustuu projektin oma-aloitteisuuteen ja sen soveltamiseen löytyy napakka koulutus yrityksen kurssialustalta. Projektit eivät saa tukea alkuvaiheessa projektin käytäntöjen luomiseen, mutta käytänteet auditoidaan laatuorganisaation toimesta jälkeenpäin. Auditointilöydösten kautta pyritään toimintatapojen parantamiseen jatkossa.

Parhaiden käytänteiden osalta syntyy keskustelua ketterän sprintin ja Design Sprintin eroista. Periaatteessa yksi ketterän kehittämisen yleistymisen etu oli, että palaute toteutuksesta saadaan nopeammin kuin perinteistä vesiputousmallia noudattamalla. Design Sprintin avulla kuitenkin saadaan palautetta jo ennen toteuttamista, joka on huomattavan paljon nopeampaa ja kustannustehokkaampaa kuin palautteen kerääminen vasta MVP (Minimum Viable Product) -toteutuksen jälkeen. Keskustelun aikana huomataan, että ketterä kehittäminen on tuonut ohjelmistokehitykseen samoja ajatusmalleja (moniosajatiimi, yhteiskehittäminen, aikaisempi palaute) kuin mitä myös muotoiluajattelussa tunnustetaan. Muotoiluajattelu ja Design Sprintit ovatkin omiaan kiihdyttämään ketterän kehittämisen hyötyjä.

Selkeänä haasteena nähdään se, että muotoilutiimi tekee määrittelyä ilman arkkitehtia ja tämä johtaa usein sellaiseen tuotekonseptiin, jota ei voida kohtuullisella työmäärällä toteuttaa. Toinen vahvasti esiin tuleva haaste on se, että konseptointi ja käyttöliittymäsuunnittelu tehdään projektin alussa ja jää näin ollen irralliseksi. Muotoilun tulisi olla jatkuvaa tekemistä koko projektin ajan.

Parannusehdotuksina nousee esiin ydinprosessien ohjeistuksen päivittäminen siten, että muotoilu tulee kiinteäksi osaksi toimitusprosessia ja ketterään tiimiin määritellään uudet roolit (palvelumuotoilija, käyttöliittymäsuunnittelija). Päivityksen avulla voitaisiin nopeasti saada vaikuttavuutta, sillä päivitys mahdollistaisi löydökset auditointivaiheessa. Löydökset

puolestaan mahdollistavat mittaamisen, jolloin pystytään seuraamaan uuden toimintatavan omaksumista organisaatiossa.

3.4.2 Haastateltavana Ohjelmistopalvelut-yksikön vetäjä ja johtava arkkitehti

Asiakaslähtöisyyden lisääntyminen:

Asiakkaat vaativat parempaa käyttökokemusta ja sen myötä palvelumuotoilun merkitys on korostunut. Muotoilua on opittu - ja koko ajan opitaan - myymään tehokkaammin ja nopeammin. Muotoilu oli aiemmin yksittäinen osa määrittelyä, nyt ymmärretään, että muotoilun tulee olla jatkuvaa

Asiakaslähtöisyys on aina ollut keskeinen näkökulma uusien digitaalisten ratkaisujen tuottamisessa. Sen kysyntä on kasvanut parin-kolmen viime vuoden aikana. Sekä toimittajat että asiakkaat ymmärtävät muotoilun olevan välttämätöntä digitaalisten palveluiden tuottamisessa. Kasvua on näkynyt myös ostopersonien ja ostopolkujen kysynnässä. Lisäksi datan hyödyntäminen nousee esille entistä useammin.

Muotoilun uhat ja mahdollisuudet:

Palvelumuotoilu saattaa avata asiakkaan toiveiden tynnyrin ja alkuperäiseen sisältöön on vaarassa syntyä huomattava määrä lisää tekemistä. Asiakas haluaa yleensä teknisen ja toiminnallisen kuvauksen sekä työmääräarvion toteutettavasta ratkaisusta ja aina ei ymmärretä, että uusi tekeminen ei sisälly alkuperäiseen alustavaan kustannus- ja aikatauluarvioon. Tällöin MVP-sisällön määrittely ja tuotteen kehityspolun suunnittelu ovat tärkeässä roolissa.

Pidemmän aikajänteen näkyvyyden luominen tuotteen kehityspolun avulla ja nopea aloitus MVP:n avulla on johtanut siihen, että asiakkaat ovat alkaneet ostaa lopputuotteen sijaan tiimiä tietyksi ajaksi. Tämä on ehdottomasti positiivinen muutos, joka tuottaa pidempiä projekteja ja laadukkaampia tuotteita.

Käytännössä tämä on johtanut siihen, että tuotteen laatu on parantunut ja vastaa paremmin asiakkaan tarpeisiin. Ketterän tiimin allokaatiota on helpompi myydä kuin toteutusta, jonka lopputuotos on vielä epävarma. Yhteistyö asiakkaan kanssa on syventynyt: ollaan aidosti kumppaneita toimittaja-asiakassuhteen sijaan.

Ketterä kehittäminen ja muotoilu:

Ajatus ketterän kehittämisen ja muotoilun yhdistämisestä vaikuttaa mielenkiintoiselta ja hyödylliseltä. Siinä on varmasti paljon mahdollisuuksia. Malli on uusi, mutta käytäntö selkiytynee kokeilujen kautta.

On tärkeää oppia tunnistamaan, milloin ja missä projekteissa uudenlaista muotoiluohjautuvaa mallia voi käyttää ja sanoittaa ne materiaalissa auki.

Design sprintin hyödyntäminen:

Design Sprint silloin tällöin voisi tehdä hyvää: tuotteen tavoite pysyisi sen avulla kirkkaana ja sovitussa sisällössä pysyminen korostuu. Ketterän sprintin johtamisongelmaa ei voi korjata tällä mallilla vaan se vaatii aina juurianalyysin.

Toimintamallien ohjeistus (ydinprosessit):

Nykyinen ohjeistus ei kuvasta millään tavalla nykyisiä käytännön toimintatapoja. Nykytilaansa ohjeistus kuvaa pitkälti vesiputousmallia ja siitä puuttuu paljon nykyaikaisen ohjelmistokehityksen käytänteitä.

3.4.3 Haastateltavana määrittely- ja toteutusvaiheen toimijat

Haastateltavien roolit määrittely- ja toteutusvaiheessa ovat palvelumuotoilija, käyttöliittymäsuunnittelija, ohjelmistokehittäjä ja ratkaisuarkkitehti.

Asiakaslähtöisyyden lisääntyminen:

Asiakaskohtamisissa ja tarjouspyynnöissä tulee usein palautetta siitä, että ratkaisua on lähdetty luomaan liian teknologialähtöisesti ja tarjouksiin pyydetään tarkennuksia asiakaslähtöisyyteen liittyen. Yritys ei ole vielä osannut automaattisesti tarjota asiakaslähtöistä lähestymistapaa osana järjestelmätoimitusta.

Teknisellä puolella asiakaslähtöisyys ja sen esiin nostaminen on ollut myynnin kannalta tärkeää, mutta käytännössä tämä ei ole toteutunut juuri muulla tavoin, kuin että joskus on päästy tekemään käyttäjätestauksia. Toteutusprojekteissa ilmiö ei ole näkynyt, mutta muotoilutiimin puolella asiaa on tuotu vahvasti esiin.

Asiakkaat kaipaavat ja osaavat pyytää asiakaslähtöisyyttä. Ne asiakkaat, jotka eivät itse ole osanneet pyytää asiakaslähtöisyyttä, kiinnostuvat asiasta, kun se otetaan puheeksi. Tämän kokoiselta yritykseltä odotetaan ohjausta, inspiraatiota ja näkemyksellisyyttä.

Ketterä kehittäminen ja muotoilu:

Ajatus muotoilun tuomisesta osaksi ketterää kehittämistä ei ole vielä toistaiseksi näkynyt ohjelmistoprojekteissa, mutta tiimiläisillä on ollut vahva aavistus siitä, että tällaista toimintamallia tullaan ehdottamaan lähitulevaisuudessa. Isommissa ohjelmistohankkeissa saattaa joskus olla resursseja myös mm. muotoilutiimistä, mutta varsinaista syytä tähän ei tiedetä.

Muotoilun irrallisuus ketterästä kehittämisestä on näkynyt ohjelmistoratkaisun toimitusvaiheessa mm. asiakkaalta saatuna vakavana palautteena siitä, että toimitettu ratkaisu ei vastaa alkuperäiseen ongelmaan eikä siten myöskään vastaa asiakkaan odotuksia. Kaiken kaikkiaan kehitystiimi saa harvoin palautetta toimitetusta ratkaisusta, niinpä tiimi päätyy tyytymään oletukseen, että jos palautetta ei tule, niin kaikki on kunnossa - tai asiakas ei kehtaa huomauttaa puutteista. Valitettavan harvoin kehitystiimi saa palautetta onnistumisista. Lisäksi tiimi kaipaisi palautetta myös teknisestä toimivuudesta.

Jatkuvan määrittelyn ja muotoilun mallissa työmääräarvioiden tekeminen projektin alussa koetaan haasteelliseksi. Asiakkaalle alkumäärittelyn tärkeys on edelleen suuressa roolissa. Toisaalta koetaan, että perinteisessä projektin määrittelyvaiheessa on aina liian vähän tietoa eikä ohjelmiston toteutuksen aikana tapahtuvia toimintaympäristön muutoksia pystytty huomioimaan riittävän hyvin. Uudenlainen toimintamalli ja sen mukanaan tuomat muutokset tulisi huomioida jo myyntivaiheessa.

Design sprintin hyödyntäminen:

Joissain tarjouspyynnöissä on pyydetty viiden viikon mittaista määrittelyä. Kokemus tällaisen läpiviemisestä on ollut se, että nopean määrittelyn vetäjällä ei ollut riittävästi kokemusta ja/tai osaamista sellaisen läpivientiin.

Toimitusprojektin keskeyttäminen kehitysjonon selventämiseksi viikon mittaisen Design Sprintin ajaksi voi olla hyvinkin haastavaa projekteissa, joissa on tiukka aikaraja. Asiakas ei välttämättä näe keskeytyksen tuomaa hyötyä. Toimitusprojektin keskeytyksen juurisyy tulee selvittää, vaikka se ei olekaan ketterän kehittämisen hengen mukaista, sillä myynti tarvitsee perustelut viivästyksen ja lisätyön myymiseksi.

Ketterän tiimin kokoonpano ja muotoilu:

Käyttöliittymäsuunnittelijasta osana ketterää tiimiä koetaan olevan vahvaa potentiaalia. Käyttöliittymäsuunnittelijan tulisi olla tiimissä samalla panoksella kuin muukin tiimi, mukana Scrum-kokouksissa ja tuomassa koko ajan sisältöä frontend-kehitykseen ohjaten sen tekemistä. Lisäarvoa tuottaa etenkin yhteistyö ja ristiinkatselmointi ohjelmistokehittäjän kanssa. Tämän uskotaan tuovan tehokkuutta tiimin toimintaan.

Palvelumuotoilija on nykyisin mukana vain alussa, sen jälkeen muotoilun näkökulma häviää. Määrittelyn jälkeen projektit jäävät ohjelmistokehittäjien vastuulle, kunnes projekti päättyy kokonaan. Ilman palvelumuotoilijaa asiakkaan pyytämät muutokset jäävät arvioimatta ja soveltamatta tuotteen konseptiin - toimitusprojektin aikana tulevia muutoksia ei pitäisi käsitellä ohjelmistokehittäjien kesken. Olisi tärkeää, että palvelumuotoilija olisi sama henkilö määrittelystä toteutukseen ja hänelle olisi varattu riittävästi aikaa projektiin. Nykyisin ratkaisut

tehdään puutteellisella kokoonpanolla, mikä johtaa usein siihen, että asiakkaan odotukset valmiista tuotteesta eivät täyty. Palvelumuotoilijan ottaminen osaksi ketterää tiimiä voisi hyvinkin toimia.

Palvelumuotoilijan ja projektipäällikön roolien yhdistäminen koettiin vaativaksi, mutta hyödylliseksi. Projektipäällikön tärkein ominaisuus on suojella tiimiä asiakkaan vaatimuksilta, näin ollen palvelumuotoilijan rooli asiakasyhteyshenkilönä ja palomuurina asiakkaan ja toteutustiimin välillä olisi luonteva. Palvelumuotoiluja voisi toimia myös viestinvälittäjänä tiimiltä asiakkaan suuntaan, jolloin myös tuoteomistajan ymmärrys projektista kasvaa.

Uudessa toimintamallissa erityisesti huomioitavaa:

Asiakkaalla ei useinkaan ole riittävää tietoa ohjelmistokehityksestä, vaan pikemminkin asiakas odottaa toimittajalta ehdotusta toimintamalliksi. Toimittajalla odotetaan kyvykkyyttä tarjota asiakkaalle juuri kyseiseen projektiin sopivaa mallia, siksi myynnissä tulisi myydä kokonaisuus, ei erillisiä rooleja.

Tällä hetkellä fokus on vahvasti ohjelmiston kehitysmallissa, kokonaisuus jää puuttumaan. Asiakas jää usein ulkopuoliseksi ja passiiviseksi toimijaksi ja odottaa toimittajalta ehdotuksia ja yleisesti käytettyjä ratkaisumalleja.

Prosessia tulee vetää varmasti ja aktiivisesti eteenpäin ja huolehtia, että palavereissa käsitellään oikeita asioita. Asiakas ei vielä projektin alussa välttämättä osaa valmistautua palaveriin ja kestänee muutaman sprintin ajan, ennen kuin toimintamalli lähtee rullaamaan sujuvasti.

Oman työn mielekkyys:

Nykytilan toimintamallin siiloutumisen vaikutukset oman työn mielekkyyteen nousivat esiin haastattelun aikana. Määrittelyn painopisteen koettiin olevan vahvasti toiminnallisessa määrittelyssä ja arkkitehtuurikehitys oli toteutusta ajatellen liian suuripiirteinen. Joskus taas teknisestä ratkaisusta erillään määritelty ratkaisukonsepti saattoi olla mahdoton toteuttaa.

Määrittelytiimi puolestaan harmitteli sitä, että heille ei syntynyt kokemuseräistä työssäoppimistä, koska eivät olleet mukana enää toteutusvaiheessa ja näin ollen suunnitellun ratkaisun iteroiminen ei ollut mahdollista. Oma panos projektiin tuntui jäävän irralliseksi ja syntyi tunne siitä, että ammattitaitoa ei arvosteta, kun sitä ei enää uusien toimintojen tai muutosten suunnittelussa kaivata. Huolena oli myös se, että uusia suunnitelmia tehdään ilman alkuvaiheessa kerättyä syvää asiakasymmärrystä, jolloin suunniteltu ratkaisukonsepti voi pirstaloitua eikä lopulta enää vastaa tilaajan tarpeisiin.

3.4.4 Haastateltavana digitaalisten ratkaisujen myyjä

Asiakaslähtöisyyden lisääntyminen:

Asiakas ymmärtää, että asiakaslähtöisyys on juuri niin heikko kuin sen heikoin lenkki. Asiakaille ei kuitenkaan ole selvää, mitä sen toteuttaminen edellyttää. Pienemmissä yrityksissä tämän läpivieminen on helpompaa kuin suurissa yrityksissä. Suurten yritysten haasteena on digitiimien organisointi, joka on usein siloutunutta ja kukin osa miettii ratkaisuja vain omasta näkökulmastaan. Tällöin haasteeksi nousee asiakkaan kokonaiskokemus (asiakas 360) sekä palveluiden monikanavaisuus. Asiakas ei aina ymmärrä omaa kokoaan ja haluaa tehdä kaiken heti. Uusasiakashankinnassa palvelumuotoilu on tärkeässä roolissa. Asiakas haluaa olla asiakaskeskeinen, muttei tiedä miten toimia - tässä meidän tulee olla edistyksellinen toimija.

Ketterä kehittäminen ja muotoilu:

Muotoilun ostamiseen vaikuttaa asiakkaan digimaturiteetti: digimaturit organisaatiot ostavat helpommin muotoilulla vahvistettua ketterää tiimiä, kun taas ei-digimaturit organisaatiot eivät osaa tai uskalla ostaa laajennettua ketterää tiimiä. Tällöin myynti kannattaa tehdä pienissä paloissa: 1. Proof of Concept (PoC) kokeilu 2. MVP laajennetulla ketterällä tiimillä 3. jatkokehitys laajennetulla ketterällä tiimillä. Julkishallinto osaa yllättävän hyvin ostaa palvelumuotoilua osana kokonaistoimitusta.

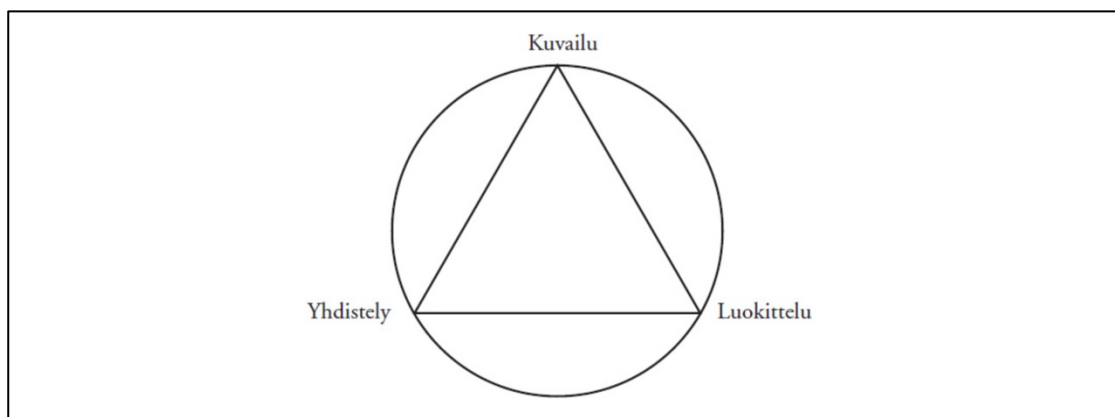
Asiakkaalle on hyvä kommunikoida selkeästi, että palvelumuotoilussa ratkottavat asiat ovat pois muusta määrittelytyöstä, se tuottaa parempaa laatua ja etenee ennustettavasti. Ketterässä mallissa korostuu asiakkaan ja toimittajan vastuu siitä, että loppuasiakas tarpeineen on projektin keskiössä. Usein palvelumuotoilua ostetaan siinä vaiheessa, kun oma hankekokoisuus on pirstaloitunut ja tarvitaan sen haltuunottoa. Tällöin pyydetään palvelumuotoilijaa selkiyttämään tilannetta. Kaikissa tilanteissa on tärkeää, että asiakaslähtöisyys ja valitut teknologiat tukevat toisiaan.

Datan hyödyntäminen:

Tällä hetkellä esiin nousee usein myös yrityksellä oleva data ja sen hyödyntäminen. Dataohjautuvuutta tulisikin hyödyntää Lean-mallissa jatkuvasti, ei vasta toiminnan jälkitarkastelussa.

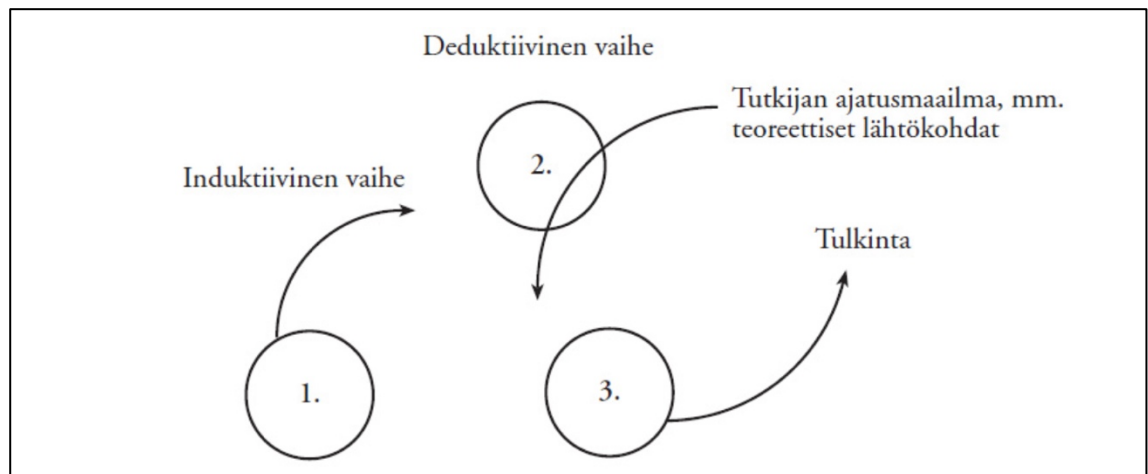
3.4.5 Koonti haastatteluista

Haastatteluista syntyi mittava määrä aineistoa. Osa aineistosta on huomioitu esitetystä ratkaisusta, sikäli kuin haastattelujen aikana syntyi uusia näkökulmia tai ideoita. Aineiston analyysi tehtiin siten, että kunkin haastattelun tuotokset kuvattiin muistiinpanojen pohjalta selkokieliseksi aineistoksi. Aineisto kokonaisuudessaan vietiin taulukkolaskentaohjelmaan, jossa sitä luokiteltiin erilaisten asiasanojen avulla. Luokittelun avulla voitiin eri näkökulmista syntyneet samansuuntaiset löydökset yhdistellä kokonaiskuvan luomiseksi. Yhdistely tarkoittaa sitä, että luokkien välille pyritään löytämään joitakin säännönmukaisuuksia, samankaltaisuuksia tai poikkeamia (kirja s 128). Kuvio 21 kuvaa kvalitatiivisen analyysin vaiheet.



Kuvio 21: Kvalitatiivinen analyysi kolmivaiheisena prosessina (Hirsijärvi & Hurme 2001, 122)

Kvalitatiivisen aineiston yhdistelyssä tehtiin Kuvio 22: Kvalitatiivisen aineiston yhdistely (Hirsijärvi & Hurme 2001, 128) mallin mukaisesti. Lähtöaineiston pohjalta kuvatut (kuvassa vaihe numero 1) havainnot luokiteltiin (kuvassa vaihe numero 2) asiasanojen avulla. Sen jälkeen tuotoksia tulkittiin koko siihen mennessä kerätyn aineiston ja teorian pohjalta (kuvassa vaihe numero 3). ((Hirsijärvi & Hurme 2001, 128-129)



Kuvio 22: Kvalitatiivisen aineiston yhdistely (Hirsijärvi & Hurme 2001, 128)

Näin syntyneet johtopäätökset esitellään tässä luvussa sanallisesti kuvattuina kokonaisvaltaisina johtopäätöksinä. Yhdistelyvaiheen tuotoksista esitetään kootusti nykytilan haasteet ja niihin liittyvät muutostoiveet sekä mahdolliset hyödyt liittyen uusiin toimintamalleihin. Eriksen on vielä listattu haastatteluissa esiin tulleita seikkoja, jotka on hyvä ottaa huomioon uutta mallia tarkemmin suunniteltaessa.

Tulkinta

Asiakkaat osaavat pääsääntöisesti vaatia käyttäjälähtöisiä ratkaisuja. Ne, jotka eivät itse osaa sitä pyytää, kiinnostuvat asiasta, kun se otetaan puheeksi. Haasteena tilanteessa on se, että asiakkaalla ei ole kokemusta siitä, millaisia asioita käyttäjälähtöisyyden toteutuminen vaatii. Niinpä toimittajalta odotetaan vahvaa ohjausta, inspiraatiota ja näkemysellisyyttä.

Ydinprosessien osalta todetaan, että ohjeistus on laajentunut ajan myötä ja tätä nykyä se on kasvanut liian laajaksi. Sen prosessikuvaus on ohjelmistokehityksen osalta pitkältä jo vanhentuneen vesiputousmallin mukainen. Siitä puuttuu suuri osa nykyaikaisen ohjelmistokehityksen käytänteistä eikä se siten kuvasta millään tavalla nykyisiä käytännön toimintatapoja. Sen päivittäminen nähdään kuitenkin tärkeänä ja mitattavana keinona tuoda uudenlaiset toimintatavat laajasti käyttöön tulevaisuudessa.

Palvelumuotoilun ymmärretään olevan olennainen osa digitaalisten palveluiden tuottamista, mutta sitä ostetaan usein vasta sitten, kun hankekokonaisuus on ajautunut umpikujaan ja ratkaisun alkuperäinen fokus on pirstaloitunut. Tämä johtunee siitä, että muotoilu on perinteisesti ollut yksittäinen vaihe projektin määrittelyssä, vaikka nykyisin aletaan nähdä, että muotoilun tulee olla jatkuvaa projektin kaikissa vaiheissa. Palvelumuotoilun avulla voidaan tuottaa parempaa laatua ja ennustettavuutta.

Haastattelujen ajankohtana toimeksiantajalla ei ole vakiintunutta tapaa tarjota ja toteuttaa käyttäjälähtöistä lähestymistapaa osana järjestelmätoimitusta. Muotoilutiimi on tuonut asiaa vahvasti esiin, mutta toteutusprojektien toimintatapoihin se ei ole vielä vakiintunut. Ajatusta ketterän kehittämisen ja muotoilun yhdistämisestä pidetään mielenkiintoisena ja hyödyllisenä. Siinä nähdään potentiaalia ratkaista useita tämän hetken haasteista. Mallia lähdetään mielellään kokeilemaan ja sen uskotaan selkiytyvän kokeilujen kautta.

Muotoiluohjautuvaa mallia on jo tämän opinnäytetyön kirjoittamisen aikana lähdetty tuomaan mukaan myyntivaiheeseen ja mallin mukaisia kokeiluja on tehty muutamia. Kokeiluista saatuja tuloksia käsitellään omassa luvussaan.

Kooste nykytilan haasteista

Taulukko 3 koostaa nykytilan haasteet ja niihin liittyvät muutostoiveet.

Nykytilan haaste	Muutostoive
Ratkaisuja tehdään liian teknologialähtöisesti.	
Fokus on ohjelmiston kehitysmallissa ja kokonaisuus jää puuttumaan.	
Asiakas jää usein ulkopuoliseksi ja passiiviseksi toimijaksi.	
Konseptointi ja käyttöliittymäsuunnittelu tehdään projektin alussa ja jää irralliseksi.	Muotoilun tulisi olla jatkuvaa koko projektin ajan.
Toteutusprojektin aikana tulevat muutokset jäävät ohjelmistokehittäjien vastuulle.	Muutoksia ei pitäisi käsitellä ohjelmistokehittäjien kesken.
Toteutettu ratkaisu ei vastaa alkuperäiseen ongelmaan eikä asiakkaan odotuksia.	Palvelumuotoilija osaksi ketterää tiimiä.
Muotoilutiimin luoma tuotekonsepti on sellainen, että sitä ei voida kohtuullisella työmäärällä toteuttaa.	Arkkitehti mukaan määrittelyyn.
Perinteisessä määrittelyvaiheessa toimintaympäristön muutoksia ei pystytä huomioimaan riittävästi.	

Nykytilan haaste	Muutostoive
Kehitystiimi saa harvoin palautetta toimituksesta ratkaisusta ja sen teknisestä toimivuudesta.	
Määrittelytiimille ei synny kokemuseräistä oppimista, koska eivät ole mukana toteutusvaiheessa.	Muotoilu mukaan myös toteutusvaiheeseen.
Määrittely koki, että oma panos projektiin jää irralliseksi.	Muotoilu mukaan myös toteutusvaiheeseen.
Uusia suunnitelmia tehdään ilman alkuvaiheessa kerättyä syvää asiakasymmärrystä, jolloin suunniteltu ratkaisukonsepti voi pirstaloitua eikä lopulta enää vastaa tilaajan tarpeisiin.	
Muotoilun myyminen irrallisena on haasteellista.	Asiakkaalle tulisi myydä kokonaisuus, ei erillisiä rooleja.
Lopullinen tuote on vaikea määritellä ja siten vaikea myydä.	Myynti tehdään pienissä paloissa: 1. PoC koikeilu 2. MVP laajennetulla ketterällä tiimillä 3. jatkokehitys laajennetulla ketterällä tiimillä.
Asiakas haluaa olla asiakaskeskeinen, muttei tiedä miten toimia.	Tämän kokoiselta yritykseltä odotetaan ohjausta, inspiraatiota ja näkemyksellisyyttä.
Toimittajalla odotetaan kyvykkyyttä tarjota asiakkaalle juuri kyseiseen projektiin sopiva mallia.	

Taulukko 3: Nykytilan haasteet

Koonti uusien toimintamallien mahdollisista hyödyistä

Haastattelujen yhteydessä käytiin läpi myös vertailu- ja trendianalyysin löydöksiä ja pohdittiin niiden soveltuvuutta toimeksiantajan haasteiden ratkaisemiseksi.

Taulukko 4 koostaa haastatteluissa esiin tulleet mahdolliset hyödyt liittyen uusiin toimintamalleihin.

Mahdolliset hyödyt
Ketterässä mallissa korostuu asiakkaan ja toimittajan vastuu siitä, että loppuasiakas tarpeineen on projektin keskiössä. Asiakas ei jää ulkopuoliseksi toimijaksi.
Palvelumuotoilijan avulla voidaan varmistaa, että asiakkaan odotukset tuotteesta täyttyvät.
Palvelumuotoilijan rooli asiakkaan tarpeiden/tavoitteiden ja teknisen tiimin välillä nähdään tärkeäksi, jotta molemminpuolinen ymmärrys projektista kasvaa.
Palvelumuotoilijan ja projektipäällikön roolien yhdistelmä on hyödyllinen, mutta vaativa.
Palvelumuotoilijan tulisi olla sama henkilö määrittelystä toteutukseen ja hänellä tulisi olla riittävästi aikaa projektiin.
Käyttöliittymäsuunnittelija olisi tärkeä osa ketterää tiimiä. Käyttöliittymäsuunnittelijan ja ohjelmistokehittäjän yhteistyö tuo tehokkuutta tiimin toimintaan.
Käyttöliittymäsuunnittelijan tulisi olla tiimissä samalla panoksella kuin muukin tiimi, mukana Scrum-kokouksissa ja tuomassa koko ajan sisältöä frontend-kehitykseen ohjaten sen tekemistä.
Laajennetulla tiimillä on mahdollista varmistaa, että asiakaslähtöisyys ja valitut teknologiat tukevat toisiaan.
Säännöllisen selkeyttävän Design Sprintin avulla auttaa pitämään tuotteen tavoitteen kirkaana ja sovituksissa sisällössä pysyminen helpottuu.

Taulukko 4: Uusien toimintamallien mahdolliset hyödyt

Koonti asioista, jotka tulee huomioida uutta mallia kehitettäessä

Haastattelujen yhteydessä nousi esiin myös asioita, jotka olisi hyvä huomioida uutta toimintamallia täsmennettäessä ja siihen siirryttäessä. Taulukko 5 koostaa nämä huomiot.

Uuden toimintamallin yhteydessä huomioitavaa
Jo myyntivaiheessa tulisi osata tunnistaa, milloin muotoiluohjautuvasta toimintamallista saadaan hyötyä.
Palvelumuotoilu saattaa avata asiakkaan toiveiden tynnyrin ja alkuperäiseen sisältöön on vaarassa syntyä huomattava määrä lisää tekemistä. MVP-sisällön määrittely ja tuotteen kehityspolun suunnittelu ovat tärkeässä roolissa.
Toimitusprojektin keskeyttäminen kehitysjonon selventämiseksi viikon mittaisen Design Sprintin ajaksi voi olla hyvinkin haastavaa projekteissa, joissa on tiukka aikaraja. Asiakas ei välttämättä näe keskeytyksen tuomaa hyötyä.
Prosessia tulee vetää varmasti ja aktiivisesti eteenpäin ja huolehtia, että palavereissa käsitellään oikeita asioita. Asiakas ei vielä projektin alussa välttämättä osaa valmistautua palaveriin ja kestänee muutaman sprintin ajan, ennen kuin toimintamalli lähtee rullaamaan sujuvasti.

Taulukko 5: Uuden toimintamallin yhteydessä huomioitavaa

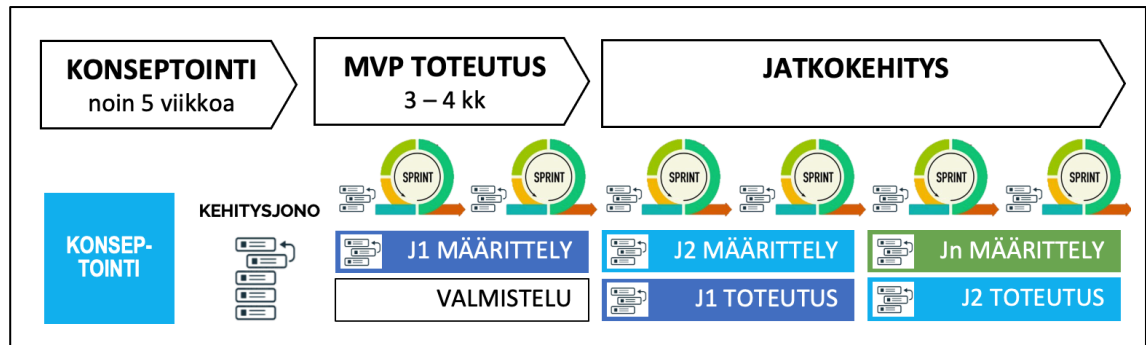
3.5 Kehitä-vaihe: Ratkaisun kehittäminen

Tämän kehittämistehtävän ratkaisuehdotuksissa pyritään ensisijaisesti ratkaisemaan alkukartoituksessa ja haastatteluissa esiin nousseet haasteet (Taulukko 3: Nykytilan haasteet). Ratkaisussa muotoilumenetelmät tuodaan luontevaksi osaksi ketterien menetelmien käytäntöjä. Mukaan tuodaan elementtejä tämän kehittämistehtävän vertailu- ja trendianalysista sekä ideoinnin yhteydessä esiin tulleista ratkaisumalleista ja parhaista käytänteistä. Ehdotettua toimintamallia kutsutaan tässä kehittämistehtävässä nimellä Muotoiluohjautuva ketterä projektimalli.

3.5.1 Muotoiluohjautuva ketterä projektimalli

Ylätasolla muotoiluohjautuva ketterä projektimalli ei juurikaan poikkea nykyisestä toimintamallista. Oleellinen ero syntyy siitä, että määrittelyvaihe on huomattavasti lyhyempi, sillä sen aikana kuvataan ratkaisu korkealla tasolla pohjautuen asiakkaan asettamiin tavoitteisiin. Tarkemman tason määrittely jatkuu koko projektin ajan. Ensimmäinen vaihe on tässä mallissa nimetty konseptoinniksi, sillä se on luonnos lopullisesta tuotteesta - ei lopullisen tuotteen tarkka kuvaus. Konseptointivaihe on ajallisesti huomattavasti lyhyempi kuin nykyisessä ketterässä mallissa ja sen aikainen työskentelytapa on intensiivinen sekä vahvasti yhteiskehittävä ja kokeileva. Tämän ansiosta myös MVP tuote voidaan yleensä julkaista nopeammin kuin perinteisessä mallissa. Toinen merkittävä ero perinteiseen ketterään toimintamalliin on se, että

määrittelytyö jatkuu toteutusvaiheen aikana kehitysjonon mukaisessa prioriteettijärjestyksessä. Näin vältetään hukkaan menevää määrittelytyötä Lean-periaatteen mukaisesti. Kun tarkempi määrittely tehdään hyvin lähellä toteutuksen alkamista, siinä voidaan huomioida toimintaympäristön muutokset.



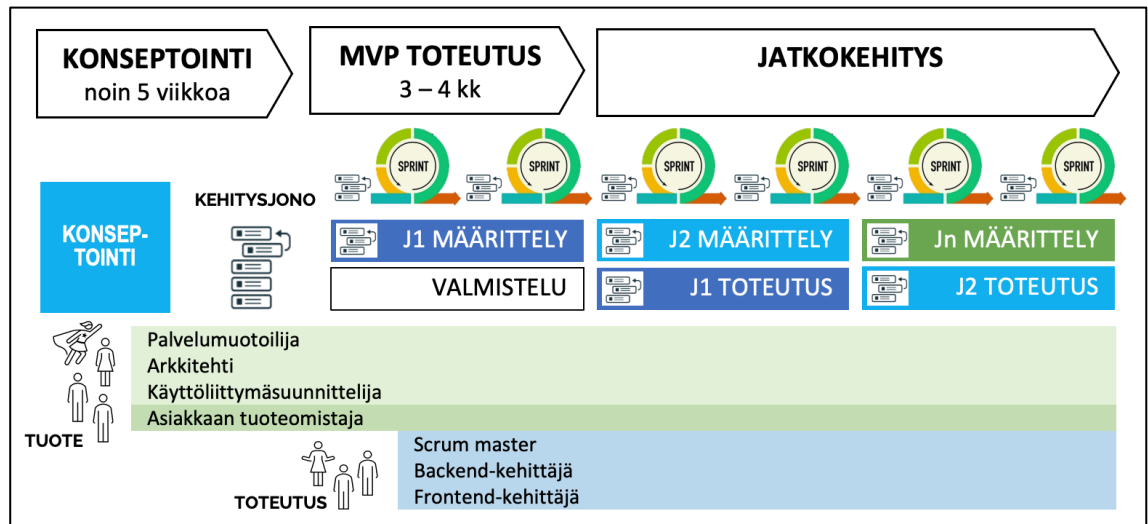
Kuvio 23: Muotoiluohjautuvan ketterän projektin vaiheet

Konseptointivaiheen avainroolit ovat palvelumuotoilija, käyttöliittymäsuunnittelija ja arkkitehti sekä tarvittavat asiakkaan edustajat tuoteomistajan johdolla. Tavoitteena on luoda yhteinen ymmärrys siten, että eri näkökulmien osaaminen tulee huomioitua ja palvelumuotoilun avulla voidaan hyödyntää kaikkien asiantuntijoiden kokemusta sekä luoda uudenlaisia ja aikaa kestäviä ratkaisukonsepteja.

Konseptointivaiheen resurssit ovat mukana myös toteutusvaiheen ketterässä tiimissä. Näin aloitusvaiheessa syntynyt syvä ymmärrys tilaajan tavoitteista ja käyttäjien tarpeista säilyy koko kehityssyklin ajan. Uudet ja muuttuneet toiminnallisuudet voidaan määrittellä luontevaksi osaksi kokonaisratkaisua muotoilun keinoin ja esitellä tilaajalle prototyypin avulla jo ennen toteutusta. Tämä auttaa varmistamaan sen, että toimitettava ratkaisu vastaa tilaajan tarpeita. Muotoiluprosessissa tilaaja on vahvasti mukana ja ymmärtää sitä kautta konkreettisemmin myös ratkaisua rajoittavat tekijät. Näin voidaan hallita tilaajan toimitettavalle järjestelmälle asettamia odotuksia.

Muotoiluohjautuvassa mallissa konseptointivaihe on lyhyt ja intensiivinen, jolloin asiakkaan helpompi ostaa se. Toteutusvaiheessa konseptointivaiheen resursseja ei enää tarvita 100 % kuormalla, mutta on tärkeää, että heille on varattu riittävästi aikaa toteutusvaiheen tekemiseen. Määrittely ja priorisointi on jatkuvaa ja niissä asiakkaalla on vahva rooli. Onkin luontevaa myydä toteutusvaiheen tiimiä tietyllä kokonaisalokaatiolla alustavan julkaisusuunnittelun osoittamaksi ajaksi. Kokonaisalokaatiolla tarkoitetaan tässä sitä, että voidaan myydä esim. 5 hengen tiimi: backend-kehittäjä, joka toimii myös arkkitehtinä, frontend-kehittäjä, Scrum Master, käyttöliittymäsuunnittelija ja palvelumuotoilija, siten että koko tiimi kuluttaa yhteensä koko projektin ajan 3 henkilötyöpäivää (FTE). Kutsumme tätä 3/5 FTE tiimiksi.

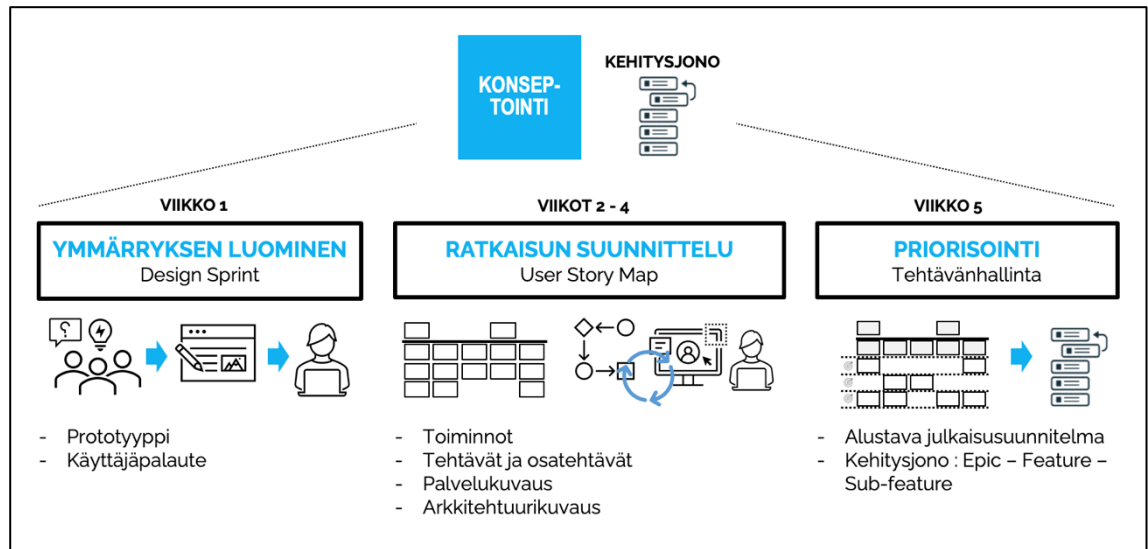
Asiakkaan on helppo ostaa kokonaisuutta, jonka keston ja sisältöön se voi aktiivisesti vaikuttaa ja jonka kuluja sen on helppo ennakoida. Muutoshallintaa tarvitaan tässä mallissa käytännössä vain silloin kun näyttää siltä, että sovittu resurssiallokaatio 3/5 ei riitä.



Kuvio 24: Muotoiluohjautuvan ketterän projektin vaiheet ja roolit

3.5.2 Konseptointivaihe muotoiluohjautuvassa projektissa

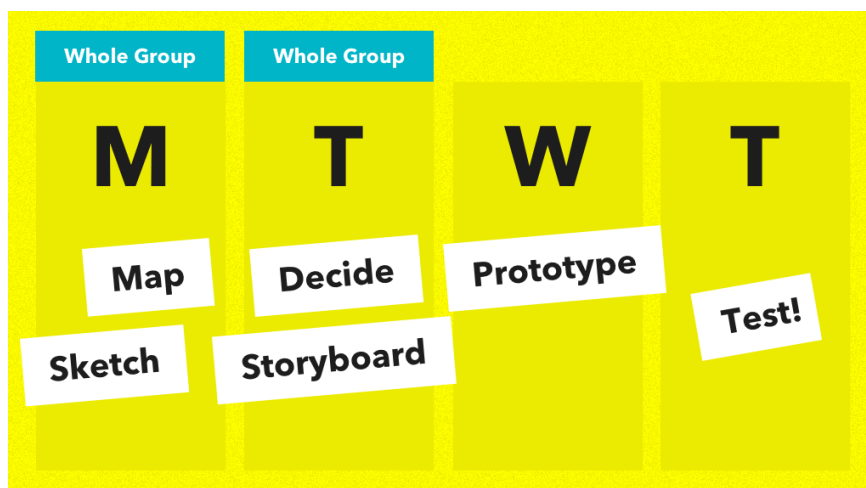
Konseptointivaiheen lopputuotoksena syntyy ratkaisun toiminnallinen ja tekninen suunnitelma sekä tuotteen kehitysjono ja alustava julkaisusuunnitelma tehtävähallintaan. Vaihe alkaa ymmärryksen luonnilla. Menetelmänä käytetään Design Sprintiä. Sen konkreettisenä lopputuotoksena on valitun osa-alueen prototyyppi, jonka avulla kerätään käyttäjäpalautetta jo heti projektin alussa. Tarvittaessa Design Sprint voidaan iteroida kokonaisuuden selventämiseksi tai jonkin tärkeän osa-alueen tarkentamiseksi. Design Sprintin pohjalta voidaan aloittaa ratkaisun suunnittelu. Siinä syntyneitä ymmärrystä syvennetään User Story Mapin avulla. Sen rinnalla toiminnallisen suunnittelun osalta luodaan palvelukuvaus ja teknisen suunnittelun osalta arkkitehtuurikuvaus. Tämän jälkeen User Story Map priorisoidaan, jolloin saadaan alustava julkaisusuunnitelma. Lopuksi User Story Mapin sisältö vietään tehtävähallintaan ketterän kehittämisen prosessin aloittamiseksi.



Kuvio 25: Muotoiluohjautuvan ketterän projektin konseptointivaihe

3.5.3 Ymmärryksen luominen Design Sprintin avulla

Ymmärryksen luomisessa käytetään Design Sprint 2.0 -menetelmää, jonka AJ&Smart on kehittänyt yhdessä Jake Knappin kanssa. Design Sprint 2.0 on neljän päivän mittainen. Kuvio 26 kuvaa kunkin päivän teemat. Kaksi ensimmäistä päivää ovat työpajapäiviä: maanantaina keskitytään haasteen ymmärtämiseen (eng. Map) ja ratkaisuihoiden luomiseen (eng. Sketch), tiistaina valitaan nyt kokeiltava ratkaisukonsepti (eng. Decide) ja määritellään käyttäjäpolku (Storyboard). Kolmas päivä on varattu prototyypin rakentamiseen ja neljäntenä päivänä suoritetaan käyttäjätestaus. (AJ&Smart 2023)



Kuvio 26: Design Sprint 2.0 päivien teemat (AJ&Smart 2023)

Menetelmässä on valittu valmiiksi työpajamenetelmät, joiden tuotokset luovat loogisen jatkumon koko kahden työpajapäivän prosessille. Valitut menetelmät ovat sellaisia, että niiden

tuotokset ovat konkreettisia. Tämä vähentää väärinymmärrysten määrää ja tukee toisten esittämien ideoiden jatkokehittämistä. Lisäksi niissä on huomioitu se, että osallistujille jää omaa miettimisaikaa ennen ryhmätyöskentelyä.

Monday - Day 1		Tuesday - Day 2	
Define the Challenge <ol style="list-style-type: none"> 1. Expert Interview 2. HMWs 3. Long Term Goal and Sprint Questions 4. Map 	Produce Solutions <ol style="list-style-type: none"> 1. Lightning Demos 2. 4-Part-Sketching 	Vote on Solutions <ol style="list-style-type: none"> 1. Heat Map Vote 2. Solution Presentation 3. Straw Poll Vote 4. Decider Vote 	The Storyboard <ol style="list-style-type: none"> 1. User Test Flow 2. Storyboarding

Kuvio 27: Design Sprint 2.0 työpajamenetelmät (AJ&Smart 2023)

Design Sprintin onnistumisen kannalta on tärkeää, että siihen valmistautuu huolellisesti. Pyri jo etukäteen selvittämään, millaisia haasteita työpajoissa tullaan käsittelemään. Huolehdi siitä, että työpajasarjalla on selkeä tavoite ja palaa työpajan tavoitteisiin säännöllisesti työpajan aikana. Valitse tiimi huolella - sen tulee edustaa riittävän montaa näkökulmaa ja paikalla tulee olla myös henkilö, joka voi tehdä ratkaisuun liittyviä päätöksiä. (AJ&Smart 2023) Muotoiluohjautuvassa mallissa toimittajan puolelta tulee olla paikalla palvelumuotoilija ja käyttöliittymäsuunnittelija, jotka yleensä myös huolehtivat työpajan fasilitoinnista, arkkitehti sekä asiakkaan tuoteomistaja. Muut asiakkaan puolelta kutsuttavat osallistujat tulee miettiä tapauskohtaisesti.

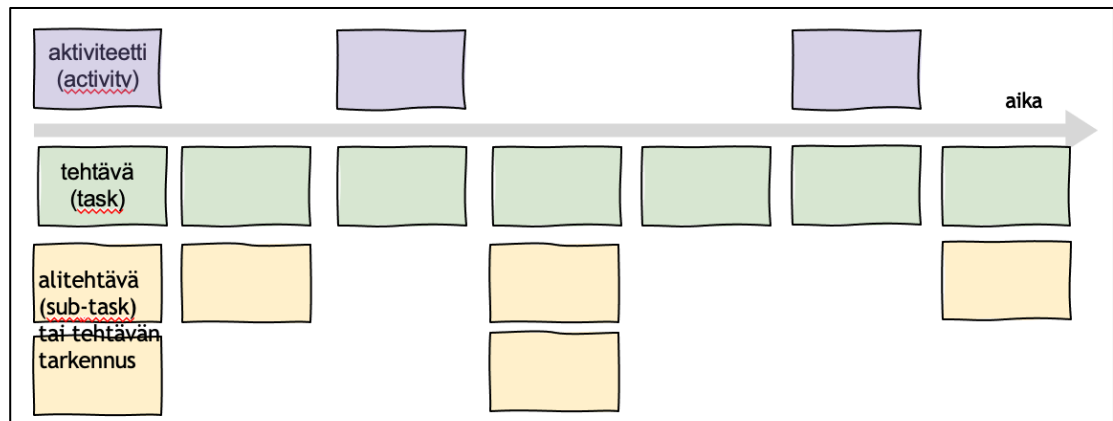
Design Sprintin lopputuotoksina syntyneet prototyyppi ja käyttäjäpalaute eivät yksinään riitä toteutusvaiheen aloittamiseksi (Sakho 2017). Toteutuksen aloittamista varten tarvitaan vielä sekä toiminnallinen että tekninen suunnittelu sekä karkea arvio työmäärästä ja julkaisuajankohdasta.

3.5.4 Ratkaisun suunnittelu ja User Story Mapping

User Story Mapping -menetelmä on käytännössä osoittautunut tehokkaaksi tavaksi luoda tuotteen kehitysjono sekä alustava kehityspolku Design Sprintissä syntyneen ymmärryksen ja prototyypin pohjalta (Sakho 2017). User Story Mapping on Jeff Pattonin kehittämä menetelmä moniulotteisen kehitysjonon luomiseksi. Se auttaa ymmärtämään tuotteen kokonaiskuvaa sen sijaan että fokus on vain yksittäisissä irrallisissa käyttäjätarinoissa. (Patton 2008)

User Story Map kuvaa hierarkkisesti tuotteen käyttäjän toiminnan. Ylin taso (eng. activity) kuvaa käyttäjän tavoitteet. Käytännössä se kuvaa tuotteen toiminnallisen sisällön ylätasolla

aktiviteetteina. Aktiviteetit (eng. activity) koostuvat yleensä joukosta tehtäviä (eng. task), jotka voidaan suorittaa vapaassa järjestyksessä. Aktiviteettiin liittyvät tehtävät kuvataan sen alle. Tehtävät ovat vielä liian suuria ollakseen ketterän kehittämisen käyttäjätarinoita (eng. User Story). Siksi ne jaetaan pienemmiksi alitehtäviksi ja kuvataan tehtävien alle. (Patton 2008)



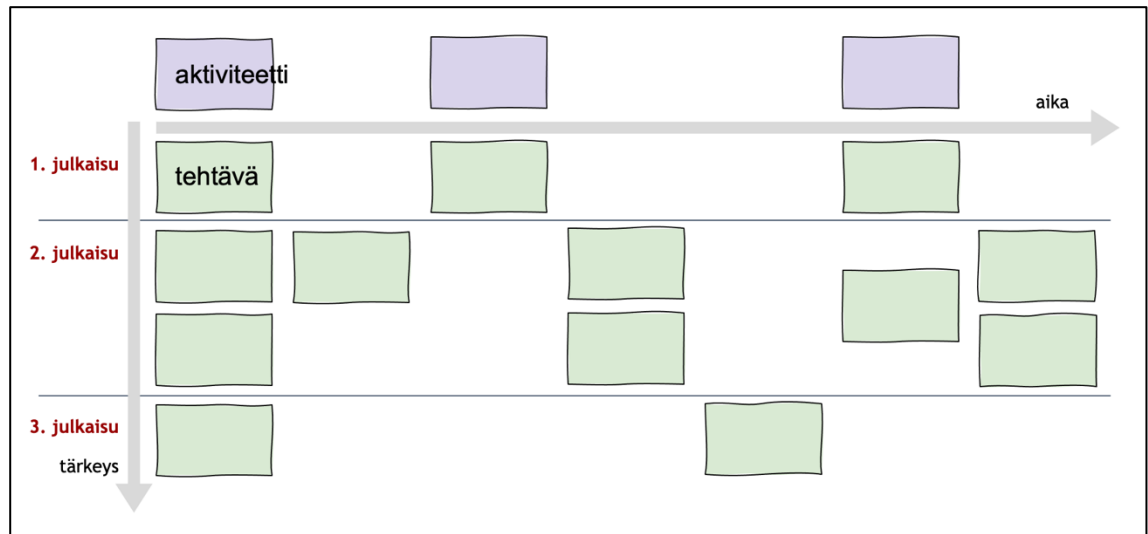
Kuvio 28: User Story Map -rakenne (Patton 2008, piirretty uudelleen)

Ensimmäinen versio User Story Mapista voidaan tuottaa Design Sprintin prototyypin pohjalta yksinkertaisesti siten, että prototyyppiä läpi käymällä kuvataan toiminnallisuus osaksi User Story Mapia. Tämän jälkeen kaikki konseptointitiimin jäsenet käyvät User Story Mapia läpi ja täydentävät sitä ymmärryksensä mukaan, kunnes voidaan todeta sen olevan riittävän valmis. (Sakho 2017).

Muotoiluohjautuvassa mallissa User Story Mapin tarkentamisen rinnalla tehdään käyttäjälähtöistä suunnittelua sekä teknistä suunnittelua. Niiden lopputuotoksena syntyy palvelukuvaus ja arkkitehtuurikuvaus. Nämä dokumentit yhdessä tuotteen kehitysjonon kanssa muodostavat alkutiedot toteutuksen aloittamiselle.

3.5.5 Priorisointi ja kehitysjonon luominen

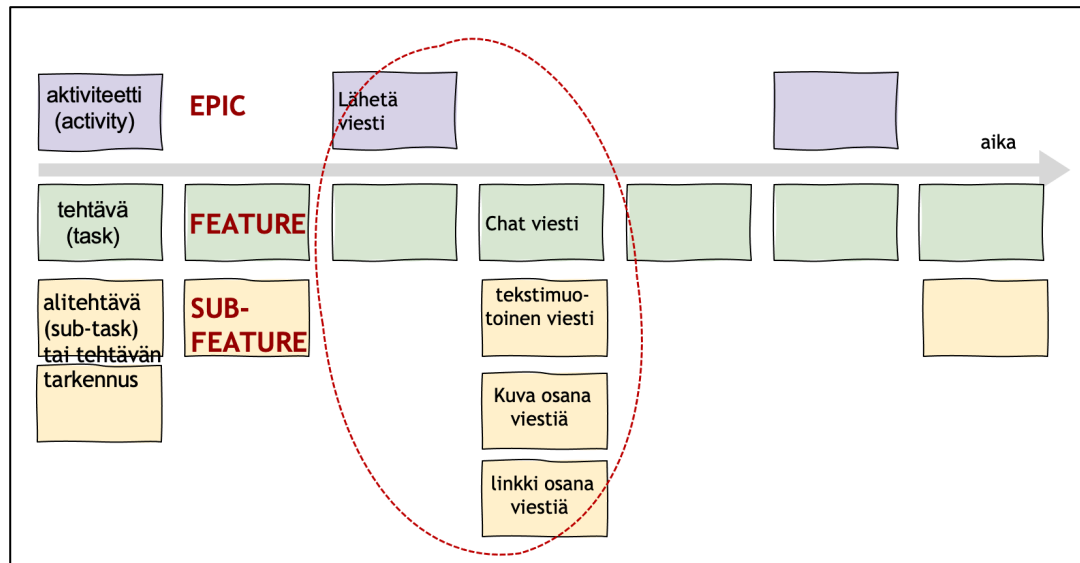
Tämän jälkeen User Story Mapin sisältö priorisoidaan ja priorisoinnin pohjalta luodaan alustava julkaisujärjestys (Patton 2015). Tehtävät priorisoidaan riveittäin siten, että tärkeimmät tehtävät ovat ylimpinä. Ylimmälle tehtäville muodostuu käytännössä pienin mahdollinen toiminnallinen kokonaisuus, jolla käyttäjä voi suorittaa tehtävän. Seuraaville riveille syntyy ehdotukset seuraavien julkaisujen sisällöstä. Kunkin julkaisun tehtäviä tulee tarkastella prioriteetin lisäksi myös loogisten ja teknisten riippuvuuksien näkökulmasta. (Patton 2008)



Kuvio 29: Julkaisun suunnittelu User Story Mapin avulla (Patton 2008, piirretty uudelleen)

Näin luotu julkaisusuunnitelma kuvaa liiketoiminnan näkökulmaa. Asiakas tekee päätöksiä ja seuraa projektin etenemistä kokonaisten aktiviteettien ja tehtävien tasolla. Toteutusprojektin suunnittelua ja ohjausta varten nämä vietään tehtävähallintajärjestelmään. Useimmat tehtävähallintajärjestelmät tukevat hierarkkista rakennetta, jossa elementit ovat abstraktiotasolta laajimmasta suppeimpaan kuvattuna: Epic, Feature, User Story ja Task. Sekä Feature-elementit että Task-elementit voivat olla sisäisesti hierarkkisia, eli hierarkiapuussa voi esiintyä Feature-elementin alapuolelle Sub-Feature -elementtejä ja Task-elementin alapuolella Sub-task -elementtejä.

User Story Mapin aktiviteetit vietään tehtävähallintajärjestelmän Epic-elementtiin ja tehtävät Feature-elementtiin. Feature-elementin avulla kuvataan yksittäistä julkaistavaa toimintoa ja Epic-elementti on puolestaan joukko Feature-elementtiin kuvattuja yksittäisiä julkaistavia toimintoja, jotka yhdessä mahdollistavat käyttäjän tavoitteen toteutumisen (AzureDevOps 2023). Usein Feature-elementteihin kuvattujen yksittäisten julkaistavien toimintojen tuottama lisäarvo voidaan lisätä progressiivisesti eri julkaisuissa. Tällöin ne kuvataan useampana osatoimintoina, jotka kuvataan Sub-Feature -elementtiin. Tällainen tilanne voisi olla esimerkiksi yksittäinen julkaistava toiminto (Feature): Chat, joka on jaettu osatoimintoihin (Sub-Feature): tekstimuotoinen viesti, kuva osana viestiä, linkki osana viestiä. Jo pelkän tekstimuotoisen chat-viestin lähettäminen tuottaa käyttäjälle lisäarvoa, ja kuvan tai linkin lisäämisen chat-viestiin lisäävät käyttävän chat-toiminnosta saamaa lisäarvoa.

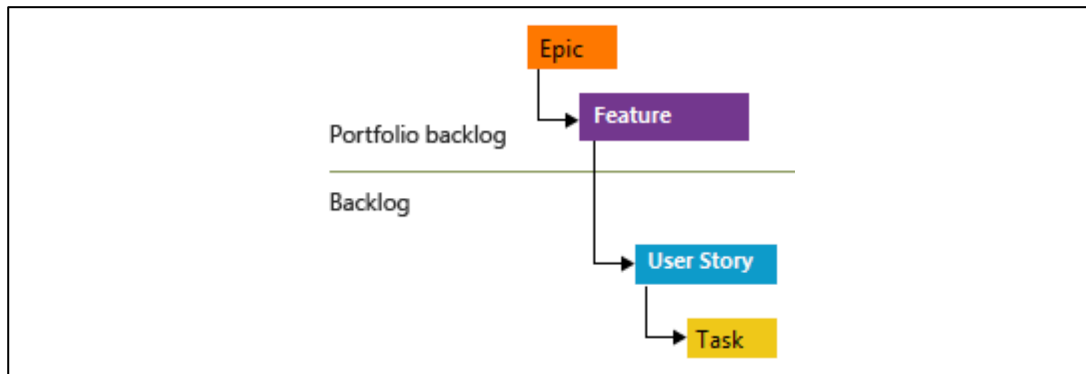


Kuvio 30: User Story Mapin elementit tehtävähallinnassa

Ketterä ohjelmistokehittäminen tapahtuu sprinteittäin, jolloin kukin tuotteen kehitysjonon tehtävän tulee olla toteutettavissa yhden sprintin aikana. Ohjelmistokehityksen näkökulmasta Feature-elementtiin kuvatut yksittäiset julkaistavat toiminnot on pilkottava pienemmiksi käyttäjätarinoiksi (User Story), joista kukin on laajuudeltaan sellainen, että se voidaan toteuttaa yhden sprintin aikana. Sprintin alussa tiimi ohjelmistokehittäjä kuvaa tehtävähallintajärjestelmään käyttäjätarinan toteuttamiseen liittyvät yksittäiset tehtävät Task-elementtiin. (AzureDevOps 2023).

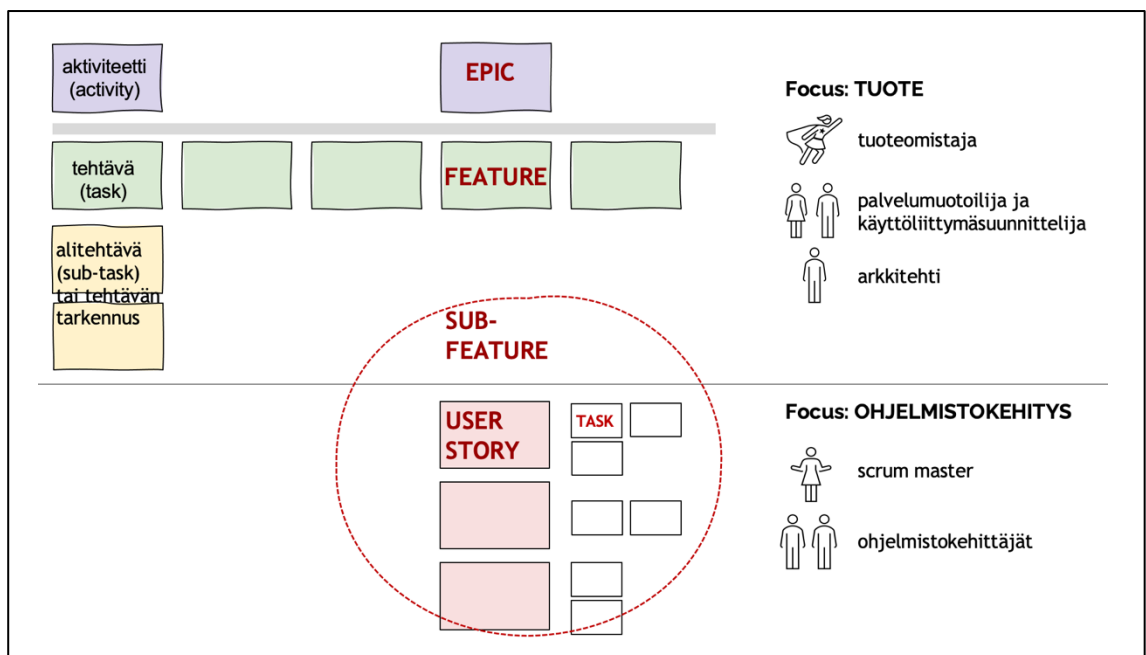
Epic- ja Feature-elementteihin kuvatut toiminnallisuudet esittävät ratkaisun liiketoiminnan näkökulmasta ja niiden fokus on julkaistavassa tuotteessa. Asiakkaan päätöksenteko ja projektin seuranta tapahtuu tällä tasolla. Tehtävähallintajärjestelmissä näitä varten voidaan luoda erillinen portfolion kehitysiono (Portfolio backlog). (AzureDevOps 2023) Tätä operoi käytännössä konseptointivaiheen ydintiimi eli tuoteomistaja, palvelumuotoilija, käyttöliittymäsuunnittelija ja arkkitehti.

User Story- ja Task-elementit ovat ketterän ohjelmistokehityksen elementtejä ja niiden fokus on ohjelmistokehityksessä. Toteutustyön suunnittelu ja seuranta tapahtuu näiden avulla. Tehtävähallintajärjestelmissä sitä varten voidaan luoda erillinen kehitysiono (Backlog). (AzureDevOps 2023) Käytännössä sitä operoi toteutusvaiheessa tiimiin liittyneet jäsenet eli Scrum Master ja ohjelmistokehittäjät.



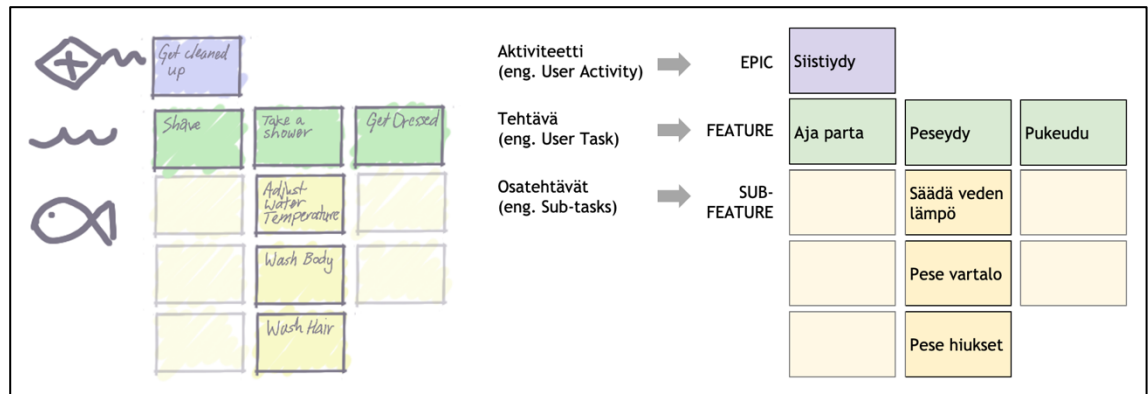
Kuvio 31: Tehtävähallintajärjestelmän elementit ja kehitysjonot (AzureDevOps 2023)

Kun nämä kaikki elementit kuvataan tehtävähallintaa hierarkkiset rakenteet säilyttäen, fokus pysyy tuotteen kokonaiskuvassa koko projektin ajan yksittäisten irrallisten käyttäjätarinoiden sijaan (Patton 2008).



Kuvio 32: Tehtävähallinnon elementit tehtävähallinnassa User Story Mapin pohjalta

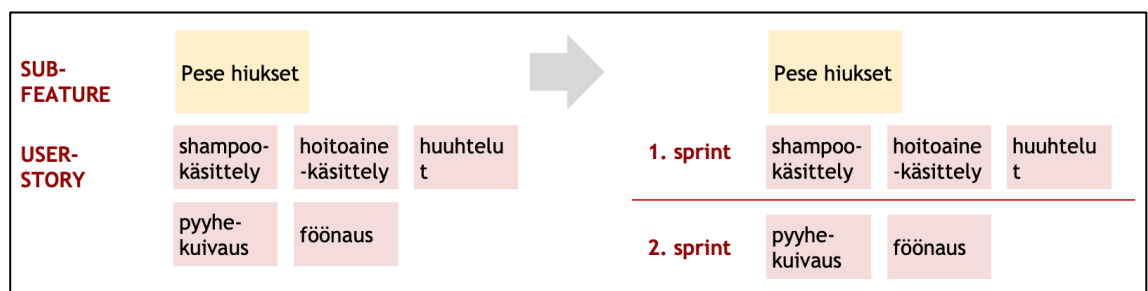
Käydään tehtävähallinnon elementit vielä läpi Pattonin (Patton 2015) esimerkin avulla. Esimerkki kuvaa arkista toimintaa: Siistiydy. Se kuvataan tehtävähallinnon Epic-elementtiin. Siistiytymiseen liittyy erilaisia tehtäviä kuten: Aja parta, Peseydy, Pukeudu. Nämä kuvataan tehtävähallinnon Feature-elementteihin. Peseydy toiminto voidaan jakaa vielä pienemmiksi osatoiminnoiksi ja ne kuvataan tehtävähallinnon Sub-feature -elementteihin: Säädä veden lämpö, Pese vartalo, Pese hiukset. Tämä on se taso, jota tuoteomistaja ja asiakkaan päättäjät operoivat.



Kuvio 33: Esimerkki tehtävähallinnan työjonon muodostamisesta: Epic - Feature (Patton 2015, osin)

Tulevassa julkaisussa on tavoitteena mahdollistaa osatehtävä Pesä hiukset, joka on kuvattu Sub-Feature -elementtiin. Tätä osatehtävää tulisi tarkastella sellaisessa Scrum-tapahtumassa, jossa valmistellaan julkaisun ja sprintin sisältöä. Vaikka tämä tapahtuma ei ole osa alkupe-
räistä tapahtumasarjaa, ovat monet tiimit todenneet sen hyödyllisiksi. Tapahtumasta käytetään usein nimitystä Preplanning (Scrumdesk 2023).

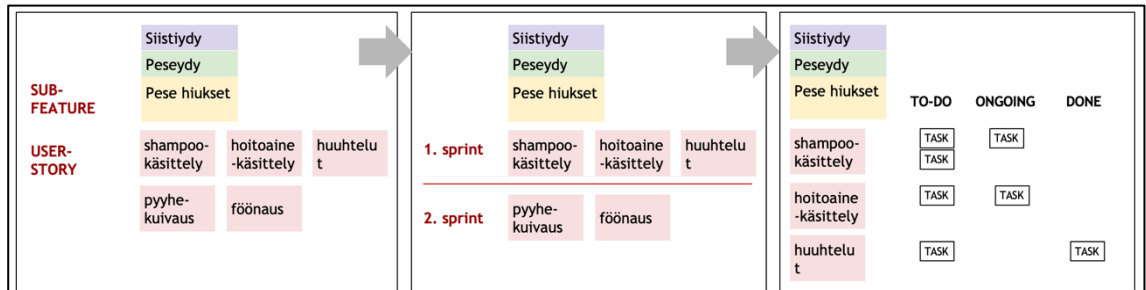
Pesä hiukset -osatehtävä on niin suuri, että sitä ei pysty toteuttamaan yhden sprintin aikana. Siksi se tulee pilkkoa sen kokoisiksi käyttäjätarinoiksi (User Story), että ne mahtuvat yhteen sprinttiin. Tässä esimerkissä julkaisu koostuu kahdesta sprintistä. Kun Pesä hiukset -tehtävää miettii toimien sarjana, sen voi jakaa seuraaviin osiin: shampoo käsittely, hoitoainekäsittely, huuhtelu, pyyhekuivaus, föönaus. Nämä kuvataan User Story -elementteihin ja ne päätetään jakaa julkaisun sisällä toteutettavaksi siten, että ensimmäisessä sprintissä toteutetaan shampoo- ja hoitoainekäsittely sekä huuhtelu ja toisessa sprintissä pyyhekuivaus ja föönaus. Kyseiset toimia kuvaavat käyttäjätarinat menevät osaksi asianomaisten sprinttien kehitysjojoja.



Kuvio 34: Esimerkki tehtävähallinnan työjonon muodostamisesta: Feature - User Story

Tehtävähallinnassa nämä kirjoitetaan User Story -elementtiin viralliseen muotoon: < käyttä-
järooli >:na haluan <tavoite> jotta < jokin syy>. Esimerkiksi huuhtelu: Hiustenpesijänä haluan
huuhdella käsittelyaineet pois hiuksista, jotta päänahka ei vaurioituisi.

Kun sprint käynnistyy, pilkotaan käyttäjätarinat pienemmiksi, ohjelmistokehitykseen liittyviksi yksittäisiksi tehtäviksi (eng. Task), joiden avulla ohjataan ohjelmistokehityksen päivittäistä toimintaa.



Kuvio 35: Esimerkki tehtävähallinnan työjonon muodostamisesta: User Story - Task

4 Johtopäätökset ja pohdintaa

Kehittämistehtävän tutkimuskysymys kuului: **”Kuinka voimme tuoda muotoilutoiminnan kiinteäksi osaksi organisaation toimintamallia sekä ohjelmistokehityksen että myynnin näkökulmasta?”** Tämä rajattiin koskemaan uusien digitaalisten palveluiden määrittelyä ja toteutusta. Lisäksi tutkittiin laajemmin, onko muotoilutoiminnan yhdistämiselle organisaation toimintamalliin tarvetta ja miksi tarve on syntynyt.

Lopputuotoksena toivottiin syntyvän **”Muotoiluohjautuva ketterä projektimalli”**, jossa muotoiluajattelun ja ketterän kehittämisen menetelmät luovat yhtenäisen prosessimallin asiakaslähteisten digitaalisten palveluiden tuottamiseksi. Tällä hetkellä sellaista vakiintunutta ohjelmistotuotannon kehitysmallia ei ole.

Luku Kehitä-vaihe: Ratkaisun kehittäminen kuvaa kehitysmallin ja resursoinnin vaiheittain.

Vertailu- ja trendianalyysin yhteydessä tuli ilmi asiakasymmärryksen muuttunut merkitys jopa kriittisenä liiketoiminnan elementtinä, kun digitaalisia ratkaisuja oli aiemmin tarkasteltu lähinnä kustannustehokkuuden näkökulmasta (McKinsey & Company 2020). Se siis osoittautui tärkeämmäksi tekijäksi kuin alun perin oletettiin. Markkinamuutoksen taustalla on vuoden 2020 Covid-19 Pandemian aiheuttama digitalisaation kiihtyminen (McKinsey & Company 2020). Tämän vuoksi toimeksiantajan kannattaa tuoda muotoilutoiminta kiinteäksi osaksi nykyisiä toimintamalleja.

4.1 Muotoiluohjautuva malli ja toimeksiantajan haasteet

Kehittämistehtävässä tuli luoda ehdotus sellaisesta toimintamallista, jossa muotoilun ja ketterän kehittämisen menetelmät yhdistyvät luontevaksi kokonaisuudeksi hyödyntäen niiden

vahvuuksia. Alkuhaastatteluisissa ja ymmärryksen syventämisen vaiheessa kartoitettiin nykyisen toimintamallin haasteita, joita uuden toimintamallin odotetaan ratkaisevan. Taulukko 6 kuvaa kootusti, miten nykytilan haasteet on otettu huomioon muotoiluohjautuvassa mallissa.

Nykytilan haaste	Miten haaste on huomioitu
Ratkaisuja tehdään liian teknologialähtöisesti.	Muotoilu- ja arkkitehtuurinäkökulma on mukana määrittelyssä koko projektin ajan
Fokus on ohjelmiston kehitysmallissa ja kokonaisuus jää puuttumaan.	User Story Map toimii siltana toiminnallisen määrittelyn ja toteutuksen välillä. Sen pohjalta ratkaisukokonaisuus kuvataan myös tehtävähallintaan.
Asiakas jää usein ulkopuoliseksi ja passiiviseksi toimijaksi.	Asiakas on mukana määrittelyssä ja priorisoinnissa koko projektin ajan, myös ymmärrys toteutustiimin näkökulmasta välittyy asiakkaalle.
Konseptointi ja käyttöliittymäsuunnittelu tehdään projektin alussa ja jää irralliseksi. Muutostoive: Muotoilun tulisi olla jatkuvaa koko projektin ajan.	Muotoilu on jatkuvaa toimintaa koko projektin ajan.
Toteutusprojektin aikana tulevat muutokset jäävät ohjelmistokehittäjien vastuulle. Muutostoive: Muutoksia ei pitäisi käsitellä ohjelmistokehittäjien kesken.	Konseptointivaiheen tekijät ovat mukana tiimissä koko projektin ajan.
Toteutettu ratkaisu ei vastaa alkuperäiseen ongelmaan eikä asiakkaan odotuksia. Muutostoive: Palvelumuotoilija osaksi ketterää tiimiä.	Palvelumuotoilija toimii koko projektin ajan läheisesti asiakkaan tuoteomistajan kanssa. Asiakas määrittelyn avainroolissa koko projektin ajan, myös ymmärrys toteutustiimin näkökulmasta välittyy asiakkaalle.
Muotoilutiimin luoma tuotekonsepti on sellainen, että sitä ei voida kohtuullisella työmäärällä toteuttaa. Muutostoive: Arkkitehti mukaan määrittelyyn.	Muotoilu- ja arkkitehtuurinäkökulma on mukana määrittelyssä koko projektin ajan.

Nykytilan haaste	Miten haaste on huomioitu
Perinteisessä määrittelyvaiheessa toimintaympäristön muutoksia ei pystytä huomioimaan riittävästi.	Määrittelyä ei tehdä valmiiksi konseptointivaiheessa vaan se on koko projektin ajan jatkuvaa toimintaa ja asiakas on siinä avainroolissa.
Kehitystiimi ei saa harvoin palautetta toimintatuesta ratkaisusta ja sen teknisestä toimitavuudesta.	Kehitystiimi ja asiakas toimivat koko projektin ajan läheisessä yhteistyössä, jolloin palautteen antaminen ja pyytäminen on luontevaa.
Määrittelytiimille ei synny kokemuseräistä oppimista, koska eivät ole mukana toteutusvaiheessa. Muutostoive: Muotoilu mukaan myös toteutusvaiheeseen.	Konseptointivaiheen tekijät ovat mukana tiimissä koko projektin ajan ja määrittelytyö on jatkuvaa toimintaa.
Määrittely koki, että oma panos projektiin jää irralliseksi. Muutostoive: Muotoilu mukaan myös toteutusvaiheeseen.	Konseptointivaiheen tekijät ovat mukana tiimissä koko projektin ajan ja määrittelytyö on jatkuvaa toimintaa.
Uusia suunnitelmia tehdään ilman alkuvaiheessa kerättyä syvää asiakasymmärrystä, jolloin suunniteltu ratkaisukonsepti voi pirstaloitua eikä lopulta enää vastaa tilaajan tarpeisiin..	Konseptointivaiheen tekijät ovat mukana tiimissä koko projektin ajan ja määrittelytyö on jatkuvaa toimintaa.
Muotoilun myyminen irrallisena on haasteellista. Muutostoive: Asiakkaalle tulisi myydä kokonaisuus, ei erillisiä rooleja.	Muotoilu myydään kiinteänä osana toteutustiimiä.
Lopullinen tuote on vaikea määrittellä ja siten vaikea myydä. Muutostoive: myynti tehdä pienissä paloissa: 1. PoC kokeilu 2. MVP laajennetulla ketterällä tiimillä 3. jatkokokeily laajennetulla ketterällä tiimillä.	Muotoiluohjautuvassa mallissa tuote kehittyy koko ajan ja asiakas on aktiivisessa roolissa sekä ratkaisun määrittelyssä että julkaisujen suunnittelussa. PoC ja MVP ovat tässäkin mallissa hyviä konkretiapisteitä.
Asiakas haluaa olla asiakaskeskeinen, muttetiedä miten toimia. Muutostoive: Tämän	Muotoiluohjautuva malli koostuu moderneista muotoilu- ja ohjelmistokehitysmenetelmistä.

Nykytilan haaste	Miten haaste on huomioitu
kokoiselta yritykseltä odotetaan ohjausta, inspiraatiota ja näkemyksellisyyttä.	
Toimittajalla odotetaan kyvykkyyttä tarjota asiakkaalle juuri kyseiseen projektiin sopivaa mallia.	Muotoiluohjautuvaa mallia voi - ja pitää - muokata kuhunkin projektiin sopivaksi.

Taulukko 6: Miten nykytilan haasteet on huomioitu muotoiluohjautuvassa mallissa

Muotoiluohjautuvassa mallissa on siis pyritty vastaamaan kaikkiin haastatteluissa esiin tulleisiin haasteisiin. Kun määrittelytyö tehdään moniammatillisen tiimin yhteistyönä, vältytään varmemmin tilanteelta, jossa suunniteltu järjestelmä on mahdoton toteuttaa. Sen lisäksi usein pystytään ratkaisukonseptiin tuomaan mukaan uusimman teknologian mahdollistamia ominaisuuksia. Näin pystymme osoittamaan tilaajalle näkemyksellisyyttä ja monipuolista syväosaamista toimittajana. Lisäksi kaikkien määrittelytyöhön osallistuvien henkilöiden kokemuspohjainen työssäoppiminen vahvistuu ja antaa mahdollisuuden tuoda seuraavaan projektiin uusia, innovatiivisia näkökulmia. Yhteiskehittämisen mallissa kaikkien näkökulmien panos on yhtä tärkeää. Jokainen kokee oman panoksensa mielekkääksi ja arvostetuksi projektissa.

4.2 Havainnot osittaisista kokeiluista

Muotoiluohjautuvaa mallia on kehittämistehtävän tekemisen aikana päästy joiltain osin pieni-muotoisesti kokeilemaan. Sitä on tuotu mukaan myyntivaiheeseen, mutta mallin mukaisia projektikokeiluja on tehty vasta muutamia, eikä muutos ole vielä laajassa mittakaavassa näkynyt kehitystiimille.

Myynnille muutos on näkynyt siten, että asiakkaat ovat alkaneet ostaa lopputuotteen sijaan tiimiä tietyksi ajaksi. Tämä on ehdottomasti positiivinen muutos, joka tuottaa pidempiä projekteja ja laadukkaampia tuotteita. Ketterän tiimin allokaatiota on ollut huomattavasti helpompaa myydä kuin toteutusta, jonka lopputuotos on vielä epävarma.

Projektikokeiluissa on jo MVP-vaiheessa huomattu, että yhteistyö asiakkaan kanssa on syventynyt: ollaan aidosti kumppaneita toimittaja-asiakassuhteen sijaan. Samalla tuotteen laatu on parantunut ja se vastaa paremmin asiakkaan tarpeisiin.

Aivan ongelmaton ei muotoiluohjautuva mallikaan vielä ole. Kokeiluissa on huomattu, että se vaatii sekä asiakkaan että tiimin kouluttamista jatkuvan määrittelyn malliin, sillä ketterän toimintamallin peruskäytännöt eivät tue jatkuvaa määrittelyä, vaan keskittyvät toteutuksen sujuvuuteen. Kouluttaminen onnistuu parhaiten yhteisöllisenä oppimisena eli mallin näyttämisen ja toimintamallin fasilitoinnin avulla.

Toteutustiimin huolenaiheena on ollut se, valmistuvatko toteutuksen tarvitsemat riittävän tarkat määrittelyt ajoissa ja sitoutuuko asiakas jatkuvaan määrittelyyn. Mallin jatkokehityksessä olisikin hyvä pohtia, voisiko ketterän kehittämisen mallia soveltaa myös tuotetason kontekstiin eli määrittelyyn ja sovittaa nämä kaksi kokonaisuutta toimimaan rinnakkain. On myös hyvä tutkia, löytyisikö tähän jo valmiiksi todettuja hyviä käytänteitä esim. SAFe (Scaled Agile Framework):sta tai DevOps (Development Operations):sta.

Lähteet

Painetut

Hirsjärvi, S. & Hurme H. 2001. Tutkimushaastattelu - Teemahaastattelun teoria ja käytäntö. Helsinki: Yliopistopaino.

Knapp, J., Zeratsky, J. and Kowitz, B. 2016. SPRINT - How to Solve Big Problems and Test New Ideas in Just Five Days. Bantam Press. Great Britain.

Polaine, A., Lovlie, L. and Reason, B. 2013. Service design - From insight to implementation. Brooklyn: Rosenfeld Media.

Tschimmel, K. 2012. Design Thinking as an Effective Toolkit for Innovation. Teoksessa: Proceedings of the XXIII ISPIIM Conference: Action for Innovation: Innovating from Experience. Barcelona.

Sähköiset

AJ&Smart. Design Sprint 101: What Is a Design Sprint, What Are They Used for, and How Can You Run Your Own? Workshopper. Viitattu 23.7.2023
<https://www.workshopper.com/post/design-sprint-101>

AWS. What Is SDLC (Software Development Lifecycle)? Amazon. Viitattu 3.8.2023.
<https://aws.amazon.com/what-is/sdlc/>

Azure DevOps. 2023. Define features and epics, organize your product and portfolio backlogs in Azure Boards. Microsoft. Viitattu 23.7.2023. <https://learn.microsoft.com/en-us/azure/devops/boards/backlogs/define-features-epics?view=azure-devops&tabs=agile-process>

Bestman, B. 2019. Design Sprints for Scrum Projects. Medium. Viitattu 15.9.2022. <https://medium.com/strive-studio/design-sprints-vs-scrum-ad4c15cf59c1>

Brown, Tim. 2008. Design Thinking. Harvard Business Review, June. Viitattu 29.6.2023.
http://www.ideo.com/images/uploads/thoughts/IDEO_HBR_Design_Thinking.pdf

Clatworthy, S. 2017. Service Design Thinking. ResearchGate. Viitattu 10.7.2023.
https://www.researchgate.net/publication/322733238_Service_design_thinking

Deloitte. 2020. The post-pandemic world is digital. Viitattu 15.9.2022. <https://www2.deloitte.com/nl/nl/pages/consumer/articles/the-post-covid-19-world-is-digital.html>

Design Council. 2023. The Double Diamond. Viitattu 20.9.2022. <https://www.designcouncil.org.uk/our-resources/the-double-diamond/>

Design Council. 2015. Design Methods Step 1: Discover. Viitattu 20.9.2022. <https://www.designcouncil.org.uk/our-resources/archive/articles/design-methods-step-1-discover/>

Design Council. 2015. Design Methods Step 2: Define. Viitattu 20.9.2022. <https://www.designcouncil.org.uk/our-resources/archive/articles/design-methods-step-2-define>

Design Council. 2015. Design Methods Step 3: Develop. Viitattu 20.9.2022. <https://www.designcouncil.org.uk/our-resources/archive/articles/design-methods-step-3-develop/>

Design Council. 2015. Design Methods Step 4: Deliver. Viitattu 20.9.2022. <https://www.designcouncil.org.uk/our-resources/archive/articles/design-methods-step-4-deliver/>

Distinguished. 2022. How does Software Development Process and Planning work? Viitattu 3.8.2023. <https://distinguished.io/blog/software-development-process-and-planning>

Innanen, P. 2018. Palvelumuotoiluprosessin vaiheet. Blogi. Viitattu 6.9.2022. <https://www.palvelumuotoilupalo.fi/blogi/palvelumuotoilun-prosessin-vaiheet/>

Interviewbit, 2022. Agile Model. Blogi. Viitattu 4.8.2023. <https://www.interviewbit.com/blog/agile-model/>

Interviewbit, 2022. What is Waterfall Model in Software Engineering? Blogi. Viitattu 4.8.2023. <https://www.interviewbit.com/blog/waterfall-model-in-software-engineering/>

Interviewbit, 2022. Iterative Model - Software Engineering. Blogi. Viitattu 4.8.2023. <https://www.interviewbit.com/blog/iterative-model/>

Knapp J. and Zeratsky, J. 2023. A brief history of the Design Sprint. The Sprint Book. Viitattu 21.7.2023. <https://www.thesprintbook.com/the-design-sprint>

McKinsey & Company. 2020. How COVID-19 has pushed companies over the technology tipping point—and transformed business forever. Viitattu 15.9.2022. <https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever>

Nakamura, P. 2021. The 4 Core Principles of The Design Sprint Process. Peter Nakamura. Viitattu 21.7.2023. <https://peternakamura.com/2021/02/03/the-4-core-principles-of-the-design-sprint-process/>

Patton, Jeff 2005. It's All in How You Slice. Better Software, January. Viitattu 17.7.2023. <https://jpattonassociates.com/the-new-backlog/>

Patton, Jeff. 2008. The New User Story Backlog is a Map. Jeff Patton & Associates. Viitattu 17.7.2023. https://jpattonassociates.com/wp-content/uploads/2015/01/how_you_slice_it.pdf

Patton, Jeff. 2015. Reference Guide for Story Mapping. Jeff Patton & Associates. Viitattu 21.7.2023. http://www.jpattonassociates.com/wp-content/uploads/2015/03/story_mapping.pdf

Sakho, A. 2017. Bridging Design Sprints and Development Sprints. Medium. Viitattu 21.7.2023. <https://sprintstories.com/bridging-design-sprints-and-development-sprints-4ab75dace97f>

Schwaber, K., Sutherland, J. 2020. The Scrum Guide. Viitattu 10.7.2023. <https://scrum-guides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

Scrum.org. 2023. What is Scrum. Viitattu 10.7.2023 <https://www.scrum.org/resources/what-scrum-module>

Scrumdesk. Sprint Preplanning. Scrumdesk. Viitattu 23.7.2023. <https://www.scrumdesk.com/start/manual-for-scrumdesk-start/sprint-preplanning/>

Serafinelli, S. 2016. Boosting the Lean Startup Cycle with Google Design Sprints. Medium. Viitattu 12.5.2023. <https://blog.prototypr.io/boosting-the-lean-startup-cycle-with-google-design-sprints-4398bbd6932c>

Stefanov, K. 2020. Agile Scrum in 5 minutes. Viitattu 10.7.2023. <https://www.linkedin.com/pulse/agile-scrum-5-minutes-kliment-stefanov/>

Vetan, J. 2020. Design Sprints vs Design Thinking vs Agile. Medium. Viitattu 5.10.2022. <https://medium.com/@icvetan/design-sprints-vs-design-thinking-vs-agile-49afea5bedfe>

Kuviot	
Kuvio 1: Tietoperustan lähtökohdat	10
Kuvio 2: Käyttökokemus syntyy useiden palvelukanavien kokonaisuudesta (Polaine, Lovlie & Reason 2013, 23, piirretty uudelleen).	12
Kuvio 3: Muotoilun prosessi ja sen kolme tilaa (Brown 2008, 6-7)	14
Kuvio 4: SDLC - ohjelmistokehityksen elinkaarimalli (Distinguished 2022, piirretty uudestaan)	16
Kuvio 5: SDLC vesiputousmalli (inverviewbit waterfall 2022, piirretty uudestaan)	17
Kuvio 6: SDLC iteratiivinen malli (interviewbit iterative 2022, piirretty uudestaan)	17
Kuvio 7: SDLC agile-malli (interviewbit agile 2022, piirretty uudestaan)	18
Kuvio 8: Scrumin peruspilarit ja arvot (Scrum.org 2023)	20
Kuvio 9: Scrum prosessin tapahtumat ja tuotokset	22
Kuvio 10: Nykyisin vallalla olevan ohjelmistotuotannon vaiheet	23
Kuvio 11: Nykyisin vallalla olevan ohjelmistotuotantoprosessin vaiheet ja roolit	23
Kuvio 12: Tuplatimantti-malli (Design Council 2023).....	24
Kuvio 13: Tuplatimantin soveltaminen tässä kehittämistehtävässä.....	26
Kuvio 14: Yrityksen organisaatio yleisellä tasolla.....	28
Kuvio 15: Kehittämistehtävän rajaus	29
Kuvio 16: Hypoteesin testaaminen Lean mallissa (Serafinelli 2016, piirretty uudelleen)	32
Kuvio 17: Design Sprint uuden projektin alussa (Bestman 2019, muokattu).....	33
Kuvio 18: Design Sprint vaatimusten muuttuessa (Bestman 2019, muokattu).....	33
Kuvio 19: Design Sprint selkiyttämisessä ja tiimin kokoonpanon muutoksessa (Bestman 2019, muokattu).....	33
Kuvio 20: Teema-alueet tutkimuskokonaisuudessa (Hirsijärvi & Hurme 2001, 56)	34
Kuvio 21: Kvalitatiivinen analyysi kolmivaiheisena prosessina (Hirsijärvi & Hurme 2001, 122)	41
Kuvio 22: Kvalitatiivisen aineiston yhdistely (Hirsijärvi & Hurme 2001, 128)	42
Kuvio 23: Muotoiluohjautuvan ketterän projektin vaiheet	47
Kuvio 24: Muotoiluohjautuvan ketterän projektin vaiheet ja roolit	48
Kuvio 25: Muotoiluohjautuvan ketterän projektin konseptointivaihe.....	49
Kuvio 26: Design Sprint 2.0 päivien teemat (AJ&Smart 2023)	49
Kuvio 27: Design Sprint 2.0 työpajamenetelmät (AJ&Smart 2023).....	50
Kuvio 28: User Story Map -rakenne (Patton 2008, piirretty uudelleen)	51
Kuvio 29: Julkaisun suunnittelu User Story Mapin avulla (Patton 2008, piirretty uudelleen) ..	52
Kuvio 30: User Story Mapin elementit tehtävähallinnassa	53
Kuvio 31: Tehtävähallintajärjestelmän elementit ja kehitysjonot (AzureDevOps 2023)	54
Kuvio 32: Tehtävähallinnan elementit tehtävähallinnassa User Story Mapin pohjalta	54
Kuvio 33: Esimerkki tehtävähallinnan työjonon muodostamisesta: Epic - Feature (Patton 2015, osin)	55
Kuvio 34: Esimerkki tehtävähallinnan työjonon muodostamisesta: Feature - User Story	55

Kuvio 35: Esimerkki tehtävähallinnan työjonon muodostamisesta: User Story - Task 56

Taulukot

Taulukko 1: Keskeisimmät ominaisuudet ja käyttötarkoitus - Muotoiluajattelu, Design Sprint ja Ketterä Sprint (Vetan 2020)	11
Taulukko 2: Ketterän kehittämisen (Scrum) hyödyt (Stefanov 2020)	19
Taulukko 3: Nykytilan haasteet	44
Taulukko 4: Uusien toimintamallien mahdolliset hyödyt	45
Taulukko 5: Uuden toimintamallin yhteydessä huomioitavaa.....	46
Taulukko 6: Miten nykytilan haasteet on huomioitu muotoiluohjautuvassa mallissa.....	59

Liitteet

Liite 1: Haastattelun tukimateriaali	67
--	----

Liite 1: Haastattelun tukimateriaali

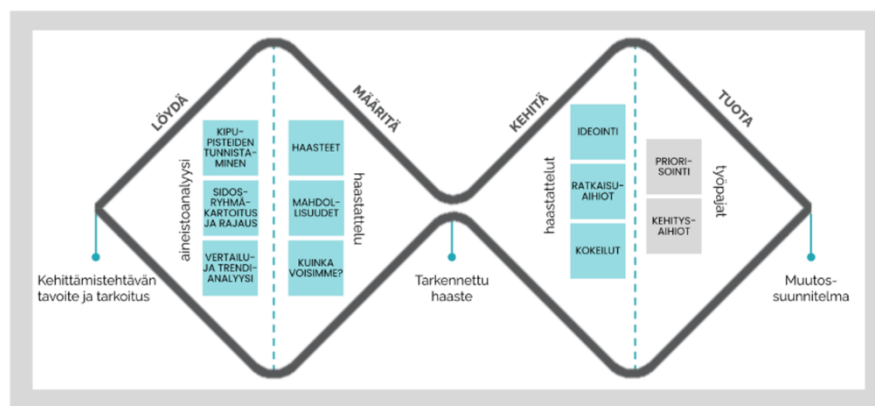
Käyttäjäkeskeisten ohjelmistotuotteiden kehittämistyön vauhdittaminen yhdistämällä muotoilumenetelmät ketteriin menetelmiin.

Muotoilumenetelmien yhdistäminen ketteriin menetelmiin.

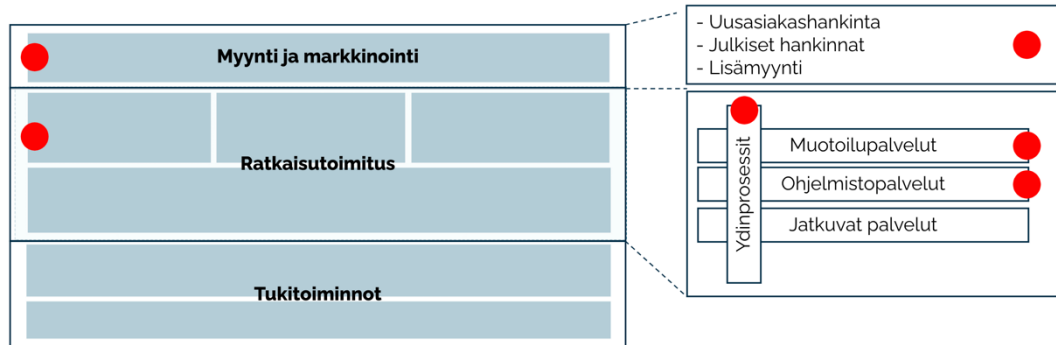
HAASTATTELUT

Tilaja: Design Studio

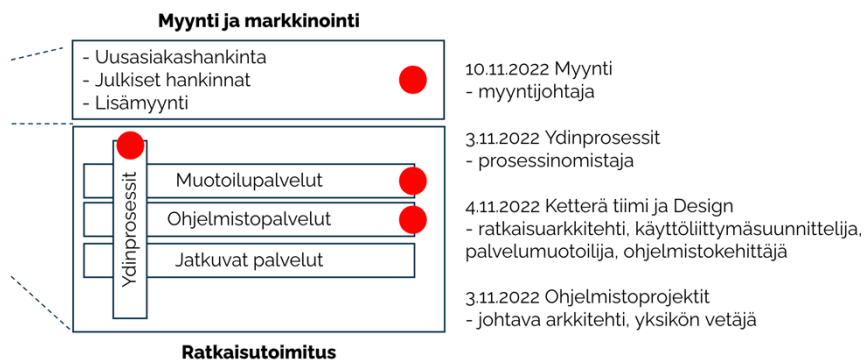
Kehittämissuunnitelma



Rajaus



Haastateltavat



Trendit

Kvanttihyppy digimaailmassa

- **Pandemian** pakotti kuluttajat ja yritykset siirtämään vuorovaikutuksen digitaalisiin kanaviin. Kun fyysiset kontaktit oli pakko pitää mahdollisimman vähäisinä, tuli **digitaalisuudesta** monien organisaatioiden **elinehto**.
- McKinseyn kyselytutkimuksen vastaajien mukaan, yritysten digitaalisten tuotteiden portfolio oli kutakuinkin sama ennen pandemiaa ja sen aikana. Tämä viittaa siihen, että **pandemian aikana yritykset ovat muokanneet lähinnä tarjontaansa sen sijaan, että olisivat muokanneet tuotteitaan.**

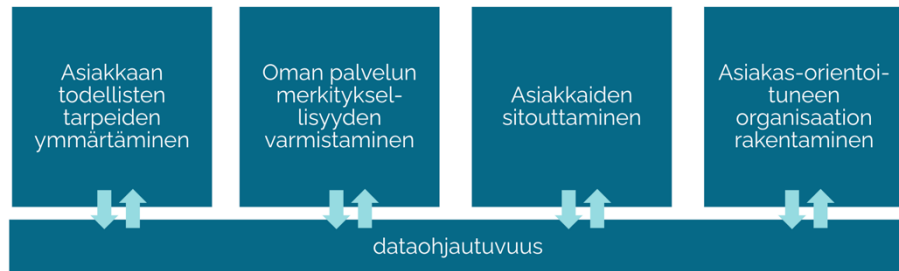
Pandemia osoitti, mitä digitaalisuus todella tarkoittaa:

- **Kyse ei ole** vain hienoista **sovelluksista, vaan ratkaisuketjusta**, joka ulottuu syvälle prosesseihin, ihmisiin ja teknologiaan.
- Kyse on **perustavanlaatuisesta muutoksesta** tavassamme työskennellä, elää ja harjoittaa liiketoimintaa.

Miten selviytyä kiihtyvässä digitalisaatiossa

Selvitäkseen organisaation on mukauduttava pandemian jälkeiseen maailmaan.

Selviytymisen peruspalikat:



[McKinsey Global Survey of executives](#)
[Deloitte Post-Covid World](#)

**Jos asiakaslähtöisyys ei ole
osa yrityksen DNA:ta,
niin mikään data tai teknologia
ei voi pelastaa sitä.**

[Deloitte Post-Covid World](#)

Kysymyksiä trendeistä

Onko ilmiö näkynyt omassa toimintakentässäsi? Miten?

Millaisia uhkia / haasteita ilmiö on luonut?

Millaisia mahdollisuuksia se on luonut?



POHDINTAA

Mikä avuksi?

Miten luoda kyky tuottaa asiakaslähtöisiä digitaalisia ratkaisuja nopeasti markkinoille?

- Ketteristä menetelmistä on muodostunut toimintamalli, joka auttaa tiimejä toimittamaan parempia ratkaisuja nopeammin
- Digitaalisen tarjonnan lisääntyessä asiakaslähtöisyydestä ja käyttökokemuksesta on muodostunut entistä tärkeämpi menestystekijä.
- Voisiko ketterien tiimien toimintamalliin yhdistää muotoilu- ja asiakaskeskeisyyttä esim. Design Sprintin avulla?
- Design Sprint on oikotie oppimiseen ja mahdollistaa idean testaamisen toteuttamatta sitä ensin.

	DESIGN THINKING	DESIGN SPRINT	AGILE SPRINTS
Problem definition	x		
Problem solving	x	x	
Solution definition		x	x
Solution delivery			x
Scope	Broad scope	Targeted problem	Defined by team
Problem size	Big picture	One problem	One increment
Feedback	Instant feedback (kvalitative)	Instant feedback (kvalitative)	Delayed feedback (kvantitative and kvalitive)
Iterative	x	x	x
Highly collaborative	x	x	
Cross-functional teams	x	x	
Strong Process	x	x	x
Needs facilitation	x	x	x

Medium: Design Sprints vs scrum

Design Sprint 2.0



Mahdollisia soveltamistapoja

CASE:
5 viikon määrittely



CASE:
puuroutunut tuotekehitys



CASE:
osa-alueen
selkeyttäminen



PI planning / Roadmapping Kvartaaleittain
Product Increment - ylätaso

NYKYTILA

Ohjelmistokehityksen resursointi ja design

Ketterä tiimi

Myydään tietty määrä resurssia (FTE)

Ydintiimi:

- Scrum Master
- Front koodari
- Back koodari / ohjelmistoarkkitehti
- Projektipäällikkö (hallinnollinen)

Vierailevat roolit:

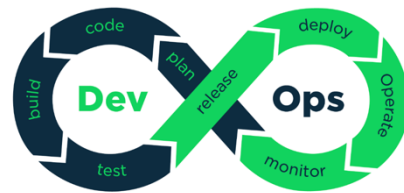
- Palvelumuotoilija
- UX
- asiantuntijat

Asiakas:

- Tuoteomistaja

Määrittelytiimi

- Palvelumuotoilija (+ PP)
- Käyttöliittymäsunnittelija
- Asiakas



EHDOTUS

Ohjelmistokehityksen resursointi ja design

Ketterä tiimi

Myydään tietty määrä resursseja (FTE)

Ydintiimi:

- Scrum Master + PP (hallinnollinen)
- Front koodari
- Käyttöliittymäsuunnittelija
- Back koodari / ohjelmistoarkkitehti
- Palvelumuotoilija "2. PO"

Määrittelytiimi

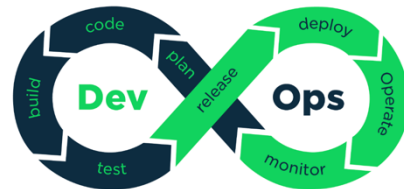
- Palvelumuotoilija (+ PP)
- Käyttöliittymäsuunnittelija
- Ohjelmistoarkkitehti
- Asiakas

Vierailevat roolit:

- asiantuntijat

Asiakas:

- Tuoteomistaja



Kysymyksiä toimintamallista

Millaisia ajatuksia menetelmien yhdistäminen herättää?

Millaisia haasteita tämä luo / voi luoda?

Näetkö että siitä voisi olla hyötyä? Millaista?

