

Roosa Onttonen

**YHTEISTYÖKEHITTÄMISEN MALLI
LOW-CODE -TEKNOLOGIALLE JA SEN
HYÖDYNTÄMINEN PALVELUN
TUOTTEISTAMISESSA**

Diplomityö
Johtamisen ja talouden tiedekunta
Tarkastaja: Miikka Palvalin
Tarkastaja: Jonna Käpylä
Tammikuu 2024

TIIVISTELMÄ

Roosa Onttonen: Yhteistyökehittämisen malli low-code -teknologialle ja sen hyödyntäminen palvelun tuotteistamisessa
Diplomityö
Tampereen yliopisto
Tietojohdamisen tutkinto-ohjelma
Tammikuu 2024

Digitalisaation merkitys on korostunut yhä enemmän viime vuosina yritysten liiketoiminnassa, jonka myötä yrityksille on syntynyt uudenlaisia tarpeita tietojärjestelmien näkökulmasta. On huomattu, että mobiilisovellukset ovat kasvattaneet suosiotaan paitsi kuluttajien keskuudessa myös yritysmaailmassa. Suuri digitaalisten ratkaisujen tarve on luonut haasteen ohjelmistokehittäjien pulan näkökulmasta, johon on pyritty löytämään ratkaisua. Low-code -teknologia on noussut viime vuosina yhdeksi vastaukseksi tähän haasteeseen. Low-code -teknologia mahdollistaa erilaisten digitaalisten ratkaisujen kehittämisen nopeammin ilman aiempaa osaamista ohjelmistokehityksestä. Kun low-code -ratkaisujen kysyntä ja tarve on kasvanut, tulee low-code -kehitystä samalla systematisoida, jotta samoja asioita ei tarvitse tehdä useaan kertaan sekä päällekkäisyyksiä voidaan välttää. Teknologiaa hyödyntävien osajien tulee siis ymmärtää, miten sitä käytetään parhaalla mahdollisella tavalla. Näiden perustelujen myötä tutkimus on rajattu tarkastelemaan low-code -kehitystä, yhteistyökehittämistä sekä low-code -palveluiden tuotteistamista. Tutkimuksen tavoitteena on luoda malli yhteistyökehittämiseen low-code -teknologialla, jota voidaan hyödyntää myös palvelun tuotteistamisessa. Mallin tavoitteena on helpottaa ja selkeyttää sisäistä liiketoimintaa niin low-code -kehittämisen kuin tuotteistamisen näkökulmista. Palvelun tuotteistamisen näkökulmassa pyritään ottamaan kantaa myös low-code -palveluiden ulkoiseen tuotteistamiseen.

Tutkimus toteutettiin kvalitatiivisena tapaustutkimuksena konstruktiivisella tutkimusotteella, jossa empiirinen osuus muodostui puolistrukturoidusta teemahaastattelusta sekä ratkaisun testaamisesta heikolla markkinatestillä. Tutkimus toteutettiin yhteistyössä case-organisaation kanssa, jonka rooli korostui erityisesti tutkimuksen empiirisessä osuudessa. Haastatteluiden pohjana hyödynnettiin low-code -teknologian, yhteistyökehittämisen, arvon yhteisluonnin sekä palveluiden tuotteistamisen kirjallisuutta. Haastattelurungon kysymyksiin tehtiin olettaus, että haastateltavalla on aiempaa kokemusta tai tietämystä low-code -teknologian ympäriltä. Empiirinen osuus koostui haastatteluista sekä heikosta markkinatestistä. Tutkimuksessa haastateltiin kuutta case-organisaation asiakasta sekä kahta case-organisaation työntekijää. Heikossa markkinatestissä tutkimuksessa luotua mallia esiteltiin yhdelle case-organisaation asiakkaalle sekä työntekijälle, jotka arvioivat mallin selkeyttä ja hyödynnettävyyttä.

Tutkimuksen tuloksena syntyi yhteistyökehittämisen malli low-code -teknologialle, joka toimii pohjana low-code -palveluiden sisäiselle tuotteistamiselle. Mallin avulla voidaan selkeyttää low-code -kehitystä yhteistyötiimin roolien sekä vastuiden osalta. Yhteistyökehittämisen mallilla pystytään myös esittämään eri vaihtoehtoja, miten yhteistyötiimin voi koota ja mitä vaiheita ja vastuita low-code -kehityksessä on tunnistettu. Malli vahvistaa ymmärrystä siitä, mitä osa-alueita low-code -palveluiden sisäinen tuotteistaminen sisältää. Tutkimuksessa otettiin kantaa myös low-code -palveluiden ulkoiseen tuotteistamiseen viestinnän, työn edistämisen sekä hinnoittelun näkökulmista. Tämä tutkimus vahvistaa low-code -teknologian, yhteistyökehittämisen ja palveluiden tuotteistamisen tutkimuskenttää lisäämällä aiempaan teoriaan uutta tietoa sekä muokkaamalla sitä.

Avainsanat: Low-code, yhteistyökehittäminen, palveluiden tuotteistaminen.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Roosa Onttonen: Co-development model for low-code technology and its utilization in service productization
Master's Thesis
Tampere University
Master's Degree Program in Information and Knowledge Management
January 2024

The importance of digitization has become more and more emphasized in the business operations of companies in recent years, which has created new needs for companies from the perspective of information systems. It has been noticed that mobile applications have grown in popularity not only among consumers but also in the business world. The great need for digital solutions has created a challenge in form of shortage of software developers, for which an effort has been made to find a solution. Low-code technology has emerged in recent years as one answer to this challenge. Low-code technology enables the development of various digital solutions faster without prior knowledge of software development. When the demand and need for low-code solutions has grown, simultaneously the development of low-code must be systematized, so that there is no need to do same things several times and duplications can be avoided. Experts who use technology must therefore understand how to use it in the best possible way. With these justifications, the research has been limited to examining low-code development, co-development and the productization of low-code services. The goal of the research is to create a model for co-development with low-code technology, which can also be used in the productization of the service. The goal of the model is to facilitate and clarify internal business from both low-code development and productization perspectives. From the point of view of service productization, the aim is also to take a position on the external productization of low-code services.

The research was carried out as a qualitative case study with a constructive research sample, where the empirical part consisted of a semi-structured thematic interview and testing the solution with a weak market test. The research was carried out in cooperation with the case organization, whose role was emphasized especially in the empirical part of the research. The interviews were based on the literature on low-code technology, co-development, value co-creation and productization of services. The questions of the interview body were made on the assumption that the interviewee has previous experience or knowledge around low-code technology. The empirical part consisted of interviews and a weak market test. Six customers of the case organization and two employees of the case organization were interviewed in the study. In a weak market test, the model created in the study was presented to one customer and an employee of the case organization, who evaluated the model's clarity and usability.

As a result of the research, a co-development model for low-code technology was created, which serves as a basis for the internal productization of low-code services. The model can be used to clarify low-code development regarding the roles and responsibilities of the cooperation team. With the co-development model, it is also possible to present different options, how to assemble a collaborative team and what phases and responsibilities have been identified in low-code development. The model strengthens the understanding of what areas the internal productization of low-code services includes. The study also took a position on the external productization of low-code services from the perspectives of communication, work promotion and pricing. This research strengthens the research field of low-code technology, co-development and productization of services by adding new information to previous theory and modifying it.

Keywords: Low-code, co-development, service productization.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Ajatus siitä, että kirjoitan nyt viimeisiä lauseita opintojeni viimeiseen työhön, tuntuu uskomattomalta, hienolta sekä haikealta. Opintojen aikana on selätetty monia haasteita, onnistuen sekä epäonnistuen. Diplomityö oli mielestäni hieno kokemus kaikkine onnistumisen sekä epäonnistumisen tunteiden kanssa ja oli eräänlainen lyhyt versio matkasta, jonka olen opintojeni aikana kulkenut. Haikeutta aiheuttaa aikakauden loppu, kun miettii opintoja. Toisaalta samalla tuntuu hienolta, että opinnot ovat vihdoin paketissa ja on uusien haasteiden aika sekä uuden aikakauden alku.

Ensinnäkin haluan kiittää työpaikkaani Navakka Group Oy:ta tästä upeasta mahdollisuudesta tehdä diplomityö työtehtävien ohella. Kiitos kaikki kollegat tuesta, tsempeistä ja oikoluvusta. Tuella oli iso rooli siinä, että diplomityön kirjoitusprosessi sujui hyvin ja työtä oli antoisaa tehdä. Erityiskiitos myös Navakan puolen ohjaajalleni Tommi Skogbergille arvokkaista huomioista, tuesta sekä avusta. Haluan kiittää myös muita sidosryhmiä, jotka ovat olleet mukana diplomityössäni eri tavoin. Erityisesti kaikki haastateltavat, suuret kiitokset ajastanne, mielenkiintoisista huomioista sekä näkemyksistä, joita haastatteluissa esititte.

Olen saanut erinomaisen tuen paitsi case-organisaatiolta myös yliopiston puolelta. Haluankin siis kiittää ohjaajaani Miikka Palvalinia tuesta ja avusta sekä erinomaisesta tavasta ohjata diplomityöntekijää. Haluan kiittää myös perhettäni, läheisiä sekä ystäviä avusta sekä tuesta, joka on ollut korvaamatonta. Koen, että kaikkien edellä mainittujen asioiden avulla sain tehtyä tästä diplomityöstä onnistuneen kokonaisuuden.

Tampereella, 5.1.2024

Roosa Onttonen

SISÄLLYSLUETTELO

1. JOHDANTO.....	1
1.1 Tutkimuksen taustaa.....	1
1.2 Tutkimuksen tavoitteet ja rajaukset.....	3
2. TUTKIMUKSMENETELMÄT JA TOTEUTUS.....	5
2.1 Lähestymistavat ja tutkimusfilosofia ja -strategia.....	5
2.2 Konstruktiivinen tutkimus.....	6
2.3 Kirjallisuus.....	9
2.4 Haastattelu tutkimusmenetelmänä sekä tutkimuksen otanta.....	11
2.5 Haastatteluiden toteutus ja haastatteluprosessi.....	13
2.6 Aineiston analysointi.....	14
3. LOW-CODE -TEKNOLOGIA JA YHTEISTYÖKEHITTÄMINEN.....	17
3.1 Low-code -teknologia.....	17
3.2 Yhteistyökehittäminen low-code -teknologialla.....	21
3.3 Case Microsoft: Fuusiokehittäminen ja Power Apps.....	22
3.4 Ketterän sovelluskehityksen piirteet.....	24
3.5 Arvon yhteisluonti.....	27
4. PALVELUIDEN TUOTTEISTAMINEN.....	31
4.1 Palveluajattelu.....	31
4.2 Palveluiden tuotteistaminen.....	33
4.3 Prosessimallit palveluiden tuotteistamisessa.....	36
4.4 Palveluiden paketointi.....	40
5. YHTEISTYÖKEHITTÄMINEN LOW-CODELLA JA SEN HYÖDYNTÄMINEN PALVELUN TUOTTEISTAMISESSA.....	44
5.1 Havaintoja kirjallisuudesta.....	44
5.2 Kirjallisuuden vastaukset tutkimuskysymyksiin.....	47
6. EMPIIRISEN TUTKIMUKSEN TULOKSET.....	50
6.1 Asiakashaastattelut.....	50
6.2 Palveluntarjoajan haastattelut.....	56
6.3 Yhteenveto haastatteluista.....	59
7. PÄÄTELMÄT.....	62
7.1 Yhteistyökehittämisen malli low-code -teknologialle.....	62
7.2 Yhteistyökehittämisen mallin hyödyntäminen low-code -palveluiden tuotteistamisessa.....	67
7.3 Yhteistyökehittämisen mallin validointi.....	73
8. YHTEENVETO.....	76

8.1	Yhteenveto ja keskeiset havainnot	76
8.2	Teoreettinen ja käytännön kontribuutio	77
8.3	Tutkimuksen arviointi	80
8.4	Jatkotutkimusmahdollisuudet.....	82
	LÄHTEET	84
	LIITE A: HAASTATTELURUNKO ASIAKASHAASTATTELUT	89
	LIITE B: HAASTATTELURUNKO CASE-ORGANISAATIO	91
	LIITE C: HEIKKO MARKKINATESTI	93

KUVALUETTELO

<i>Kuva 1: Konstruktiivisen tutkimuksen elementit (mukaillen lähteestä Kasanen et al. 1993).</i>	6
<i>Kuva 2: Konstruktiivisen tutkimuksen prosessi (mukaillen lähteestä Lukka 2006).</i>	7
<i>Kuva 3: Teemoittelun vaiheet (mukailtu lähteestä Braun & Clarke 2006).</i>	15
<i>Kuva 4: Merkittävät tapahtumat low-code -teknologian historiassa (mukaillen lähteestä Di Ruscio et al. 2022).</i>	18
<i>Kuva 5: Low-code -teknologian ominaisuudet. (Pinho et al. 2022; Rosa-Bilbao et al. 2023; Sanchis et al. 2019).</i>	19
<i>Kuva 6: Low-code -toimittajat (Vincent et al. 2022).</i>	20
<i>Kuva 7: Fuusiokehittämisen malli (mukailtu lähteestä Boyer et al. 2021).</i>	23
<i>Kuva 8: Scrum prosessimalli (mukailtu lähteestä Abrahamsso et al. 2017).</i>	25
<i>Kuva 9: Ketterä kehitys vs. low-code-kehitys (mukailtu lähteestä Al Alamin et al. 2021).</i>	27
<i>Kuva 10: Arvon yhteisluonti (mukailtu lähteestä Saarijärvi et al. 2013).</i>	28
<i>Kuva 11: Grönroos-Voima -malli arvon yhteisluonnille (mukailtu lähteestä Grönroos 2017).</i>	30
<i>Kuva 12: Palvelun laatumalli (Mukailtu lähteestä Polyakova & Mirza 2015).</i>	32
<i>Kuva 13: Palveluyrityksen asiakassuhteet (mukailtu lähteestä Grönroos 1982).</i>	33
<i>Kuva 14: Tuotteistamisen asteet (mukailtu lähteestä Sipilä 1995).</i>	34
<i>Kuva 15: Palvelun tuotteistamisprosessi (Shostack 1984).</i>	37
<i>Kuva 16: Palveluinnovaatioprosessi- vaihemalli (mukailtu lähteestä Trott 2008).</i>	38
<i>Kuva 17: Palvelukehitysmalli – iteratiivinen malli (mukailtu lähteestä Trott 2008).</i>	39
<i>Kuva 18: Asiakkaan saaman palvelun jaottelu (mukaillen Sipilä 2003).</i>	41
<i>Kuva 19: Fuusiokehittämisen malli (mukailtu lähteestä Boyer et al. 2021).</i>	45
<i>Kuva 20: Ketterä kehitys vs. low-code -kehitys (mukailtu lähteestä Al Alamin et al. 2021).</i>	45
<i>Kuva 21: Kirjallisuudesta tunnistetut tekijät ja yhteydet tutkittavien aiheiden ympäriltä.</i>	46
<i>Kuva 22: Haastatteluista tunnistetut alateemat low-coden ja arvonluonnin teemasta.</i>	59
<i>Kuva 23: Haastatteluista tunnistetut alateemat yhteistyökehittämisen ja low-code -teknologian teemasta.</i>	60
<i>Kuva 24: Haastatteluista tunnistetut alateemat palveluiden ja tuotteistamisen teemasta.</i>	61
<i>Kuva 25: Yhteistyötiimin roolit ja vastuut yhteistyökehittämisessä low-code -teknologialla.</i>	63
<i>Kuva 26: Yhteistyötiimin roolien vastuut low-code -kehityksen eri vaiheissa.</i>	66
<i>Kuva 27: Tiivistetty kuvaus yhteistyökehittämisen mallista sekä sen roolista sisäisessä tuotteistamisessa.</i>	68
<i>Kuva 28: Malli low-code -ratkaisun työmääräarvion luomiseen.</i>	70
<i>Kuva 29: Tutkimuksen päätelmät low-code -palveluiden tuotteistamisen näkökulmasta.</i>	72

TAULUKKOLUETTELO

<i>Taulukko 1: Hakutermit ja -lausekkeet aihealueittain.....</i>	<i>10</i>
<i>Taulukko 2: Haastattelun edut ja haitat (mukailtu lähteestä Hirsjärvi & Hurme 2022).....</i>	<i>11</i>
<i>Taulukko 3: Haastateltavat koodattuina, joihin viitataan empiirisessä osuudessa.....</i>	<i>13</i>
<i>Taulukko 4: Ketterän ohjelmistokehityksen manifesti (Misra et al. 2012).....</i>	<i>24</i>
<i>Taulukko 5: Palveluiden tuotteistamisen hyödyt (mukailtu lähteestä Harkonen 2021).....</i>	<i>35</i>
<i>Taulukko 6: Haastateltavat koodattuina.....</i>	<i>50</i>
<i>Taulukko 7: Yhteistyötiimin kokoonpanon eri mahdollisuudet.....</i>	<i>64</i>

LYHENTEET JA MERKINNÄT

Case-organisaatio Navakka Group Oy
LCDP Low-code development platform

1. JOHDANTO

Tässä luvussa esitellään tutkimuksen taustaa sekä tavoitteita. Tämän lisäksi luvussa perehdytään määritettyihin tutkimuskysymyksiin.

1.1 Tutkimuksen taustaa

Datasta sekä analytiikasta on tullut nykypäivänä yritysten liiketoimintastrategioiden päätekijä. Erityisesti tätä muutosta on kiihdyttänyt covid-19 pandemia, jonka aikana työnteko siirtyi digitalisaation varaan usealla eri liiketoiminnan osa-alueella. Digitalisaatio ja sen johtaminen vaatii uusia näkemyksiä ja kykyä tehdä päätöksiä. Tähän liittyy esimerkiksi kysymys siitä, miten organisaation johtamista tehdään tulevaisuudessa. (Duncan 2021) Low-code -kehitysalustat tarjoavat teknologiamekanismeja, jotka helpottavat ja automatisoivat ohjelmistosovellusten kehittämistä tukemaan nykyisiä yritysten tarpeita ja edistämään digitaalista muutosta (Sanchis et al. 2019).

Sanchis et al. (2019) mukaan termi low-code on keksitty vuonna 2014 Forresterin tutkimuksessa. Low-code -kehitysalustat ovat ekosysteemejä, joiden avulla voidaan kehittää erilaisia sovelluksia sekä automaatioita. Low-code -kehitysalustat ovat esikonfiguroitu, jolloin varsinaista koodausta tarvitaan suhteellisen vähän, kun verrataan perinteiseen ohjelmistokehitykseen. Pääideana low-code -kehitysalustoilla on tuoda sovelluskehitys lähemmäs liiketoiminnan eri osa-alueiden työntekijöitä, myös ohjelmistokehittäjien ulkopuolelle. Tämä lisää yritysten ketteryyttä ja digitalisaatiota sekä tuo edullisemmän ja nopeamman vaihtoehdon uuden sovellusratkaisun kehittämiseksi. (Sanchis et al. 2019)

Duncan (2021) arvioi, että low-code -teknologialla on merkittävä rooli ERP-liiketoiminnassa. Artikkelissa arvioidaan, että vuoteen 2025 mennessä yli 65 % käyttöönoton jälkeisistä ERP-muutoksista tehdään kansalaiskehittäjien toimesta, jotka hyödyntävät low-code -sovellusalustoja. Artikkelissa painotetaan erityisesti sitä, että ERP-strategiaa suunniteltaessa tulee ottaa huomioon jatkuvan liiketoiminnan arvon luominen. (Duncan 2021)

Sanchis et al. (2019) esittelevät tutkimusta, jossa analysoitiin yli 3300 IT-ammattilaisen vastauksia. Vastauksista havaittiin selkeitä syitä, miksi low-code -sovelluskehitystä kannattaisi hyödyntää. Esimerkkejä ovat digitalisaation kiihtyminen yrityksessä,

yrittäjien reagoitavuuden lisääminen, vähentää riippuvuutta palkata vaikeasti tavoitettavissa olevaa teknistä taitoa sekä mahdollistaa kansalaiskehittäjien edistää yrityksen liiketoiminnan prosesseja. Vastaajista suurin osa oli yhtä mieltä siitä, että low-code -teknologialla on merkittävä yhteys digitalisaation kiihtymiseen yrityksessä (66 %). (Sanchis et al. 2019) On kuitenkin huomattava, että teknologian hyödyntämisessä on paljon huomioitavia asioita, jotta sitä voidaan käyttää mahdollisimman tehokkaasti sekä turvallisesti. Sanchis et al. (2019) tuovat artikkelissaan esiin sen, että low-code -kehitysalustoista ei tiedetä vielä kovinkaan paljoa. Toisaalta pelätään myös sitä, että lukkiudutaan low-code -ratkaisun toimittajan kanssa. Koska low-code -alustoista ei tiedetä vielä kovinkaan paljoa, eikä sen mahdollisuuksia ymmärretä, niin sen vuoksi pelätään sitä, että sillä ei ole mahdollista tehdä sellaista sovellusta kuin on tarve sekä kyseenalaistetaan skaalautuvuus. (Sanchis et al. 2019) Se miksi low-code -alustoista ei vielä tiedetä paljoa, voidaan olettaa liittyvän teknologian uutuuteen, terminäkin low-code on keksitty vasta vuonna 2014. Myös low-codella toteutettujen sovellusten turvallisuus huolettaa, minkä voidaan olettaa liittyvän osittain myös tiedon puutteeseen (Sanchis et al. 2019).

Kun mietitään low-code -teknologian tulevaisuutta voidaan nähdä, että sillä on valtava potentiaali kehittyä ja se onkin jo suosion alkuvaiheessa kehittynyt paljon. Sundberg & Holmström (2023) mukaan tulevaisuudessa on syntymässä yhä enemmän uusia digitaalisia alustoja, jotka ovat ns. ihmisen ja tekoälyn välisiä hybridijärjestelmiä. Käytännön esimerkkinä on Microsoftin esittelemä Copilot, jonka avulla voidaan mahdollistaa tekoälyavustaja sovelluksen loppukäyttäjälle. Loppukäyttäjät voivat hyödyntää Copilot-ominaisuutta luonnollisella kielellään saaden lisätietoa haluamistaan tiedoistaan. Copilot on lisättävissä esimerkiksi Microsoftin Power Apps-alustalla tehtyyn low-code -sovellukseen. (Microsoft 2023)

Low-code -teknologian suosion kasvaessa myös siihen liittyvien palveluiden kysynnän voidaan olettaa kasvavan. Palveluilla on vaikutusta, mutta ei muotoa, eikä niitä voida fyysisesti varastoida tai hallita (Shostack 1984). Palvelut ovat riippuvaisia työvoimasta, jonka vuoksi niiden jäljittely on paljon vaikeampaa, kuin tuotteiden kohdalla. Tästä syystä palvelut nähdään kilpailuedun lähteenä. (Oliva & Kallenberg 2003) Jotta low-code -palveluiden näkökulmasta pystytään olemaan kilpailukykyisiä, tulee yritysten pohtia myös low-code -palveluiden tuotteistamista. Parantainen (2007) mukaan asiantuntijapalveluiden tuotteistamisessa korostuu dokumentointi, jatkuva kehitystyö, prosessikuvaukset, kurssit sekä ohjeet. Dokumentoinnin merkitystä ei voi korostaa tarpeeksi, mikäli halutaan monistaa ammattitaitoa ja kutsua palvelua palvelutuotteeksi (Parantainen 2007).

Sipilä (1995) mukaan palvelun onnistuminen vaatii asiakkaan osallistumista, jotta palvelun tuottaja voi onnistua. Hänen mukaansa yhteistyö palvelun tuottajan ja asiakkaan välillä on avainasemassa, kun mietitään miten asiantuntijapalvelut syntyvät. Myös low-code -teknologian ympärillä on tunnistettu erilaisia malleja, jossa eri liiketoiminnan osa-alueilta hyödynnetään osaamista yhteistyössä. Kirjallisuudesta tunnistetaan useita termejä yhteistyökehittämisen ympäriltä, kuten yhteistuotanto, yhteisluominen, fuusiokehittäminen ja fuusiotiimit. Ayyar et al. (2023) mukaan fuusiotiimit ovat yksi keino edistää yrityksen ketterää digitaalista liiketoimintaa. Fuusiotiimien tarkoituksena on, että yhdistetään teknologian sekä liiketoiminta-alueen asiantuntemusta. Yhteistyö eri liiketoiminnan asiantuntijoiden kanssa lisää avoimuutta esimerkiksi palautteen annossa, jolloin toimintaa voidaan ymmärtää ja muotoilla paremmin. Yhteisluominen auttaa tekemään parempia päätöksiä monella eri osa-alueella. Parhaassa tapauksessa fuusiotiimeillä saavutetaan kilpailuetua ja huipputason kasvua digitalisaatiossa. (Ayyar et al. 2023) Voidaan siis tunnistaa yhteistyökehittämisen merkitys sekä yhteys niin low-code -teknologian, että palveluiden näkökulmista. Tässä tutkimuksessa käytetään käsitettä yhteistyökehittäminen, jonka ympärille on koottu tieto edellä mainittujen käsitteiden kirjallisuudesta.

1.2 Tutkimuksen tavoitteet ja rajaukset

Tutkimuksen tavoitteena on luoda malli yhteistyökehittämiseen low-code -teknologialla, jota voidaan hyödyntää myös palvelun tuotteistamisessa. Mallin tavoitteena on helpottaa ja selkeyttää sisäistä liiketoimintaa niin low-code -kehittämisen kuin tuotteistamisen näkökulmista. Oletetaan, että mallin avulla sisäinen tekeminen low-coden parissa selkeytyy, jolloin asiantuntijuutta ja osaamista on helpompaa esitellä ulkopuolelle. Tavoitteena on pohtia yhteistyökehittämistiimin kokonaisuutta, sen kokoa ja jäseniä sekä sitä, miten se tulisi muodostaa. Tavoitteena on ottaa kantaa ratkaisun elinkaaren näkökulmasta yhteistyökehittämisen malliin. Miten saadaan kehitettyä palvelua niin, että se mahdollistaa mahdollisimman hyvän tuotteen luonnin, joka on myös jatkossa kehitettävissä. Huomataan siis se, että tutkimuksessa tarvitsee ottaa myös kantaa siihen millaisella kehitysmallilla low-code -ratkaisuja kannattaisi tehdä. Tuotteistamisessa pohditaan sekä ulkoista, että sisäistä tuotteistamista. Tässä kohtaa oletetaan, että yhteistyökehittämisen mallin luonti antaa mahdollisuuden palvelun tuotteistamiselle. Palvelujen tuotteistamisen näkökulmasta tavoitteena on selkeyttää sisäisiä prosesseja sekä parantaa prosesseja, jotka näkyvät myös ulkopuolelle.

Tutkimus toteutetaan yhteistyössä case-organisaation Navakka Group Oy:n kanssa. Case-organisaatio on keskiössä erityisesti tutkimuksen empiirisessä osuudessa.

Navakka Group Oy on vuonna 2018 perustettu pilvipohjaisen toiminnanohjausjärjestelmän Microsoft Dynamics 365 Business Centralin toimittaja (Navakka 2023, a). Case-organisaatio tarjoaa myös lisäarvotkaisuita toiminnanohjausjärjestelmään sekä sen ympärille (Navakka 2023, b). Tässä tutkimuksessa keskitytään low-codella toteutettuihin lisäarvotuotteisiin, erityisesti sovelluksiin.

Pohditaan seuraavaksi tutkimuskysymyksiä, jotka toimivat ohjaavina tekijöinä koko tutkimuksen ajan. Tavoitteena on löytää selkeät vastaukset kysymyksiin tutkimuksen lopussa.

Päätutkimuskysymys:

1. Miten low-code -palveluja voidaan tuotteistaa yhteistyökehittämisen mallin avulla?

Päätutkimuskysymyksen tarkoituksena on löytää vastaus siihen, miten palvelujen tuotteistaminen, yhteistyökehittäminen sekä low-codella tehdyt teknologiset ratkaisut liittyvät toisiinsa ja miten ne yhdessä voisivat muodostaa parhaan mahdollisen tavan toimia toimeksiantajan kontekstissa.

Alatutkimuskysymykset:

1. Mitä palveluiden tuotteistaminen tarkoittaa low-code -teknologialle?
2. Miten yhteistyökehittämistä voidaan hyödyntää tuotteistamisessa?

Alatutkimuskysymysten tarkoituksena on tukea laajempaa päätutkimuskysymystä. Ensimmäisen alatutkimuskysymyksen avulla pyritään löytämään vastaus siihen, mitä palvelujen tuotteistaminen tarkoittaa ja miten sitä voisi hyödyntää low-code -teknologian ympärillä. Toisella alatutkimuskysymyksellä pyritään löytämään vastaus siihen, mitä yhteistyökehittäminen on ja miten ne liittyvät palveluiden tuotteistamiseen.

2. TUTKIMUKSMENETELMÄT JA TOTEUTUS

Luvussa kuvataan tutkimusmenetelmät, lähestymistavat ja filosofia. Lisäksi esitellään aiempaa kirjallisuutta, jota tutkimuksessa hyödynnetään. Luvun tarkoituksena on selkeyttää kokonaisuudessa tutkimuksen toteutustapaa ja valintoja.

2.1 Lähestymistavat ja tutkimusfilosofia ja -strategia

On oletettavaa, että tutkijalla on olettamuksia aihepiiriin ympäriltä ennen tutkimuksen tekemistä. Tämä ohjaa tutkijaa, jonka vuoksi tutkimusfilosofiaa on tärkeää pohtia. Saunders et al. (2019) esittelee viisijakoisen näkemyksen eri tutkimuksen tarkoituksista, liiketoiminnan ja johtamisen filosofioista: positivismi, kriittinen realismi, interpretivismi, postmodernismi sekä pragmatismi. Perehtymisen jälkeen voidaan todeta, että pragmatismi kuvailee parhaiten tässä tutkimuksessa käytettävää filosofiaa tutkimuksen tavoitteiden mukaisesti. Pragmatismi on saanut alkunsa 1800–1900 lukujen taitteessa, jonka lähestymistapana on tarkastella käytännöllisellä tasolla maailmaa. Pragmatismi tarkastelee teorioita, käsitteitä, ideoita ja hypoteeseja toiminnan keinoin. Pragmatikolle ominaista on, että käytännön vaikutukset ovat tärkeitä ja ratkaisuja löydetään käytännön ongelmiin. (Saunders et al. 2019)

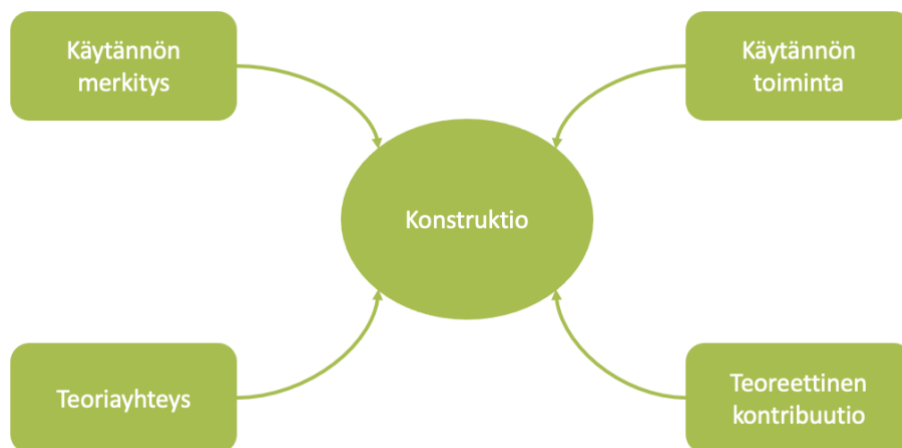
Pohditaan seuraavaksi lähestymistapoja teorian kehittämiseen. Saunders et al. (2019) esittelee kolme eri tapaa: induktiivinen, deduktiivinen ja abduktiivinen päättely. Deduktiivisessa lähestymistavassa suunnitellaan tutkimusstrategia teorian testaamiseksi. Ajatuksena siinä siis on, että lähdetään teoriasta liikkeelle ja vahvistetaan tai kumotaan hypoteeseja. Induktiivisesta lähestymistavasta puhutaan silloin, kun luodaan uutta teoriaa keräämällä tietoa. Lähdetään siis liikkeelle havainnoista tavoitteena luoda uutta teoriaa. Abduktiivisella lähestymistavalla voidaan tunnistaa teemoja, kerätä tietoa, luoda uutta teoriaa tai muokata olemassa olevaa teoriaa. Se on siis induktiivisen ja deduktiivisen lähestymistavan yhdistelmä. (Saunders et al. 2019) Tutkimuksen lähestymistapa on abduktiivinen, sillä teorian ja empiirisen tutkimuksen välillä tulla tekemään havaintoja kumpaankin suuntaan.

Pohditaan seuraavaksi tutkimusstrategioita, joita kvalitatiiviseen tutkimukseen liittyy. Saunders et al. (2019) esittelee tärkeimmät tutkimusstrategiat, jotka liittyvät kvalitatiiviseen tutkimukseen: toimintatutkimus, tapaustutkimus, etnografia, grounden teoria sekä narratiivinen tutkimus. Tutkimusstrategialla on suuri merkitys siihen, kuinka tutkija aikoo vastata tutkimuskysymykseensä. Lopputuloksena todetaan, että sopivin

tutkimusstrategia tässä kontekstissa on tapaustutkimus. Tapaustutkimuksessa pyritään ymmärtämään tutkittavaa aihetta tapauksen kontekstissa. Tapaus viittaa tässä kontekstissa toimeksiantajaan eli case-organisaatioon. (Saunders et al. 2019)

2.2 Konstruktiivinen tutkimus

Konstruktiivinen lähestymistapa on pragmaattinen, joka mahdollistaa käytännön ongelmanratkaisun sekä tieteellisen teorian yhdistämisen (Pasian 2015). Konstruktiivisessa lähestymistavassa olennaista on, että ongelma ja ratkaisu sidotaan teoreettiseen tietoon. Toisaalta myös todellinen toimivuus on osoitettava. (Kasanen et al. 1993). Parhaassa tapauksessa jokin ongelma ratkaistaan ottamalla uusi ratkaisu käyttöön, kuten hallintatyökalu, tekniikka tai prosessi (Pasian 2015). Kuvassa 1 esitellään konstruktiivisen tutkimuksen elementit, jotka ovat käytännön merkitys ja toiminta, teoriayhteys sekä teoreettinen kontribuutio.

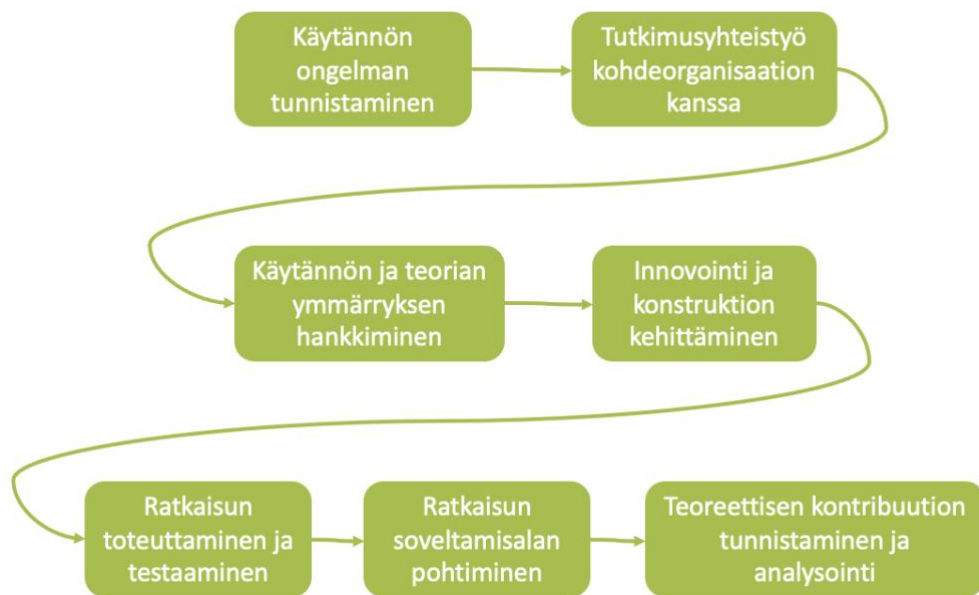


Kuva 1: Konstruktiivisen tutkimuksen elementit (mukaillen lähteestä Kasanen et al. 1993).

Kun tarkastellaan kuvaa 1, on perusteltua, että tässä tutkimuksessa hyödynnetään konstruktiivista lähestymistapaa. Tutkimuksessa sidotaan tutkittava ongelma ja ratkaisu teoriaan, toisaalta sillä on käytännön merkitystä case-organisaatiossa. Tutkimuksen tavoitteena on, että ratkaisulla on vaikutusta case-organisaation ja asiakkaiden käytännön toimintaan sekä sen todellinen toimivuus esitetään testaamalla muodostettua ratkaisua.

Konstruktiivisessa lähestymistavassa olennaista on, että löydetään relevantti käytännöllinen ongelma, joka on myös tutkimuksen näkökulmasta potentiaalinen. Tarkoituksena on hankkia kattava käsitys aiheesta, joka tässä tutkimuksessa toteutetaan

aiemman kirjallisuuden sekä empiirisen tutkimuksen avulla. Empiirinen osuus keskittyy haastatteluihin case-organisaation kontekstissa, jonka toteutus esitellään myöhemmin luvussa 2. Sen jälkeen, kun tutkijalla on kattava käsitys aiheesta, innovoidaan ratkaisuidea. Jotta pystytään todentamaan ratkaisun toimivuus, tulee se varmistaa esimerkiksi testaamalla. Tässä tutkimuksessa innovoitua mallia tullaan testaamaan konstruktiivisen lähestymistavan mukaisesti. Lopuksi ratkaisun teoreettista yhteyttä ja tutkimuksen kontribuutio tulee esittää sekä pohtia ratkaisun sovellettavuutta. (Kasanen et al. 1993) Lukka (2006) esittelee konstruktiivisen tutkimusprosessin, joka on tiivistetty kuvaan 2.



Kuva 2: Konstruktiivisen tutkimuksen prosessi (mukaillen lähteestä Lukka 2006).

Kun tarkastellaan kuvaa 2, nähdään selkeä prosessimalli, jota tämä tutkimus noudattaa. Käytännön ongelma on tunnistettu yhdessä case-organisaation kanssa, johon pyritään tutkimuksen tavoitteiden mukaisesti löytämään ratkaisu. Samalla case-organisaation kanssa tehdään tutkimusyhteistyötä niin, että tutkija sekä case-organisaatio ovat sitoutuneet koko tutkimusprosessiin. Tämä näkyy erityisesti empiirisessä osuudessa, jossa haastatellaan case-organisaation työntekijöitä sekä asiakkaita. Aihepiirin teorian ymmärrys hankitaan tutkimalla aiempaa kirjallisuutta. Empiirinen osuus vastaa käytännön ymmärryksen hankkimista, mikä tässä tutkimuksessa toteutetaan puolistrukturoiduilla haastatteluilta. Tutkimuksen tavoitteena on luoda yhteistyökehittämisen malli low-code -teknologialle ja hyödyntää sitä palvelun tuotteistamisessa, mikä vastaa prosessimallissa innovointi ja konstruktion kehittämisen -vaihetta. Kun ratkaisu eli malli saadaan kehitettyä, sitä testataan ja todennetaan sen

toimivuus. Lopuksi tullaan pohtimaan sen soveltamisalaa sekä analysoimaan teoreettista kontribuutiota.

Perehdytään seuraavaksi tarkemmin, mitä teoreettisella kontribuutiolla tarkoitetaan sekä pyritään ymmärtämään, miten kuvan 2 viimeinen kohta voidaan tässä tutkimuksessa saavuttaa. Suomen kielessä kontribuutiolla tarkoitetaan mm. panosta tai myötävaikutusta, mutta myös sana kontribuutio on yleisesti käytetty. Zhou et al. (2017) mukaan teoreettinen kontribuutio on laaja ja tärkeä kysymys, jossa tutkijan on otettava huomioon, miksi lisätään tai vähennetään teorian muuttujia olemassa olevista muuttujista. Toisaalta tutkijan tulee pohtia myös, kuinka muuttujan muutos vaikuttaa olemassa olevaan teoriaan (Zhou et al. 2017). Lukka (2001) korostaa, että teoreettisen kontribuution tunnistaminen ja analysointi onnistuu, mikäli tutkija pystyy ottamaan etäisyyttä omaan empiiriseen työhönsä. Zhou et al. (2017) esittää, että teoreettinen kontribuutio voidaan jakaa kahteen ulottuvuuteen, joista ensimmäinen ulottuvuus on omaperäisyys. Tämän avulla pyritään selittämään, lisätäänkö olemassa olevaan teoriaan ylimääräistä muuttujaa ja kuinka tämä vaikuttaa nykyisiin ilmiöihin. Eli tavoitteena on esittää, kuinka uusi muuttuja parantaa tai muuttaa olemassa olevaa teoreettista viitekehystä. Toinen näkökulma, johon omaperäisyyden ulottuvuus ottaa kantaa on se, kuinka teoria paljastaa piilotetut totuudet, joita ei ole vielä löydetty eli johtaako tämä uuden teorian kehittämiseen. (Zhou et al. 2017)

Teoreettisen kontribuution toinen ulottuvuus on hyödyllisyys, joka voidaan jakaa tieteellisiin- sekä käytännön hyötyihin. Tieteellisen hyödyn mittarina toimii teorian hyödyllisyys muilla tutkimusaloilla. Käytännön hyötyjä voidaan mitata teoreettisten käsitteiden käytännön toteuttamisella organisaatioissa ja siten on löydettävä käsitteiden vaikutus ja tulos. Teoreettinen kontribuutio on tunnistettu ja analysoitu vasta, kun muuttujien yhteen- ja vähennyslasku on perusteltu ja vaikutukset löydetty. (Zhou et al. 2017)

Kasanen et al. (1993) esittelee kolmiportaisen mallin konstruktion testaamiseen: heikko markkinatesti, puolivahva markkinatesti sekä vahva markkinatesti. Mikäli johtaja on valmis soveltamaan konstruktiota varsinaisessa päätöksenteossa, voidaan todeta, että heikko markkinatesti on suoritettu. Keskivahvan markkinatestin läpäisystä voidaan puhua, kun konstruktiio on yleistynyt yrityksissä. Vahva markkinatesti puolestaan tarkoittaa, että konstruktiota hyödyntäneet liiketoimintayksiköt ovat menestyneet systemaattisesti paremmin, kun tarkastellaan taloudellisia tuloksia. (Kasanen et al. 1993) Tässä tutkimuksessa testaaminen suoritettiin heikolla markkinatestillä, esittelemällä luotu malli yhdelle case-organisaation asiakkaalle ja yhdelle case-organisaation työntekijälle. Molemmat heistä osallistui myös empiirisen osuuden haastatteluihin.

Henkilöt toimivat asiantuntija sekä johtotehtävissä. Asiakkaan kanssa mallin testaaminen suoritettiin etäpalaverissa, kun case-organisaation työntekijä tavattiin kasvotusten. Henkilöille kerrottiin etukäteen, että heiltä tullaan kysymään mielipidettä ja näkemystä luotuun malliin sekä arvioimaan mallien selkeyttä. Tavoitteena oli siis selvittää, kuinka ymmärrettävät ja selkeät luodut mallit ovat sekä voisiko henkilöt hyödyntää luotuja malleja omassa työssään.

Tapaamisten alussa kerrottiin tarkemmin, että mallin testaaminen on osa konstruktivistista tutkimusta ja mikä kirjallisuuden määritelmä on heikosta markkinatelistä. Henkilöille esiteltiin luvun 7.1 mukaisesti yhteistyökehittämisen malli (kuvat 23, 24 ja taulukko 6) ja tämän jälkeen heitä haastateltiin muutamien kysymysten kautta (liite C). Kysymykset ja mallien kuvat olivat esillä haastateltaville koko palaverin ajan. Haastatteluissa painotettiin kuitenkin kysymysten sijaan haastateltavien näkemyksiä ja ajatuksia malleista, jonka takia esiintyi myös jatkokysymyksiä ja vapaampaa keskustelua haastateltavan sekä tutkijan puolesta. Haastateltavilla oli aiempaa kokemusta low-code -teknologiasta ja erilaisista low-codella tehdyistä ratkaisuista. Testin tulokset esitellään luvussa 7.3.

2.3 Kirjallisuus

Salminen (2011) määrittelee, että kirjallisuuskatsauksia on kolmea eri tyyppiä: kuvaileva ja systemaattinen kirjallisuuskatsaus sekä meta-analyysi. Kuvaileva kirjallisuuskatsaus on yleiskatsaus aiheesta ilman tiukkoja sääntöjä tai rajoituksia. Kuvaileva kirjallisuuskatsaus voidaan jakaa kahteen alaosiin narratiiviseen sekä integroivaan. Narratiivisen kirjallisuuskatsauksen tarkoituksena on antaa laaja kuva aiheesta. Pyritään löytämään aiheesta paljon tietoa helposti luettavaksi. Integroivassa kirjallisuuskatsauksessa tarkoituksena on kuvailla aihetta mahdollisimman laajasti. Tämä tapa ei seulo niin tarkasti tutkimusaineistoa kuin systemaattinen kirjallisuuskatsaus, jossa käydään läpi runsaasti tutkimusmateriaalia systemaattisesti. Systemaattinen tapa tehdä kirjallisuuskatsaus luo tutkimukselle tieteellistä uskottavuutta. Salminen (2011) kuvaa Finkin (2005) mallia tehdä systemaattinen kirjallisuuskatsaus, joka on seitsemänvaiheinen. Kolmantena esitellään meta-analyysi, joka voidaan jakaa vielä kvalitatiiviseen sekä kvantitatiiviseen. Kvantitatiivisessa meta-analyysissä hyödynnetään tilastotieteen menetelmiä yhdistäen ja yleistäen kvantitatiivisia tutkimuksia toisiinsa. Kvalitatiivinen meta-analyysi jaetaan kahteen eri osaan metasynteesiin sekä metayhteenvetoon. Metasynteesi selittää ja pyrkii ymmärtämään tutkittavaa aihetta, kun metayhteenveto tiivistää laadullisen tutkimuksen kvantitatiivisilla menetelmillä. (Salminen 2011)

Tässä työssä tullaan muodostamaan kirjallisuudesta teoreettinen viitekehys systemaattisen tiedonhaun menetelmin. Tutkimuskatsauksen avulla pyritään kuvaamaan käsitteellistä taustaa sekä perustelevaan, miten diplomityö liittyy jo aiempaan tutkimukseen. (Kunnela 2022) Systemaattista kirjallisuuskatsausta tullaan siis hyödyntämään osittain, etenkin tiedonhaun osalta.

Esitellään seuraavaksi tietokannat, joista hakuja tehtiin. Tietokantojen valinnassa hyödynnettiin Tampereen yliopiston kirjaston kokoamaa listausta tietokannoista. Lista suodatettiin tietojohtamisen alan tietokantoihin selkeyttämään valintaa. Listaus sisältää kuvaukset tietokannoista, joiden perusteella valinnat tehtiin. Tietokannat, joista hakuja tehtiin ovat Google Scholar, Emerald, Gartner, Web of Science sekä ScienceDirect. Hakutermit, joilla hakuja tehtiin, valittiin aihepiirin käsitteiden avulla yksittäin sekä yhdistelemällä. Taulukkoon 1 on koottu esimerkkejä hakutermeistä ja -lausekkeista, joilla hakuja tehtiin edellä mainituista tietokannoista.

Taulukko 1: Hakutermit ja -lausekkeet aihealueittain.

Aihealue	Hakutermit ja -lausekkeet
Low-code	"low-code"
Yhteistyökehittäminen	"fusion teams"
	"co-development"
	"co-development" AND "low-code"
	"co-creation" AND "low-code"
	"collaborative development" and low-code
	"co-creation" AND "low-code"
	"low-code" AND "fusion teams"
	"value co-creation"
Palvelut ja tuotteistaminen	"low-code" AND "productization"
	"service productization"
	"service thinking"
	"service packaging"
	service productization process model

Taulukosta 1 nähdään, että hakulausekkeissa on yhdistelty eri käsitteitä sekä hakuja on tehty myös yksittäisillä sanoilla. Kuten aiemmin mainittiin, taulukossa 1 esitetyt hakulausekkeet ovat esimerkkejä eli jokaista hakulauseketta ei ole taulukossa esitetty. Muutamia esimerkkejä halutaan kuitenkin esittää, koska ne antavat viitteitä siitä, millaisella haulla aiheen ympäriltä on mahdollista löytää tietoa tutkimuksen toistettavuutta ajatellen. Haussa tehtiin myös helmenkasvatusta eli hyödynnettiin relevanttien tutkimusten lähdeluetteloita tiedonhaussa. Kirjallisuudesta huomattiin, että yhteistyökehittäminen ei ole käsitteenä kovinkaan vakiintunut termi yksittäin, vaan sitä käytetään rinnakkain usean eri käsitteen kanssa, kuten yhteisluominen, yhteistuotanto ja fuusiokehittäminen. Tästä syystä hakuja on tehty usealla eri termillä, kuten taulukosta 1 nähdään. Tässä tutkimuksessa hyödynnetään eri käsitteiden alta teoriaa, jotka

tiivistetään yhteistyökehittämisen alle. Low-code -teknologian liittyvä tutkimus on hyvin uutta ja tästä syystä teoriassa hyödynnetty kirjallisuus on viime vuosilta. Yhteistyökehittämisen ympäriltä kirjallisuutta löydettiin eri termien vuoksi nykyhetkestä, mutta myös historiasta. Palveluista ja niiden tuotteistamisesta löydettiin paljon kirjallisuutta jopa 20–30 vuoden takaa. Huomattavaa oli, että monet viitekehykset ja prosessimallit esiintyvät edelleen nykyhetken kirjallisuudessa, minkä vuoksi myös tässä tutkimuksessa nähtiin relevanttina hyödyntää näitä tutkimuksia.

2.4 Haastattelu tutkimusmenetelmänä sekä tutkimuksen otanta

Haastattelututkimus on tapa kerätä tietoa, käsityksiä, uskomuksia sekä sen avulla voidaan ymmärtää miksi ihmiset toimivat jollakin tietyllä tapaa. Toisaalta haastattelu voidaan nähdä myös keskusteluna, jossa on tietty tarkoitus, vaihdetaan ajatuksia haastateltavan ja haastattelijan kesken. Haastattelututkimusta tehdessä on erittäin tärkeää, että otetaan huomioon tutkimusongelma ja rajaukset, sillä näillä on suuri merkitys lopputuloksia ajatellen. (Hirsjärvi & Hurme 2022) Pohditaan seuraavaksi syitä, miksi empiiriseen osuuteen valittiin haastattelu tutkimusmenetelmäksi. Hirsjärvi & Hurme (2022) tuo esiin neljä kohtaa, joita tulisi miettiä, kun menetelmävalintoja tehdään: tehokkuus, taloudellisuus, tarkkuus ja luotettavuus. He kokoavat myös etuja ja haittoja, joita itse haastattelututkimuksessa on havaittu (taulukko 2).

Taulukko 2: Haastattelun edut ja haitat (mukailtu lähteestä Hirsjärvi & Hurme 2022).

Edut	Haitat
Ihminen on subjekti	Haastattelijan taitoa ja kokemusta vaaditaan
Ihmisen puhe sijoitetaan laajempaan kontekstiin	Haastattelijan rooli haastava, vaatii koulutusta
Vastausten selventäminen	Haastattelu vie aikaa
Saatavien tietojen syventäminen	Haastattelu sisältää virhelähteitä
Arkojen ja vaikeiden aiheiden tutkiminen	Haastattelusta aiheutuu kustannuksia
	Vapaamuotoisessa haastatteluissa analysointi, tulkinta ja raportointi ongelmallista

Hirsjärvi & Hurme (2022) tuovat esiin näkemyksiä siitä, miksi haastattelu kannattaisi valita ja mitä asioita ylipäättään tutkimusmenetelmää valittaessa tulisi huomioida. Tämän perusteella voidaan todeta, että haastattelulla pystytään vastaamaan tutkimuskysymyksiin sekä se on taloudellisesti järkevä tapa toteuttaa tutkimusta. Haastatteluiden avulla pystytään keräämään tarkasti tietoa, sillä vastauksia pystytään syventämään ja tarkentamaan haastattelun aikana. Luotettavuutta voidaan lisätä tarkoilla muistiinpanoilla sekä nauhoituksella. Myös haastateltavien valinnalla on merkitystä, jonka vuoksi tavoitteena oli, että tutkimukseen valitaan relevantteja haastateltavia. Tällöin voidaan olettaa, että vastaukset ja tieto, jota tutkimuksessa kerätään, on arvokasta. Tärkeää on kuitenkin huomioida taulukossa 2 esiin nostetut

haitat, sillä tiedostamalla ongelmat etukäteen voidaan pyrkiä minimoimaan niiden vaikutukset.

Haastattelut voidaan jakaa eri lajeihin kuten strukturoitu ja strukturoimaton haastattelu. Hirsjärvi & Hurme (2022) esittelevät myös puolistrukturoidun haastattelun eli teemahaastattelun. Tämä on laji, jolla tämän tutkimuksen haastattelut toteutettiin. Lajina hyödynnettiin teemahaastattelua, sillä se tuo tietyt raamit haastattelulle, mutta antaa kuitenkin haastattelijalle vapauden haastattelun aikana esimerkiksi vaihtaa kysymysten järjestystä sekä muotoilla kysymyksiä eri tavoin. Teemahaastattelussa oletetaan myös, että haastateltavat ovat kokeneet tietyn tilanteen. Tämä oli huomioitava asia haastateltavia valittaessa. Teemahaastattelussa tyypillistä on myös se, että se jaetaan tiettyihin teemoihin, joka helpottaa niin haastateltavaa, että haastattelijaa. Teemahaastattelu ei sido myöskään haastattelukierrosten määriin vaan pureutuu nimenomaan tarkastelemaan haastatteluita teemojen ympäriltä. (Hirsjärvi & Hurme 2022)

Tutkimuksessa tehtiin haastatteluita toimeksiantajan asiakkaille sekä työntekijöille. Asiakkaat valittiin niin, että he ovat aihealueen ympäriltä relevantteja haastateltavia. Haastattelurunkoa muodostaessa huomioitiin se, että kysymykset ovat puolueettomia eivätkä ne anna mahdollisuutta haastateltavan ohjaukseen. Hirsjärvi & Hurme (2022) toteavat, että haastattelut ovat yhteistyön tulos, jolloin on riskinä, että haastattelija vahvistaa haastateltavan ajatuksia omilla näkemyksillään. Toinen tärkeä tekijä on, että haastattelurunko rakennettiin teemahaastattelun mukaisesti eli keskiössä pidettiin teemat, joita haluttiin tarkastella.

Jotta tutkimuksen toteutus on mahdollisimman läpinäkyvä, tarkastellaan seuraavaksi eettisiä näkökulmia. Haastatteluja suorittaessa on tärkeää muistaa haastateltavien suostumus, luottamuksellisuus sekä tietosuoja (Hirsjärvi & Hurme 2022). Haastateltavalle onkin siis hyvä tuoda esiin tutkimuksen tarkoitus sekä haastattelun purkaminen eli litterointi. Haastateltavien on hyvä saada vastaukset näiltä osin ennen varsinaisen haastattelun aloitusta, parhaassa tapauksessa, jo viestissä, missä kysytään suostumusta haastatteluun. Huomioitavaa on kuitenkin raja tarpeellisen tiedon esittämisestä, jotta haastatteluiden vastaukset eivät vääristy (Hirsjärvi & Hurme 2022). Tietosuojan näkökulmasta tulee miettiä myös tietosuojailmoituksen tekemistä, mikäli esimerkiksi ääntä nauhoitetaan, jolloin kerätään henkilötietoja. Tutkimuksessa huomioitiin raportoinnin näkökulma, josta Hirsjärvi & Hurme (2022) tuo esiin luottamuksellisuuden sekä mahdolliset seuraukset, joita haastateltaville näistä seuraa. Tutkimuksessa huomioitiin haastateltavien anonymisointi, paikka mihin tiedot

haastatteluista tallennettiin sekä mitä kaikkea tietoa lopulliseen diplomityöhön raportoitiin.

2.5 Haastatteluiden toteutus ja haastatteluprosessi

Haastattelurunko rakennettiin tutkimuksen teoriaan pohjautuen. Koska haastattelut toteutettiin teemahaastatteluina, pohdittiin teemoja, joihin haastattelun kysymykset jaoteltiin. Tutkimuksen tavoitteiden mukaan valittiin abduktiivinen päättely tutkimuksen lähestymistavaksi, joka mahdollistaa teorian täydentämisen sekä vuoropuhelun empiirisen osuuden kanssa (Saunders et al. 2019). Liitteissä A ja B on nähtävissä sekä asiakkaille, että toimeksiantajan edustajille esitetyt kysymykset. Asiakashaastatteluiden tavoitteena oli ymmärtää heidän näkökulmiansa yhteistyökehittämisestä low-code -teknologian kontekstissa. Näiden vastausten analysoinnilla pyrittiin myös ymmärtämään, mitä tällä hetkellä palveluntarjoajalla tehdään oikein ja missä asioissa voisi kehittyä. Analyysit tukevat palvelun tuotteistamista ja prosessien selkeyttämistä. Asiakashaastatteluihin valittiin henkilöitä, joilla on kokemusta tai ymmärrystä jollakin tasolla low-code -ratkaisuista. Tämä valinta tehtiin siitä syystä, että pystyttiin kysymään kysymyksiä olettamuksella, että haastateltavalla on käsitys, mitä low-code -teknologia tarkoittaa ja miten ratkaisuita on aiemmin toteutettu. Taulukossa 3 esitellään haastateltavat sekä yritykset koodattuna.

Taulukko 3: Haastateltavat koodattuina, joihin viitataan empiirisessä osuudessa.

Haastateltava	Yritys
H1	A
H2	B
H3	C
H4	D
H5	A
H6	E
H7	F
H8	F

Taulukosta 3 nähdään, että asiakashaastatteluita oli 6 kpl, joista kaksi H1 ja H5 työskentelevät samassa yrityksessä. Toinen kohderyhmä oli toimeksiantajan haastattelut eli case-organisaation ammattisovelluskehittäjien haastattelut. Taulukkoon 3 nämä haastateltavat on koodattu harmaalle pohjalle lyhenteillä H7 ja H8. Tämän ryhmän haastateltavat valikoituivat myös niin, että henkilöillä oli aiempaa kokemusta low-code -ratkaisujen toteuttamisesta. Myös tässä tapauksessa rajaus tehtiin heihin, sillä tällöin kysymysrunkoon voitiin tehdä tiettyjä olettamuksia. Molemmissa kohderyhmissä

ajateltiin, että saadaan mahdollisimman relevantteja vastauksia, sillä low-code -teknologia ja sen hyödyntäminen on kuitenkin suhteellisen uutta. Valintoja voidaan perustella myös tutkimuksen tavoitteella, jonka tarkoituksena on löytää parhaita käytäntöjä eikä niinkään markkinoida tai lisätä low-code -teknologian tunnettavuutta.

Asiakashaastattelut järjestettiin virtuaalisesti etänä Teams-sovelluksella ajankäytön tehostamiseksi sekä minimoidaksemme turhan matkustamisen kaupungista toiseen. Haastateltaville lähetettiin alustavat haastattelupyynnöt sähköpostitse, jossa kysyttiin kiinnostusta osallistua haastatteluun. Viestissä korostettiin haastatteluiden vapaaehtoisuutta sekä informoitiin, että vastaukset tullaan anonymisoimaan lopulliseen diplomityöhön. Haastattelukutsu sisälsi aiheen sekä haastattelun teemat, mutta haastattelukysymyksiä ei haastateltaville lähetetty etukäteen. Haastateltaville lähetettiin myös tietosuojailmoitus ennen haastatteluita, sillä haastattelut nauhoitettiin, jolloin kerättiin henkilötietoja. Koska haastattelussa noudatettiin puolistrukturoitua mallia, haastattelussa kysymysten järjestystä vaihdeltiin ja muotoiltiin vapaasti. Kysymysten paikkoja vaihdettiin erityisesti ensimmäisen ja toisen haastattelun jälkeen, kun huomattiin, että kyseinen järjestys oli parempi. Haastattelut toteutettiin satunnaisessa järjestyksessä eli haastattelujärjestykseen ei mietitty etukäteen rakennetta. Haastatteluiden ajankohta määräytyi asiakkaiden sekä palveluntarjoajan työntekijöiden aikataulujen sekä tutkimussuunnitelmassa esitetyn aikataulun tavoitteiden mukaisesti elo-syyskuun 2023 aikana.

Case-organisaation työntekijöiden haastateltavat toimivat teknisen konsultin sekä kehittäjän rooleissa. Haastateltavilla on siis kokemusta ohjelmistokehityksestä, low-code -kehityksestä sekä konsultoinnista. Asiakashaastatteluissa haastateltavat sijoituivat eri rooleihin IT:n ja liiketoiminnan välille. Jokaisen roolissa toistui vastuu liiketoiminnan ja IT:n kehittämisestä eri toimialan yrityksissä. Asiakkaat toimivat eri kokoisissa yrityksissä, millä voidaan olettaa olevan merkitystä vastauksiin.

2.6 Aineiston analysointi

Laadullisessa tutkimuksessa on käytettävänä useita eri analyysimenetelmiä, joiden tarkoituksena on tuoda selkeyttä tutkimukseen sekä luoda uutta tietoa tutkittavasta asiasta tai ilmiöstä. Analyysin avulla voidaan myös tiivistää aineistoa, mikä helpottaa tutkijaa tunnistamaan merkittävät tekijät aineistosta. Analyysin tulkintaan Eskola & Suoranta (1998) esittelee kaksi eri lähestymistapaa: tulkintojen rakentaminen aineistosta tai toisaalta aineistoa voidaan pitää teoreettisen ajattelun lähtökohtana. Tunnistetaan, että tulkinnat ja niiden tekeminen on haastava vaihe tutkimuksessa, jonka vuoksi tutkimuksen analysoinnin ja analyysimenetelmän valintaan kiinnitettiin erityisen paljon

huomiota. Analyysimenetelmän avulla datasta voidaan luoda informaatiota, josta edelleen voidaan tehdä tulkintoja. (Eskola & Suoranta 1998)

Eskola & Suoranta (1998) mukaan aineisto voidaan purkaa ja edetä suoraan analyysiin tai purkamisen jälkeen koodata aineisto ja tämän jälkeen siirtyä analyysiin. Kolmannessa vaihtoehdossa yhdistellään edellä esitetyt vaihtoehdot eli hyödynnetään sekä purkamista koodamisvaiheet. Tutkimuksen aineistonkeruumenetelmäksi on valittu teemahaastattelut tutkimuksen tavoitteiden pohjalta. Teemojen avulla jäseneltyä haastattelua hyödynnetään myös aineiston läpikäymisessä järjestelmällisesti. Aineiston koodaus voidaan jakaa kahteen eri lähestymistapaan. Ensimmäisenä esitellään aineistolähtöisyys, jossa lähdetään analysoimaan aineistoa ilman ennako-oletuksia. Toinen lähestymistapa ottaa huomioon myös teorian ja liittyy siihen tietoisesti mukaan jonkin näkökulman. (Eskola & Suoranta 1998)

Braun & Clarke (2006) esittelevät yksityiskohtaisesti teemoittelun eri vaiheet. Kuvaan 3 on koottu yhteenveto päävaiheista, joita teemoittelussa tulisi noudattaa.



Kuva 3: Teemoittelun vaiheet (mukailtu lähteestä Braun & Clarke 2006).

Kuten kuvasta 3 nähdään, ensimmäisenä tutkijan tulee tutustua aineistoon. Braun & Clarke (2006) huomauttavat, että mikäli kyseessä on sanallinen data, kuten haastattelut, tulee tiedot litteroida kirjalliseen muotoon. Tässä tutkimuksessa hyödynnettiin Teams-työkalua, jolla haastattelut nauhoitettiin sekä litteroitiin. Haastattelija teki myös omia muistiinpanoja haastatteluiden aikana. Aineiston tutustuminen tehtiin siis suurimmaksi osin Teamsin tekemän litteroinnin sekä omien muistiinpanojen pohjalta. Mikäli jokin kohta oli edellä mainituissa teksteissä epäselvää, tutkija palasi haastattelun nauhoitukseen ja kirjasi tiedon tätä kautta talteen. Tässä vaiheessa kerättiin mielenkiintoisia kommentteja ja huomautuksia. Seuraavana kuvassa 3 siirrytään vaiheeseen 2, jonka tarkoituksena on tuottaa koodaukset datasta, johon tutkija on tutustunut. Koodatut tiedot eroavat teemoista, jotka ovat laajempia kokonaisuuksia (Braun & Clarke 2006). Vaihe 3 alkaa, kun koodaus on valmis ja tiedot on lajiteltu. Tämä vaihe mahdollistaa analyysiin tekemisen laajempien teemojen tasolla. Tässä vaiheessa voidaan hyödyntää visuaalista tapaa kuten ajatuskarttoja, jotta löydetään oikeat teemat. Vaiheessa 4 jalostetaan vaiheessa 3 luotuja teemoja ja varmistetaan niiden toimivuudesta datan analysoinnissa. Huomioitavaa on, että teemojen välillä tulee olla tunnistettavat erot ja teeman sisällä olevien tietojen tulee olla yhteensopivia. Vaiheessa

5 syvennetään teemoittelua nimeämällä sekä määrittelemällä teemat. Viimeisessä vaiheessa jokaisesta teemasta kirjoitetaan analyysi, jonka tarkoituksena on vakuuttaa lukija aineiston pätevydestä. Tärkeää on, että kirjoitetaan johdonmukainen ja looginen kuvaus datasta teemojen sisällä ja niiden välillä. (Braun & Clarke 2006)

Teemoittelu valittiin tutkimuksen tavoitteiden sekä filosofian pohjalta, jota myös Eskola & Suoranta (1998) tukee: ”Teemoittelu on suositeltava analysointitapa jonkin käytännöllisen ongelman ratkaisemisessa.”. Kuten aiemmin mainittiin tutkimuksen aineistonkeruu, toteutettiin teemahaastattelulla, joten myös tämä vahvisti aineiston analyysimenetelmän valintaa. Teemoittelussa vertaillaan ja tutkitaan tiettyjen teemojen esiintymistä. Koska tutkimuksessa hyödynnetään abduktiivista päättelyä, koodaamisessa hyödynnetään sekä teoria- ja aineistolähtöisyyttä. Koska aineisto kerättiin teemahaastattelulla, hyödynnettiin jo tehtyä teemoittelua myös koodauksessa (Eskola & Suoranta 1998).

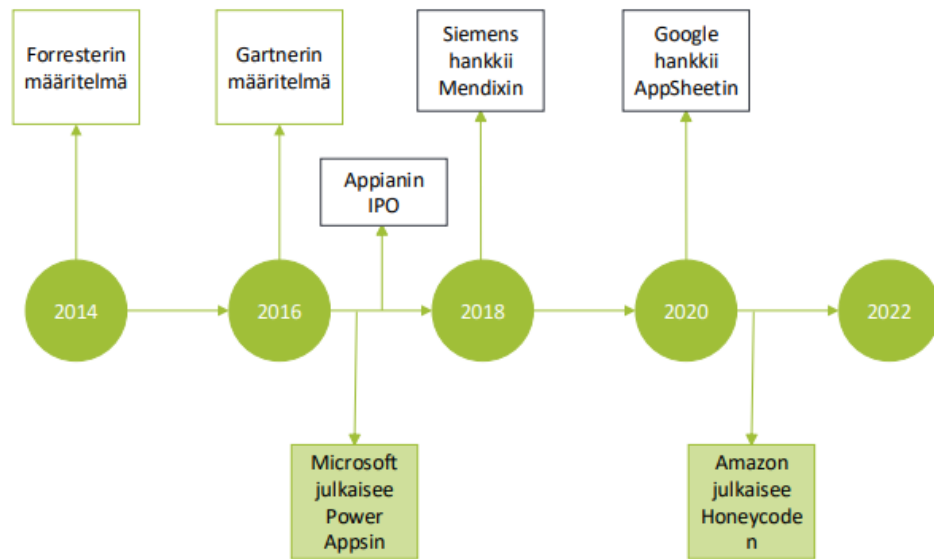
3. LOW-CODE -TEKNOLOGIA JA YHTEISTYÖKEHITTÄMINEN

Luvussa esitellään low-code -teknologian markkinoita, hyötyjä, haasteita sekä ominaisuuksia. Luvussa käsitellään myös yhteistyökehittämistä sekä sitä, mitä se tarkoittaa low-code -teknologian kontekstissa. Koska tutkimuksen tavoitteena on luoda yhteistyökehittämisen malli low-code -teknologialle, tutkitaan myös ketterää ohjelmistokehitystä ja sen roolia low-code -kehityksessä. Luvussa esitellään myös mitä tarkoitetaan arvon yhteisluonnilla. Koska sen nähdään olevan yhteydessä yhteistyökehittämiseen ja toisaalta palveluiden tuotteistamiseen, perustellaan sen esittely jo tässä luvussa olennaiseksi.

3.1 Low-code -teknologia

Vaikka low-code -teknologia on käsitteenä suhteellisen uusi, on kuitenkin tunnistettu, että nykyiset low-code -tuotteet ovat olleet olemassa jo historiassa, mutta eri muodossa. Tuotteita on uusittu, eikä niistä ole aiemmin käytetty termiä low-code. (Bock & Frank 2021) Viime vuosikymmenien aikana on tunnistettavissa eri trendejä, joiden tarkoituksena on ollut vähentää tehdyn koodin määrää esimerkkeinä 4GL 1980-luvulla ja Rapid Application development 1990-luvulla. (Di Ruscio et al. 2022) Kuten jo aiemmin mainittiin luvussa 1.1, käsite low-code on mainittu vuonna 2014 Forresterin tutkimuksen myötä ensimmäisen kerran (Sanchis et al. 2019; Di Ruscio et al. 2022). Käsitettä on kehitetty Forresterin toimesta, joten vuonna 2017 LCDP:t määriteltiin yksityiskohtaisemmin sovelluskehitykseen tarkoitetuiksi tuotteiksi ja/tai pilvipalveluiksi. Määritelmä otti kantaa myös siihen, että ohjelmoinnin sijaan hyödynnetään visuaalista, deklarativista tekniikkaa sekä alustat ovat saatavilla alhaiseen hintaan. (Di Ruscio et al. 2022)

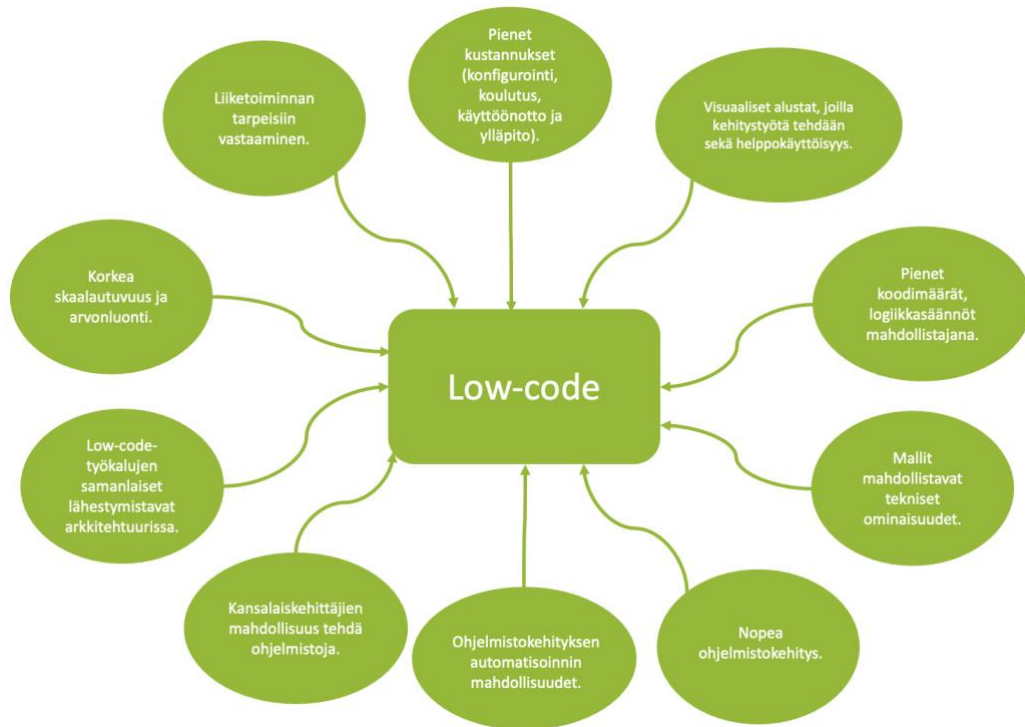
Kuvassa 4 esitellään Di Ruscio et al. (2022) tunnistama low-code -teknologian kehitys vuodesta 2014 alkaen. Kuvassa nähdään low-code -teknologian kehitysasteet vuoteen 2021 asti. Kuvan tarkoituksena on esitellä tämä tutkimuksen lukijalle low-code -teknologian nykyisen käsityksen historiaa ja auttaa ymmärtämään merkittäviä vaiheita, joita teknologian ympärillä on ollut.



Kuva 4: Merkittävät tapahtumat low-code -tekniikan historiassa (mukaanlaskien lähteestä Di Ruscio et al. 2022).

Kuten kuvasta 4 nähdään, 2010-luvun loppupuoli on ollut low-code -tekniikalle merkittävää aikaa. Useat suuret teknologiayritykset ovat investoineet ja panostaneet low-codeen, sillä sen merkitys liiketoiminnan ratkaisuihin on alkanut näkymään sekä sen potentiaalia on ymmärretty yhä enemmän. Kuten aikajana nähdään Microsoftin Power Apps julkaistiin marraskuussa 2016 (Di Ruscio et al. 2022; Phillips 2016). Myös muita isoja teknologia yhtiöitä on lähtenyt mukaan low-code -liiketoimintaan joko hankkimalla tai kehittämällä oman alustan. Kuvasta 4 nähdään esimerkkejä näistä yhtiöistä kuten Appian, Siemens, Google sekä Amazon. Kehityksen suuruudesta kertoo esimerkiksi Appianin vuonna 2017 alkanut listautumisanti, joka vuonna 2018 kasvoi markkina-arvoltaan lähes 2 miljardiin (Di Ruscio et al. 2022).

Pinho et al. (2022) mukaan kirjallisuudessa ei ole edelleenkään low-code -tekniikalle täysin yksikäsitteistä määritelmää. He esittävät huomion, jossa no-code katsotaan sisältävän osittain samoja päällekkäisiä piirteitä low-coden kanssa, mutta sen voidaan katsoa kuitenkin oleva eri työkalu. No-code -kehityksessä ei Pinho et al. (2022) määritelmän mukaan tarvitse kirjoittaa koodirivejä ollenkaan. Kirjallisuudesta havaitaan kuitenkin se, että termejä low-code sekä no-code määritellään osittain päällekkäin ja osittain erillään. Esimerkkinä Bhattacharyya & Kumar (2021) tekemä tutkimus aiheen ympäriltä, jossa termejä käytetään pitkälti synonyymeinä. Low-coden ympärille on kuitenkin muodostunut selkeitä ominaisuuksia, jonka avulla teknologia on nykypäivänä tunnistettavissa. Kuvassa 5 esitellään tunnistettuja ominaisuuksia, jotka ovat usein havaittavissa, kun puhutaan low-code -tekniikasta.



Kuva 5: Low-code -teknologian ominaisuudet. (Pinho et al. 2022; Rosa-Bilbao et al. 2023; Sanchis et al. 2019)

Kuten kuvasta 5 nähdään, low-code -teknologian ympärille on muodostunut useita selkeitä ominaisuuksia, joista se voidaan tunnistaa. Nämä ominaisuudet tuovat yrityksille uusia mahdollisuuksia luoda sovelluksia sekä automatisoida prosesseja. Nopea ohjelmistokehitys mahdollistaa digitalisaation osaksi liiketoimintaa yhä nopeammin, kun samalla kustannukset ovat pienemmät. Ominaisuuksista nähdään, että low-code -teknologian ympäriltä voidaan tunnistaa paljon hyötyjä, miksi sitä tulisi käyttää ja hyödyntää. Vincent et al. (2021) esittelee kuitenkin tekijöitä, jotka kannattaa huomioida low-code -teknologiaan investoitaessa, jotta ymmärretään, onko sen hyödyntäminen paras vaihtoehto. Ensimmäisenä tulisi pohtia liiketoimintakriteereitä. Esimerkkinä ovat mihin alustaa käytetään, onko toimittaja jo yrityksen toisen järjestelmän toimittaja, kustannukset sekä hinnoittelumalli. Muita tekijöitä ovat teknologia, kehittäminen sekä ekosysteemi. Nämä ottavat huomioon tekniset asiat, kuten arkkitehtuurin sekä tukimallit ja kumppaneiden vaatimuksen. (Vincent et al. 2021) Nämä ovat tärkeitä tekijöitä arvioitavaksi niin palveluntarjoajalle kuin asiakkaille, jotta ratkaisujen elinikä saadaan mahdollisimman pitkäksi eikä vastaan tule tilannetta, jossa tuotetta ei voitaisi enää käyttää, ylläpitää tai kehittää.

Low-code -teknologian ympärillä on tunnistettu haasteita, jotka on hyvä huomioida myös palvelun tuotteistamisen näkökulmasta. Martinez & Pfister (2023) mukaan, low-code -alustat tarjoavat tietyt mukautusvaihtoehdot eli räätälöinnin puute voi muodostua

haasteeksi. Skaalautuvuudessa tunnistetaan myös haasteita, mikäli esimerkiksi sovellusta halutaan laajentaa suurempiin projekteihin. Haasteena nähdään myös hajanaisuus, joka liittyy yhteen toimivuuteen eri tietokantojen kanssa sekä sitoutuminen low-code -ratkaisun toimittajaan. Low-code -teknologian ympärillä tunnistettavaa on myös lisenssikustannukset, jolloin kustannukset eivät synny kehityksen alussa vaan käyttöönoton jälkeen. Turvallisuus on myös huomioitava tekijä, sillä käyttäjät voivat matalalla kynnyksellä kehittää sovelluksia ymmärtämättä tietoturvallisuuden näkökulmaa. (Martinez & Pfister 2023)

Low-code -markkinoiden voidaan nähdä kasvavan usealla eri toimialalla ja täten myös palveluntarjoajien määrän voidaan olettaa kasvavan. Jain et al. (2021) arvioi, että vuoteen 2025 mennessä 70 % yritysten kehittämistä uusista sovelluksista käyttää low-code tai no-code -teknologiaa. Kun verrataan vuoteen 2020 osuus oli alle 25 % (Jain et al. 2021). Cabot (2020) mukaan low-code -teknologian suosiolle on useita eri syitä: koodaamista tarvitsee tehdä vähemmän, sovellusskenaariot sekä mallinnus ovat selkeämpiä ja low-code -työkalut ovat mukavia sekä helppokäyttöisiä. Low-code -teknologian suosion myötä low-code -alustoja on tarjolla usealla eri ohjelmistotoimittajalla. Vincent et al. (2022) esittelee globaaleja toimittajia ja luokittelee niitä eri sektoreihin (kuva 6). Tutkimus on toteutettu internet-kyselynä helmi- ja maaliskuussa vuonna 2022. Toimittajat luokitellaan markkinoiden johtajiksi, visionääreiksi, haastajiksi sekä kapean sektorin toimijoiksi.



Kuva 6: Low-code -toimittajat (Vincent et al. 2022).

Neljän sektorin lisäksi kuvasta 6 nähdään nuolten osoittavan eri suuntiin kyvykkyyksien kasvun mukaisesti. Suorituskyvyn painotukset kasvavat y-suunnassa ja innovaatiokriteerit sekä tarjonta/tuotestrategia painotukset kasvavat x-suunnassa. Markkinajohtajat erottuvat globaaleilla markkinoilla ja osoittavat vahvaa toteutusta ja visiota. Johtavista toimittajista Microsoft, Salesforce sekä ServiceNow toimivat Saas-alustoilla. Haastajat eli Oracle sijoitetaan korkealle suorituskyvyn ja toteutuksen kannalta, mutta nähdään puutteita vielä tietokantapainotteisen alustan työnkulkuominaisuuksissa. Kolmanneksi sektoriksi kuvaan on määritelty kapean sektorin toimittajat, kuten Retool, Huawei ja Quickbase. Tälle ryhmälle ominaista on, että keskitytään tiettyyn markkina-alueeseen. Esitetyt toimittajat voidaan kuitenkin nähdä omassa sektorissaan merkittävinä, koska heidät on sisällytetty tutkimukseen. Neljäs sektori on visionäärit, joilla havaitaan puutteita suorituskyvyn painotuksessa. Toimittajat, jotka esitellään neljännessä sektorissa ovat Applan, Zoho sekä Pegasystems. Kuvasta 6 havaitaan, että low-code -teknologian toimittajia on merkittävästi ja useat isot ohjelmistotoimittajat ovat lähteneet kilpailuun mukaan. (Vincent et al. 2022) Voidaan olettaa, että tulevaisuudessa kilpailu kasvaa entisestään ja tästä syystä yritysten panostaminen low-code -kehityksen prosesseihin ja palveluihin voidaan pitää perusteltuina.

3.2 Yhteistyökehittäminen low-code -teknologialla

Yhteisluomista voidaan pitää organisaatioiden välisenä yhteistyön muotona (Bharosa et al. 2020). Yhteisluominen on asiakkaan ja tuottajan välistä yhteistyötä, jossa vuorovaikutteisesti parannetaan tuotetta tai palvelua (Ajmal et al. 2023). Eri sidosryhmät osallistuvat aktiivisesti innovaatioprosesseihin ja arvon luomiseen. Yhteisluomisessa tärkeää on, että jokainen edustaja tiimissä jakaa ideoitaan ja resursseja yhdistetään. (Bharosa et al. 2020) Vargo & Lusch (2004) esittelevät palvelukeskeisen näkemyksen vuorovaikutusprosessin, jossa tärkeää on määritellä asiakaskohtaisesti ratkaisu tarpeeseen. Prosessissa tarkoituksena on vuorovaikutteisesti määritellä, kehittää ja toimittaa asiakkaalle räätälöity ratkaisu, joka voi olla palvelu, tuote tai näiden yhdistelmä. Palvelukeskeisessä näkökulmassa erityisen tärkeä ajatusmalli on se, että tehdään yhdessä asiakkaan kanssa eikä ainoastaan asiakkaan puolesta. Tutkimus korostaa yhteisluomista, jossa tulisi pyrkiä maksimoimaan asiakkaan osallistuminen räätälöintiin. Tällä voidaan varmistaa, että lopputuote vastaa tarpeita. (Vargo & Lusch 2004) Kestävä arvonluontiprosessi, vuorovaikutteisuus sekä yhdistettävyyys nähdään kirjallisuudessa erittäin tärkeänä. Yhteisluontiprosessi nähdään kuitenkin olevan paljon monimutkaisempaa kuin vai tuotteiden ja palveluiden kehittäminen yhdessä.

Yhteistyökehittämistä tekevät yritykset voivat vaihtaa tietoa ja osaamista, joka hyödyttää molempia osapuolia. Erityisesti korostetaan yhteiskehittämisen ainutlaatuisia keinoja luoda lisäarvoa innovaatioista. (Hempe et al. 2015)

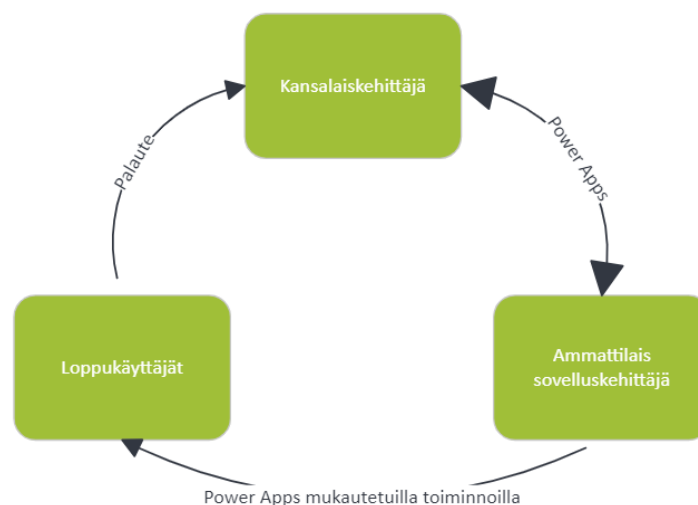
Iho et al. (2021) mukaan ”yritys nähdään rakenteena, jonka päätavoitteena on tuoda yhteen yksilöitä, joilla on erityisosaamista ja tarjota heille kokoonpano, jossa he voivat soveltaa ja integroida tätä tietoa tuottavalla tavalla.”. Yhteistyökehittämisen näkökulmasta tulee miettiä työntekijöiden roolit, niin asiakkaan kuin ohjelmiston toimittajan puolelta. Kun tehdään kahden tai useamman organisaation välistä yhteistyötä roolien ja vastuiden merkitys korostuu, jotta työskentely on tehokasta ja jokainen ymmärtää oman vastuun alueensa. Pinho et al. (2022) mainitsevat yhdeksi vaihtoehdoksi työskentelytavan low-codelle, jossa loppukäyttäjähittäjät työstävät ratkaisusta suurimman osan siihen asti, kunnes tarvitaan koodaamista. Heidän mukaansa osat, joihin tarvitaan koodaamista voisi tehdä ammattilaisohjelmistokehittäjät. Low-code -alustat mahdollistavat reaaliaikaisen muokkauksen sen visuaalisuuden ja helpon käytettävyyden myötä. Jo kehitysvaiheessa on mahdollista ottaa mukaan henkilöitä eri liiketoiminnan osa-alueilta. Muutoksista voidaan keskustella ja niistä voidaan päättää yhdessä. Low-code -kehitys mahdollistaa myös reaaliaikaisen kehittämisen esimerkiksi työpajoissa, joissa asiakkaalta voidaan saada palautetta välittömästi. Tämä mahdollistaa nopeamman kehityksen, kuin perinteinen ohjelmistokehitys. Innovaatiojohtamisen alalla on tunnistettu, että arvonluonti digitaalisilla teknologioilla vaatii vuorovaikutusta eri sidosryhmien välillä sekä osaamista eri alueilta. Artikkelissa korostetaan, että uusien lisäarvoa tuottavien ohjelmistojen kohdalla vaaditaan syvällistä liiketoiminta- sekä IT-osaamista. Haasteena tässä tunnistetaan liiketoiminta- ja IT-yksiköiden välinen kuilu, niin kulttuurin kuin työskentelytapojen suhteen. (Iho et al. 2021) Huomioitavaa on, että yhteisluominen ei ole toimintojen siirtämistä ja ulkoistamista asiakkaille. Tällaisella toimintamallilla harvoin saavutetaan tyytyväistä asiakasta. (Prahalad & Ramaswamy 2004)

3.3 Case Microsoft: Fuusiokehittäminen ja Power Apps

Tarkastellaan seuraavaksi yhteistyökehittämistä yhden suuren low-code -ratkaisuja mahdollistavan ohjelmistotoimittajan ja -tekijän näkökulmasta eli Microsoftin. Microsoft on julkaissut e-kirjan, joka käsittelee fuusiokehittämistä sovelluskehityksen yhteydessä hyödyntäen heidän omaa low-code -alustansa Power Platformia (Boyer et al. 2021). Kirjassa tarkastellaan erityisesti sovelluskehitykseen tarkoitettuja ratkaisuja, joita voidaan tehdä Microsoftin Power Appsilla. Boyer et al. (2021) kuvailee, että Power Apps -ratkaisuilla voidaan ratkaista monia tarpeita eri liiketoimintaprosesseihin, digitalisoida ja

kehittää niitä. Lähtökohtana voidaan pitää sitä, että kansalaiskehittäjät eli yrityksen liiketoiminnan keskiössä työskentelevät työntekijät tekevät sovellusratkaisut. Iho et al. (2021) määrittelee kansalaiskehittäjät LCDP-kehittäjiksi (low-code development platform). Power Apps mahdollistaa tämän esikonfiguroidulla alustalla, jossa hyödynnetään valmiita tietoliittimiä eri tietolähteiden avulla. Huomattavaa on kuitenkin, että on olemassa tarpeita, joihin ei ole olemassa valmiita liittämiä tiedon yhdistämiseen ja välittämiseen, jolloin tarvitaan ammattikehittäjien apua sovelluksen tekemiseen. Kansalais- ja ammattikehittäjät työskentelevät yhdessä jakaen vastuun low-code -kehityksen sekä tietojen liittämisen väliltä. Tällöin pystytään luomaan yhä monimutkaisempia liiketoimintaratkaisuja.

Fuusiokehittäminen (diplomityön kontekstissa yhteistyökehittäminen) kokoaa asiantuntijoita eri liiketoiminnan osa-alueilta yhteen. Ammattilaiskehittäjät tukevat kansalaiskehittäjiä tarpeen mukaan, ja loppukäyttäjät voivat antaa palautetta toimintojen ja puutteiden näkökulmista. Boyer et al. (2021) kuvailee prosessin olevan hyvin iteratiivinen ja ketterä, jolloin saadaan tuotettua nopeasti ratkaisuja liiketoiminnan tarpeisiin. Kuvaan 7 on kuvattu e-kirjassa esitelty malli fuusiokehittämiselle.



Kuva 7: Fuusiokehittämisen malli (mukailtu lähteestä Boyer et al. 2021).

Kuten kuvasta 7 nähdään, prosessi jakautuu kolmen eri tahon välille: kansalaiskehittäjille, ammattilaiskehittäjille sekä loppukäyttäjille. Nuolet osoittavat, että yhteistyötä tehdään hyvinkin tiiviisti eri ryhmien välillä. Kansalaiskehittäjä voidaan nähdä olevan hyvin keskeisessä roolissa loppukäyttäjien sekä ammattilaiskehittäjien näkökulmasta. Huomioitavaa on myös se, että kuva osoittaa kansalaiskehittäjien ja ammattilaiskehittäjien tiiviin yhteistyön ja kehittämisen merkityksen. Kuvan 7 mukaan

fuusiokehittämisen malli korostaa erityisesti vuorovaikutteisuutta eri sidosryhmien välillä sekä prosessin iteratiivisuutta. Boyer et al. (2021) mukaan fuusiokehittäminen on siis lähestymistapa luoda nopeasti uusia ratkaisuja liiketoiminnan tarpeisiin, arvostaen jokaisen tiimin jäsenen taitoja ja kykyjä. Voidaan katsoa, että yhteisvaikutuksella mahdollistetaan tehokasta kommunikointia tiimin välillä sekä iterointi on nopeaa. (Boyer et al. 2021)

3.4 Ketterän sovelluskehityksen piirteet

Perinteisissä ohjelmistokehitysmenetelmissä kuten koodi ja korjaus -menetelmä (code-and-fix), vesiputousmallissa, muunnos- sekä spiraalimenetelmässä on havaittu rajoitteita kuten kyky sopeutua projektin muuttuviin olosuhteisiin sekä menetelmien työläisyys. Näiden rajoitteiden pohjalta syntyi ketterä filosofia, jonka tarkoituksena on vastata nykypäivän organisaatioiden tarpeisiin. Ketterä ohjelmistokehitys alkoi prosessien ja dokumentoinnin vähentämisen, tuotteiden nopean kehityksen ja asiakkaiden osallistamisen kautta. Vuonna 2001 syntyi ketterä ohjelmistokehitysfilosofia ohjelmistoalan ammattilaisten toimesta. He loivat manifestin, jossa ilmaistaan arvot ja periaatteet, joita ketterässä ohjelmistokehityksessä tulee noudattaa. Taulukkoon 4 on koottu manifestin 12 periaatetta. (Misra et al. 2012)

Taulukko 4: Ketterän ohjelmistokehityksen manifesti (Misra et al. 2012).

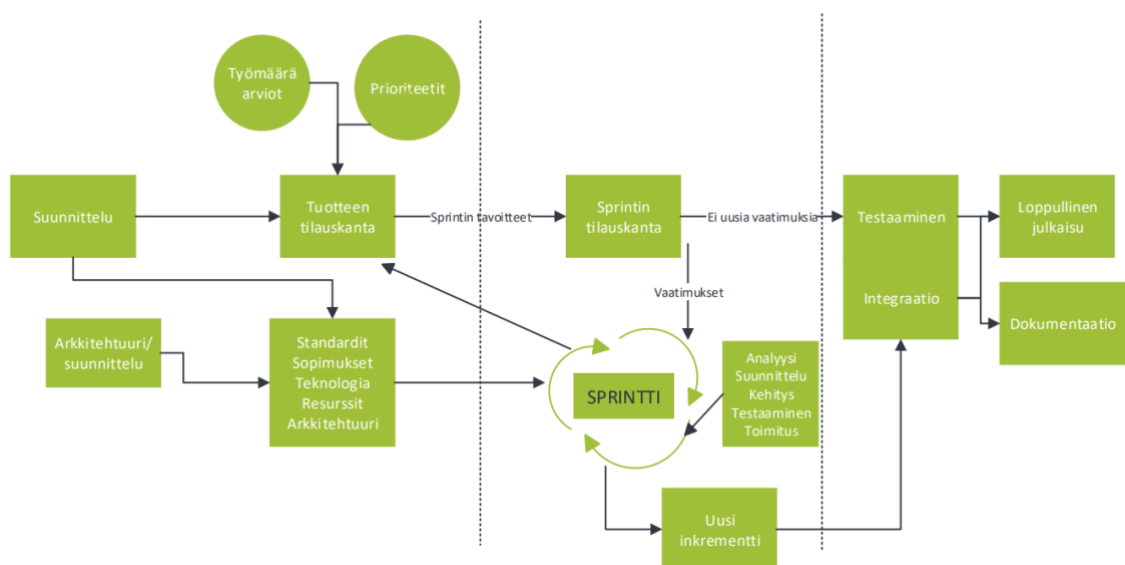
1	Ohjelmiston varhainen ja jatkuva toimitus, jolla tavoitteena tyydyttää asiakkaan tarpeet.	7	Edistymisen mittaaminen tapahtuu toimivan ohjelmiston kautta.
2	Muuttuvien vaatimusten hyväksyminen. Asiakkaan kilpailuetua pyritään saavuttamaan muutosten kautta ketterillä menetelmillä.	8	Kestävä kehitys vahvistuu ketterien prosessien kautta, tasaisen työtahdin ylläpito.
3	Toimivien ohjelmistojen toimittaminen usein, aikajakso viikoista kuukausiin.	9	Huomioidaan tekninen huippuosaaminen ja hyvä suunnittelu, jolloin ketteryys lisääntyy.
4	Liiketoiminnan työntekijät sekä ohjelmistokehittäjät työskentelevät yhteistyössä päivittäin koko projektin ajan.	10	Korostetaan yksinkertaisuutta. Olennaista on, että tehdään vain tärkeät asiat.
5	Motivoituneet työntekijät, joille mahdollistetaan ympäristö ja tuki työn suorittamiseen.	11	Itseorganisoidut ryhmät ovat arkkitehtuurin, vaatimusten ja mallien syntymisen keskiössä.
6	Tiedonvälitys kehitystiimille tehokkainta on kasvokkaisen keskustelun kautta.	12	Menestys syntyy, kun tiimi pohtii säännöllisesti tehokkuutta ja sopeuttaa toimintaa sen mukaisesti.

Kuten taulukosta 4 nähdään, ketterän ohjelmistokehityksen ympärille on kirjattu selkeät tekijät, johon toiminta perustuu. Erityisesti korostetaan tiimilähtöisyyttä, kehittämistä yhteistyössä sekä onnistumisen ja menestymisen edellytyksiä.

Ketterien menetelmien ansiosta yksinkertaisuus ja nopeus ovat mahdollisia. Ketterän työryhmän tavoitteena on keskittyä vain toimintoihin, joita ratkaisussa tarvitaan ensikädessä. Tavoitteena on myös toimittaa nämä nopeasti, jonka jälkeen palautteen myötä ratkaisuun kehitetään uusia toiminnallisuuksia, jotka parantavat ratkaisua.

Erityisesti internet-ohjelmistoteollisuus sekä mobiilisovellukset kaipaavat nopeita ja ketteriä ohjelmistokehitysprosesseja. Ketterä liike painottaa ohjelmistokehittäjien yhteisöllisyyttä sekä ihmisen roolia. Ketteryydessä tärkeää on myös tiimityöskentely sekä sitä vahvistavat toimenpiteet. (Abrahamsson et al. 2017) Tiimityöskentelyssä voidaan korostaa autonomiaa, itsenäisyyttä ja itseorganisoituvuutta, jotka mahdollistavat suuren ongelmaratkaisunopeuden sekä tehokkuuden (Rathor et al. 2023). Tavoitteena on kehittää jatkuvasti testattuja ohjelmistoja, jotka ovat toimintavarmoja. Julkaisutahti on säännöllinen, jossa korostetaan yksinkertaista ja suoraviivaista koodia, jolloin dokumentointi ei tuota liiallista työtä. Ketteryydessä korostetaan myös kehittäjien ja asiakkaiden välistä yhteistyötä. (Abrahamsson et al. 2017) Rathor et al. (2023) korostavat viestinnän merkitystä, jolla tarkoitetaan yhteisymmärrystä projektin laajuudesta, tehtävistä, toiminnoista, virstanpylväistä ja tavoitteista. Ketterän kehityksen tavoitteena on tuottaa liiketoiminta-arvoa heti projektin alusta alkaen. Huomioitavaa on mahdolliset muutokset niin ratkaisuun kuin sopimukseen prosessin aikana, jolloin myös sopimustekniset asiat tulee olla ketterästi toteutettu. (Abrahamsson et al. 2017)

Ketteriä menetelmiä on useita, joista yksi paljon käytetty on Scrum. Abrahamsson et al. (2017) mukaan Scrum on Japanista rantautunut nopea ja adaptiivinen tuotekehitysprosessi, johon kirjallisuudessa viitataan vuodesta 1986 lähtien. Scrum ottaa kantaa erityisesti siihen, miten tiimin pitää toimia muuttuvassa ympäristössä, jotta järjestelmää voidaan kehittää jatkuvasti. Perusoletuksena on, että kehitysvaiheeseen liittyy useita muuttujia, jotka muuttuvat prosessin aikana. Kuvassa 8 esitellään Scrumin perusidea prosessimuodossa.



Kuva 8: Scrum prosessimalli (mukailtu lähteestä Abrahamsson et al. 2017).

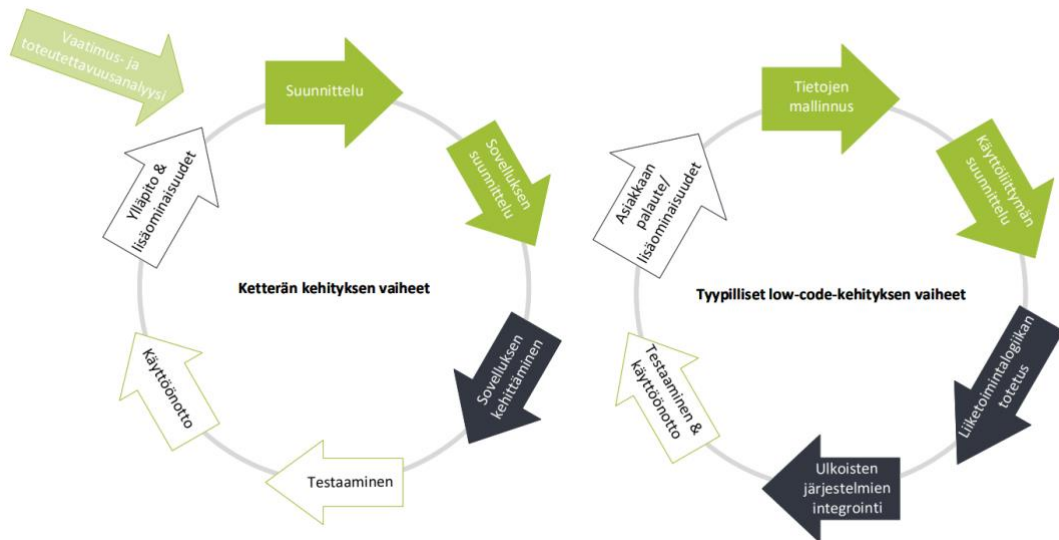
Ensimmäinen vaihe sisältää suunnittelun, jossa ratkaisu määritellään. Vaatimukset priorisoidaan ja niiden toteuttamiseen vaadittava panostus arvioidaan. Arkkitehtuurivaiheessa suunnitellaan ratkaisulle arkkitehtuuri sen hetkisen määrittelyn tiedoilla. Kehitysvaiheessa toteutetaan ketterää toimintaa. Perusideana on se, että odotetaan odottamatonta. Tällä tarkoitetaan, että tunnistetut muuttujat kuten aikakehys tai vaatimukset voivat muuttua prosessin aikana. Kehitysvaiheessa järjestelmää kehitetään sprinteissä, jotka ovat iteratiivisia syklejä, joissa ratkaisua kehitetään. Jokainen sprintti sisältää ohjelmistokehityksen vaiheet eli vaatimus-, analyysi-, suunnittelu-, kehitys- ja toimitusvaiheet. Sprintin aikana arkkitehtuuri on myös kehittyvässä roolissa. Tavallisesti yksi sprintti kestää viikosta aina yhteen kuukauteen. (Abrahamsso et al. 2017) Erytisen tärkeä ajatusmaailma on se, että sprintin toimituspäivä ei muutu, mutta puolestaan toimitettavia toimintoja voidaan vähentää, mikäli niitä ei saada määräaikaan valmiiksi (Rising & Janoff 2000). Viimeinen vaihe sisältää ratkaisun julkaisun. Tämä tarkoittaa, että ongelmia ei enää ole sekä muuttujat eivät ole enää relevantteja ratkaisun kannalta. Ratkaisun julkaisussa tulee huomioida mm. testaus ja dokumentointi. (Abrahamsso et al. 2017) Huomattavaa on siis se, että sprintin perusideana on tarjota arvokkaita toimintoja.

Scrum menetelmä sopii käytettäväksi erityisesti pienissä tiimeissä (Abrahamsso et al. 2017; Rising & Janoff 2000). Pienten tiimien hyödyt ovat niiden itsenäisessä työskentelyssä sekä tehokkuudessa. Tiimiin kuuluu yksi Scrum Master, yksi tuoteomistaja ja kehittäjä(t). Huomattavaa on myös se, että Scrum-tiimit ovat itseohjautuvia eikä niissä ole hierarkiaa. Valinnan vapaus on siis merkittävä osa työskentelytapaa Scrum-tiimeissä. (Schwaber & Sutherland 2020)

Low-code -teknologia voidaan nähdä ketteränä sen joustavuuden, nopeuden, käyttöönoton ja ylläpidon helppouden näkökulmista. Low-code -teknologian suurin motivaatio sovelluskehityksessä on käyttäjien palaute ja nopea muutosten tekeminen. Tutkimuksessa määritellään kaksi eri tapaa kehittää low-code -sovellus:

- 1) "UI to Data Design", jossa ensin luodaan käyttöliittymä ja sen jälkeen yhdistetään tarvittaviin tietolähteisiin.
- 2) "Data to UI", jossa ensin mietitään tietomallit ja sen jälkeen sovelluksen käyttöliittymä.

Molemmissa tapauksissa hyödynnetään API-rajapintoja, kun sovelluslogiikka on toteutettu, jotta integrointi saadaan toteutettua. Kuvassa 9 vertaillaan perinteisiä ketterän ohjelmistokehityksen ja low-code -kehityksen prosesseja. (Al Alamin et al. 2021).



Kuva 9: Ketterä kehitys vs. low-code -kehitys (mukailtu lähteestä Al Alamin et al. 2021).

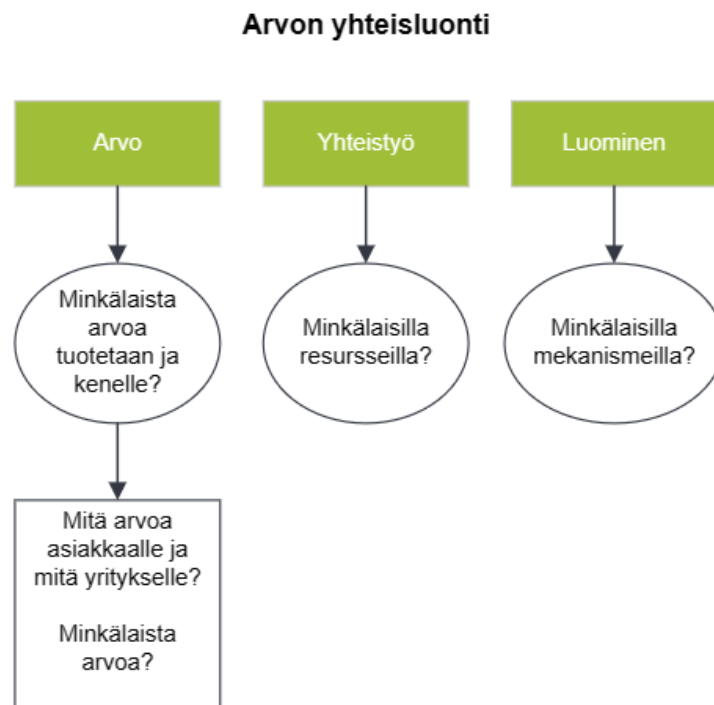
Kuvassa 9 vasemmalla puolella nähdään sykli ketterän kehityksen vaiheista ja kuvan oikealla puolella sykli low-code -kehityksen vaiheista. Kuvasta nähdään, että prosesseissa on selkeitä yhtäläisyyksiä. Low-code -kehityksessä on nähtävissä, että testaaminen ja käyttöönotto ovat yksittäinen vaihe prosessissa, kun perinteisessä ketterässä ohjelmistokehityksessä vaiheet on jaettu erilleen. Toisaalta sovelluksen kehitys on ketterän kehityksen mallissa yksittäinen, kun low-code -kehityksessä se on jaettu kahteen vaiheeseen. Prosessien alku- ja loppuvaiheet koostuvat hyvin samanlaisista tekijöistä: suunnittelu, ylläpito, palaute ja lisäominaisuudet.

3.5 Arvon yhteisluonti

Arvolle ja arvonluonnille ei ole löydetty kirjallisuudessa yhtä selkeää määritelmää. Voidaan kuitenkin todeta, että arvonluonti on erittäin tärkeää yrityksen menestymisen kannalta. (Lindman et al. 2016) Nähdään, että arvonluonti on mahdollista ainoastaan, jos tavaraa tai palvelua kulutetaan (Vargo & Lusch 2004). Eri tutkimuksissa arvoa ja arvonluontia määritellään eri tavoin, mutta Lindman et al. (2016) mukaan asiakkaat määrittävät luodaanko arvoa vai ei. Arvon määrittelyssä viitataan kokemaan, kun taas arvon luominen viittaa siihen, mitä tuotteilla tai palveluilla tehdään, jotta niistä saadaan arvoa (Sahhar & Loohuis 2022). Vargo & Lusch (2004) tukevat palvelukeskeisellä näkemyksellä ajatusta siitä, että asiakkaat ovat arvon luonnin keskiössä. Heidän mukaansa asiakkaille syntyy arvoa koko suhteen ajan, mikäli ymmärretään, että arvo on muunnettavissa yhteistuotannon kautta. Arvonluonnissa on tapahtunut selkeä muutos

historian saatossa. Esimerkiksi taloustieteessä ajateltiin, että arvo on tuotelähtöistä ja palvelut ovat tuotteiden sekä markkinoinnin apuvälineitä. (Vargo & Lusch 2004) Arvonluonti ja sen merkityksen ymmärtäminen nähdään tärkeänä myös sen vuoksi, että arvonluonti lisää yrityksen kilpailuetua. Kirjallisuudessa on tunnistettu myös se, että ostopäätökseen vaikuttaa enemmän kuin ydintuote, esimerkkinä vuorovaikutuksen laatu, jolla nähdään olevan täydentävä vaikutus asiakkaan kokeman arvon näkökulmasta. (Lindman et al. 2016)

Tarkastellaan arvon yhteisluontia käsitteenä sen selkeyttämiseksi. Saarijärvi et al. (2013) mukaan myös arvon yhteisluonti käsitteenä on hyvin monimuotoinen. Kuvassa 10 esitetään arvon yhteisluonnin ympärille asetettuja kysymyksiä, joiden avulla voidaan selkeyttää arvon yhteisluonnin merkitystä.



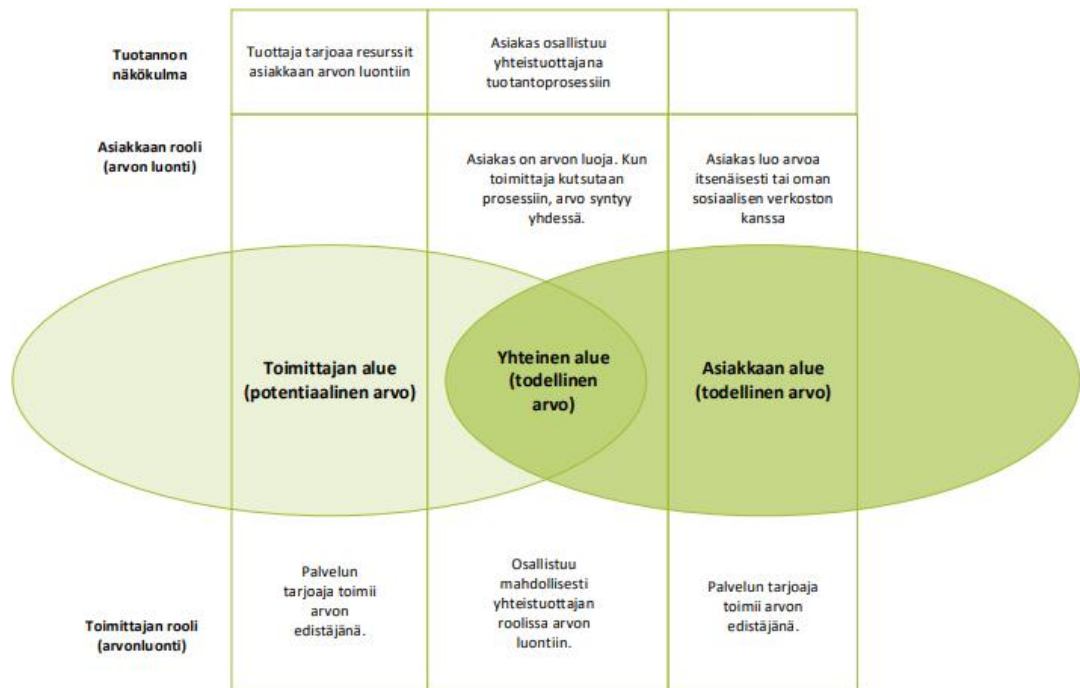
Kuva 10: Arvon yhteisluonti (mukailtu lähteestä Saarijärvi et al. 2013).

Kuvasta 10 nähdään, että arvon yhteisluontia pohtiessa olisi hyvä miettiä kenelle ja miten arvoa halutaan luoda sekä minkälaisilla työkaluilla sitä on mahdollista tehdä. Arvon yhteisluonti syntyy useasta eri osasta, osallistumisen ja sitoutumisen kautta. Eri toimijoiden tulee osallistua toimintaan, jotta arvoa syntyy vastavuoroiseen hyödyn tuottamiseen (Bonamingo et al. 2022). Bonamingo et al. (2022) nostaa sitoutumisen sekä luottamuksen avaintekijöiksi arvon yhteisluonnille. Arvon yhteisluonti vahvistaa asiakasuskollisuutta, luo positiivista asennetta sekä parantaa asiakastyytyvää. Positiivinen asenne sekä innovatiivinen ympäristö ovat kannustimia, jotka edistävät

halua arvon yhteisluontiin. (Ajmal et al. 2023) Yhteistyökehittämisen mallin näkökulmasta on tärkeää ymmärtää arvon yhteisluonti, jotta eri sidosryhmät saadaan motivoitua ja sitoutettua prosesseihin.

Ajmal et al. (2023) ovat tunnistaneeet motivaattoreita arvonyhteisluonnin kontekstissa. Heidän mukaansa asiakkaan näkökulmasta tärkein kannustin, joka motivoi heitä osallistumaan arvon yhteisluontiprosessiin ovat rahalliset palkkiot, sosiaaliset tekijät sekä asiakkaiden henkilökohtainen kokemus. Arvon yhteisluonnille nähdään myös selkeitä haasteita. Arvon yhteisluontiprosessit eivät ole virheettömiä, jonka seurauksena voi muodostua arvon yhteistuhoa. Se voi tapahtua, joko vahingossa tai tahallisesti (Sahhar & Loohuis 2022). Bonamingo et al. (2022) mukaan arvon yhteisluomista voi estää toimijoiden välinen yhteensopimattomuus, arvon yhteisluonnin kokema sekä tiedon puute. Wu et al. (2022) mukaan palvelutyöntekijän fysiologiset sekä psykologiset tekijät ja niiden hallitseminen, kuten työstä johtuva stressi voi johtaa arvon yhteistuhon, koska tällä voi olla vaikutusta vuorovaikutuksen laatuun. Yhteistyökehittämisen näkökulmasta arvonyhteistuhon tekijät ovat tärkeää tunnistaa, jotta sitä voidaan välttää ja hallita.

Tarkastellaan seuraavaksi arvon yhteisluontia Grönroos-Voima -mallin määritelmän mukaan. Arvoon johtava prosessi jakautuu arvoalueisiin, joissa on eri tavoitteet asiakkaan sekä toimittajan näkökulmista. Mallissa määritellään myös tuotannon näkökulma arvon yhteisluonnille. Grönroos (2017) korostaa, että arvoprosessi voi alkaa mistä tahansa arvoalueesta, eikä prosessi välttämättä ole lineaarinen. Hän toteaa myös sen, että malli antaa mahdollisuuden arvoprosessin iteratiivisuudelle. Kirjallisuudessa on tunnistettu, että asiakkaiden prosessin ymmärtämistä ei voida ennen tai jälkeen ostotapahtuman vähätellä. Arvoprosessi voidaan nähdä matkana, jossa arvovaikutusta voi olla jo paljon ennen fyysistä kontaktia palveluntarjoajan ja asiakkaan välillä. (Grönroos 2017) Kuvassa 11 esitellään Grönroos-Voima -malli.



Kuva 11: Grönroos-Voima -malli arvon yhteisluonnille (mukailtu lähteestä Grönroos 2017).

Tarkastellaan kuvasta 11 ensin toimittajan ja asiakkaan omia alueita. Toimittajan alueella tehdään toimenpiteitä, jotta arvonluonti on mahdollista. Tällä alueella arvo kuvaillaan potentiaalisesti arvoksi, koska todellisen arvon luo asiakas. Resurssien, prosessien ja tarjonnan kautta toimittajalla on mahdollista luoda käyttöarvoa asiakkaalle kulutuksen yhteydessä. Kuten kuva 11 esittää, asiakkaan alueella luodaan todellinen arvo. Sosiaalisen verkoston myötä voidaan vaikuttaa asiakkaan arvoprosessiin. Palveluntarjoajan toinen asiakas voi vahvistaa toisen asiakkaan arvoprosessin vertaisvuorovaikutuksessa. Yhteisellä alueella asiakas käyttää toimittajan tarjoamia, jolloin arvoa kehittyy asiakkaalle. Yhteistuotannon kautta voidaan tuottaa todellista arvoa. Arvoa syntyy siis sekä asiakkaan alueella, että yhteisellä alueella. Voidaan kuitenkin nähdä, että alueet sisältävät erilaisia prosesseja. (Grönroos 2017)

4. PALVELUIDEN TUOTTEISTAMINEN

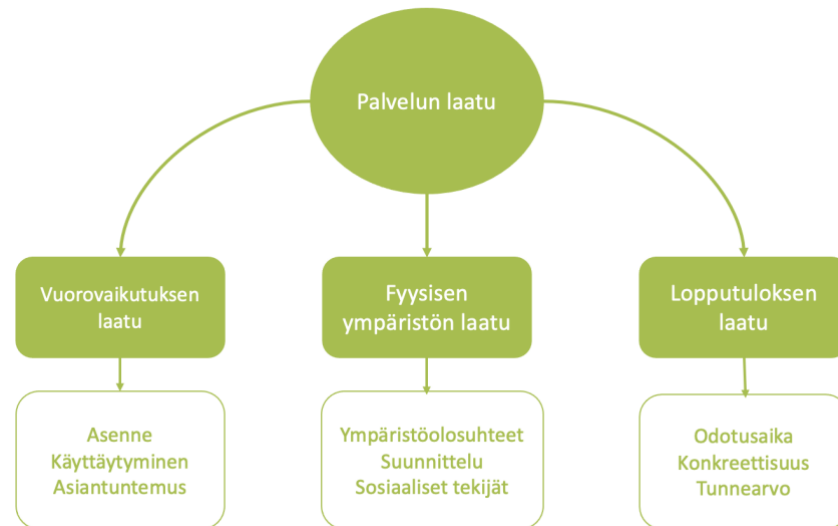
Tässä luvussa käsitellään palveluajattelua ja palveluiden tuotteistamista. Luvussa esitellään myös tuotteistamisprosesseja, joita palveluiden tuotteistamisessa voidaan hyödyntää.

4.1 Palveluajattelu

Grönroos (2001) mukaan palveluyrityksellä ei ole tuotteita vaan interaktiivisia prosesseja. Hän korostaa erityisesti prosessinäkökulmaa, jossa huomautetaan, että palvelut eivät ole asioita. Laadukkaan palvelun ja asiakkaan odotusten täyttäminen johtaa tyytyväisiin asiakkaisiin, jolloin on todennäköisempää, että asiakas ostaa toistuvasti tarjoamaa sekä suosittelee sitä myös muille. Asiakaslähtöisessä näkökulmassa nähdään, että kaikki tuotteet ovat palvelupaketteja, joiden avulla arvoa luodaan. (Russell-Bennett et al. 2013) Eiglier et al. (1997) määrittelevät 4 ominaisuutta, jotka on tunnistettu palveluille: vaihtuvuus, aineettomuus, erottamattomuus sekä pilaantuvuus. Vaihtuvuus viittaa palveluiden toteuttamiskontekstiin, laatu riippuu ihmisistä, keinoista, paikoista ja ajanjaksoista. Aineettomuudella tarkoitetaan, että ne konkretisoituvat vasta palvelun oston jälkeen. Erottamattomuudella tarkoitetaan palvelun suorittamiseen ja kuluttamiseen liittyvää samanaikaisuutta. Palveluita on vaikea tallentaa myöhempää käyttöä tai myyntiä varten, jonka vuoksi se nähdään pilaantuvana. (Eiglier et al. 1997)

Grönroosin (2001) mukaan fyysiset tuotteet ja palvelut eroavat toisistaan: fyysisten tuotteiden valmistajan tuottavat tuotteita asiakkaiden kulutettavaksi, kun palveluyritykset käyttävät resursseja, teknologiaa sekä järjestelmiä asiakkaan palvelemiseksi. Markkinoinnin näkökulmasta nähdään myös selkeä ero: perinteisissä markkinointimalleissa tuote on markkinoinnin lähtökohta, kun taas palvelumarkkinoinnissa palvelukonteksti nähdään toiminnan lähtökohtana. Palvelukontekstissa otetaan kantaa, miten resurssien tulee toimia ja millainen on toivottu lopputulos. (Grönroos 2001)

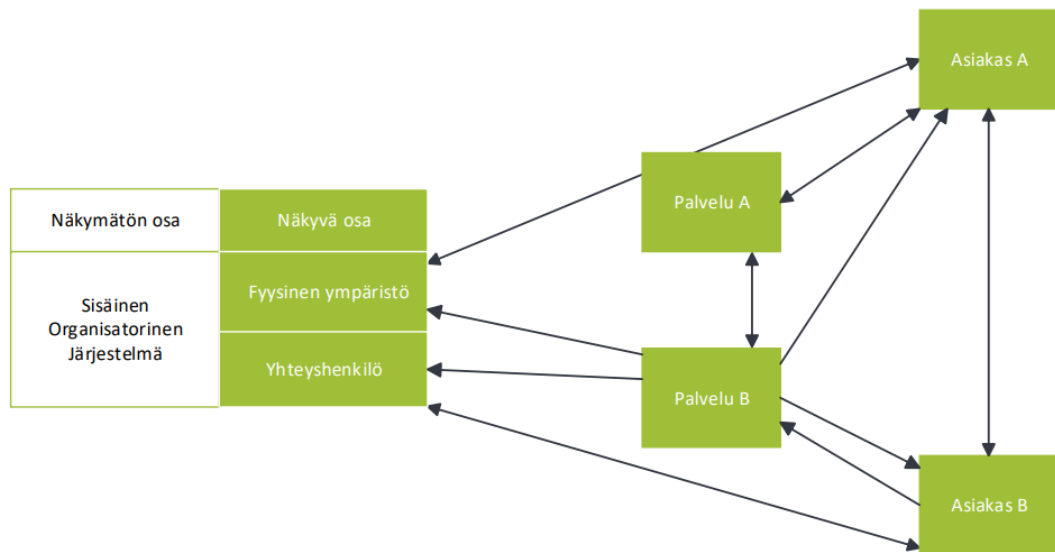
Polyakova & Mirza (2015) tutkivat artikkelissaan palveluiden laatumalleja, niiden kehitystä sekä koetun palvelun laadun käsitettä. Artikkelissa tarkastellaan kuutta eri palvelun laatumallia, josta Bradyn & Croninin (2001) esittelemän mallin selityspotentiaali aiempiin malleihin verrattuna oli paras. Kuvassa 12 esitellään Bradyn & Croninin (2001) luoma palvelun laatumalli.



Kuva 12: *Palvelun laatumalli (Mukailtu lähteestä Polyakova & Mirza 2015).*

Kuten kuvasta 12 nähdään, palvelun laatu muodostuu ensisijaisesti vuorovaikutuksen laadusta, fyysisen ympäristön laadusta sekä lopputuloksen laadusta. Jokaisen ulottuvuuden alle on tarkennettu alalottuvuuksien avulla, mitä tällä tarkemmin tarkoitetaan. Esimerkiksi vuorovaikutuksen laatu koostuu asenteesta, käyttäytymisestä sekä asiantuntemuksesta. (Polyakova & Mirza 2015) Kuvalla 12 pyritään esittämään miten ja mistä asiakkaat muodostavat näkemyksen palvelun laadusta. Brady & Cronin (2001) mukaan asiakkaat muodostavat käsityksen palvelun laadusta monitasoisen suorituskyvyn arvioinnin perusteella ja yhdistämällä nämä arvioinnit. Bradyn & Croninin (2001) mallissa on esitetty konkreettisia alalottuvuuksia, sillä tutkimuksessa tunnistettiin, että konkreettisilla tekijöillä on vaikutusta käsitykseen palvelun lopputuloksen laadusta.

Tarkastellaan seuraavaksi asiakassuhteita ja palveluita, jotta ymmärretään niiden merkitys palveluiden tuotteistamisessa. Parantainen (2007) mukaan palveluita tarjottaessa on tärkeää miettiä, kenen ongelmia halutaan ratkaista. Hänen mukaansa laaja ja epämääräinen asiakaskunta vaikeuttaa palvelun tuotteistamista. Hyötyläinen & Möller (2007) mukaan myös asiakkaan pätevyydellä on vaikutusta palvelun toimivuuteen ja asiakkaan saamaan hyötyyn. Grönroos (1982) esittelee Eiglier & Langeard (1976) kuvaaman mallin, joka kuvaa palveluyrityksen asiakassuhteita (kuva 13).



Kuva 13: Palveluyrityksen asiakassuhteet (mukailtu lähteestä Grönroos 1982).

Kuten kuvasta 13 nähdään, palveluyritys jaetaan kahteen osaan näkymättömään ja näkyvään osaan. Kuvasta nähdään myös kriittisiä resursseja: fyysinen ympäristö, yhteyshenkilö eli yrityksen edustajat esim. työntekijät sekä palvelun kuluttajat. Palvelut voidaan nähdä hyvin henkilöintensiivisinä, jonka vuoksi asiakastyytyvyyteen vaikuttaa tuotteen lisäksi palvelu. Palvelu voi olla jopa ratkaiseva tekijä siinä, kuluttaako asiakas palvelua uudestaan, vaikka tuote olisikin hyvä. Fyysisellä ympäristöllä tarkoitetaan fyysisiä ja teknisiä resursseja. Voidaan nähdä, että fyysisellä ympäristöllä, asiakassuhteilla sekä palveluilla on vaikutusta toisiinsa. Kolmantena kuvasta 13 nähdään asiakkaat eli palvelun kuluttajat, joilla on vaikutusta sekä toisiinsa, että prosessiin. Tästä mallista on tunnistettavissa asiakaslähtöisyys ja heidän vaikutuksensa prosessiin. Asiakaslähtöistä näkökulmaa ei siis palveluajattelun ja tuotteistamisen näkökulmista tule unohtaa.

4.2 Palveluiden tuotteistaminen

Tuotteistaminen mahdollistaa toistettavan prosessin, joka voidaan mukauttaa kasvavaan tuotannon määrään. Palveluiden tuotteistaminen mahdollistaa asiakastarjonnan ja liiketoiminnan tehostamisen entisestään. Toisaalta palvelu voidaan nähdä siis tuotteena. (Harkonen 2021; Hempe et al. 2015) Tuotteistaminen nähdäänkin osana arvonluontiprosessia. Hempe et al. (2015) mukaan palveluinnovaatioilla on selkeä rooli uusien markkinoiden ja mahdollisuuksien löytämisessä, erityisesti korostetaan palveluprosessien tehokasta hallitsemista. Asiakkaiden tarpeiden tunnistaminen sekä

tuottajan kyky vastata tarpeisiin nähdään tärkeänä. Tuotteistaminen nähdään käytäntönä, jossa tuotteet ja palvelut määritellään asiakkaan vaatimusten pohjalta. Innovaatioprosessit ovat yleensä B2B-ympäristössä erittäin monimutkaisia, jonka vuoksi hallinta ja organisointi nähdään erittäin tärkeässä roolissa. (Hempe et al. 2015) Sipilän (1995) mukaan tuotteistus voidaan nähdä selkeiden palvelukokonaisuuksien ja -prosessien kautta ja se voi olla eriasteista (kuva 14).



Kuva 14: Tuotteistamisen asteet (mukailtu lähteestä Sipilä 1995).

Kuvassa 14 tuotteistaminen on jaettu neljään eri asteeseen. Ensimmäinen aste on sisäisten menetelmien tuotteistaminen, jossa työmenetelmiä ja -tapoja on systematisoitu. Toisessa asteessa on palvelu, jossa hyödynnetään esimerkiksi eri it-järjestelmiä apuna antaen tuotetukea. Kolmannessa asteessa palvelu on tuotteistettu eli prosessit ja menetelmät on tuotteistettu palvelun osalta mahdollisimman pitkälle. Neljännessä asteessa on muodostunut monistettava tuote, joka on saatu fyysiseen tai sähköiseen muotoon. (Sipilä 1995)

Tuotteistaminen voidaan nähdä sekä sisäisenä, että ulkoisena tuotteistamisena. Sisäinen tuotteistaminen kertoo organisaation kyvystä luoda ja ulkoinen tuotteistaminen kertoo kyvystä myydä. Sisäinen tuotteistaminen pyrkii tehostamaan palvelun kehittämistä, hankintaa, testausta, sekä prosessin laatua. (Hempe et al. 2015) Sisäinen ymmärrys kasvaa, mikä mahdollistaa asiakkaille paremman palvelun, kun työntekijällä on tiedot selittää palvelun sisältö selkeästi (Harkonen 2021; Sipilä 1995). Sisäisessä tuotteistuksessa ydinidea on, että samaa asiaa ei tehdä useaan kertaan (Sipilä 1995). Ulkoinen tuotteistaminen kertoo organisaation ulkopuolelle palvelun painopisteestä esimerkiksi brändäyksen, markkinoinnin ja koulutuksen avulla. Palvelumallien avulla

voidaan parhaassa tapauksessa vähentää rutiinyyttä ja lisätä uusien innovaatioiden mahdollisuutta. (Hempe et al. 2015) Sipilä (1995) huomauttaa, että sisäisen toiminnan systematisointi on edellytys, jotta ulkoista tuotteistamista voidaan tehdä.

Palveluiden tuotteistaminen voidaan nähdä hyödyllisenä, kun mietitään liiketoiminnan kehittämistä. Harkonen (2021) esittelee tutkimuksessaan palvelun tuotteistamisen havaittuja hyötyjä aiemmasta kirjallisuudesta, joita on huomattavan paljon. Taulukkoon 5 on koottu yhteenveto palveluiden tuotteistamisen hyödyistä. Taulukko 5 on jaoteltu sarakkeisiin, jossa esitellään havaittu hyöty sekä kirjallisuuden määrä, jossa kyseisen hyöty on mainittu.

Taulukko 5: Palveluiden tuotteistamisen hyödyt (mukailtu lähteestä Harkonen 2021).

Havaittu hyöty		Kirjallisuuden määrä (kpl)	Havaittu hyöty		Kirjallisuuden määrä (kpl)
1	Sisäisen ja ulkoisen ymmärryksen kasvaminen	8	18	Tangibilisointi	2
2	Rutiinit	1	19	Palvelualueiden luominen	1
3	Skaalaus hyödyt	1	20	Hinnoittelulogiikka	1
4	Yksilöiden riippuvuuden vähentyminen	1	21	Keskittymisen parantuminen	1
5	Tehokkuus	9	22	Konkretisointi	2
6	Laatu	2	23	Kustannussäästöt	2
7	Jaettu ymmärrys	3	24	Siirrettävyys	1
8	Systematisointi	12	25	Linkitys dataan ja IT:hen	3
9	Kannattavuus	3	26	Yhteinen kieli	2
10	Työntekijät ymmärtävät roolinsa	1	27	Markkinointi ja myynti	2
11	Hyötysuhde	9	28	Rakenteellinen ymmärrys	7
12	Hinta	1	29	Palveluelementtien standardisointi	4
13	Ymmärrettävyys	1	30	Keskeisen hyödyn tunnistaminen	1
14	Formalisointi	11	31	Parempi hallittavuus	4
15	Konkreettisempi hinnoittelu	1	32	Palveluportfolio	4
16	Tiedon jakaminen	2	33	Kustannusrakenne	1
17	Aika	1	34	Tuki palveluvuorovaihtelukselle	2

Kuten taulukosta 5 nähdään, hyötyjä on monia, mikä tukee myös case-organisaation kannalta palvelun tuotteistamista. Kun tarkastellaan taulukkoa tarkemmin, nähdään että muutama havaittu hyöty on mainittu useasti eri lähteissä: Sisäisen ja ulkoisen ymmärryksen kasvaminen, tehokkuus, systematisointi, hyötysuhde, formalisointi sekä rakenteellinen ymmärrys. Myös Sipilä (1995) vahvistaa taulukossa 5 mainittuja hyötyjä. Hän korostaa myös tuotteistamisen merkitystä oppimisessa ja tiedonsiirrossa: henkilön osaaminen ja tuotetuki mahdollistaa hyvän palvelun. Taulukko 5 vahvistaa käsitystä siitä, että palveluiden tuotteistamisella on monitahoinen vaikutus organisaatioon, työntekijöihin sekä asiakkaisiin.

Kriittinen ajattelu havaittujen hyötyjen rinnalla on myös tärkeää. Harkonen (2021) esittelee myös huolenaiheita, joita palvelun tuotteistamisessa on havaittu. Hän mainitsee sisäiset haasteet, joita uusien strukturoitujen toimintatapojen mukana tulee, joka voi heijastua asiakkaisiin. Tässä tapauksessa korostetaan sisäistä viestintää sekä koordinoitua. Formalisoidut palvelut voidaan nähdä myös negatiivisesti, koska pelätään,

että tarpeita ja odotuksia ei oteta huomioon. (Harkonen 2021) Menetelmiä ja prosesseja kehitettäessä on tärkeää huomioida myös ketteryys toimia prosessien ulkopuolella ja muuttaa toimintaa asiakkaan tarpeiden mukaan.

Asiantuntijapalveluiden tuotteistamisen keskiössä voidaan nähdä vuorovaikutusprosessi. Prosessikonsultointi eli kyky lähestyä asiakasta ja auttaa löytämään oikea diagnoosi nähdään jopa tärkeämpänä tekijänä palveluissa kuin selkeä tekninen ratkaisu ongelmaan. Myös kyky tehdä asiakkaan puolesta toimenpide tai tieto sopivasta ratkaisusta nähdään odotuksena asiakkaan puolelta. (Sipilä 1995) Asiantuntijapalveluiden ostamisessa voidaan nähdä riskejä sen abstraktin luonteen ja aineettomuuden vuoksi. Tästä syystä nähdään, että on tärkeää kuvailla palveluita ”palvelupaketeiksi” tai ”palvelutuotteiksi”. Tämä tuo palveluihin konkretiaa, joka asiakkaan on helpompi ymmärtää ja ostaa. (Jaakkola 2011) Myös Sipilä (1995) korostaa aineettomuuden aineellistamista. Hän kuvailee konkretisointia fyysisten todisteiden keräämiseksi, joita asiakas voi hyödyntää ostopäätöksen tukena. Hänen mukaansa asiantuntijaorganisaatiossa referenssit ja case-kuvaukset toimivat erinomaisena keinona kertoa uudelle asiakkaalle, miten voidaan päästä hyvin lopputuloksiin. Myös Shostack (1984) huomauttaa konkretisoinnin tärkeydestä: asiakkaat arvioivat palvelua yleensä aihetodisteiden kuten konsultin tekemän kirjallisen raportin kautta. Tästä syystä brändäys, asiakkaille näkyvä markkinointi ja materiaalit ovat tärkeä osa palvelun tuotteistamista. Näiden avulla voidaan vahvistaa tai ristiriitaistaa kokemusta palveluntarjoajasta. (Shostack 1984)

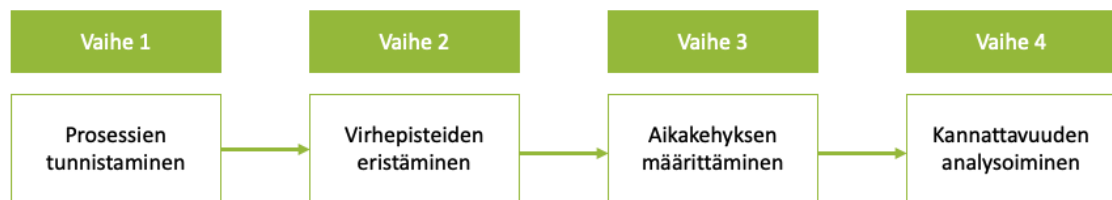
4.3 Prosessimallit palveluiden tuotteistamisessa

Tuotteistamisen tarve tunnistetaan, kun palvelutarjontaa halutaan täsmentää ja standardoida. Tämä mahdollistaa palvelun myymisen sekä ostamisen yhä helpommin. Ilman tuotteistettua palvelua on vaikeaa esittää asiakkaalle, mitä konkreettista palveluntarjoaja voi tehdä hänen hyväkseen. Standardointi helpottaa myyntiä sekä markkinointia. (Jaakkola 2011) Tuotteistamisen tarvetta korostaa myös palveluiden osuuden kasvaminen merkittäväksi osa-alueeksi kehittyneimmissä palvelutalouksissa kuten Yhdysvalloissa ja EU:ssa. Palveluilla on merkittävä osuus, kun mitataan varallisuutta sekä työllisyyttä. (Trott 2008)

Palvelut voidaan nähdä prosessina, jossa on eri toteuttamiskeinoja. Esimerkkinä konsultin tekemä raportti, joka ostetaan tiedon näkökulmasta, ei fyysisen tuotteen takia. (Shostack 1984) ICT-palvelut nähdään monimutkaisena kokonaisuutena, jonka vuoksi palvelusuunnittelu on erityisen tärkeää. Sillä voidaan vaikuttaa palvelun tuotantoon,

toteutukseen sekä asiakkaiden käsityksiin ja tyytyväisyyteen. ICT-palveluiden hallinnalla on vaikutusta myös tuotantokustannuksiin. (Hyötyläinen & Möller 2007)

Kirjallisuudesta tunnistetaan useita eri prosessimalleja, joilla tuotteistusta tehdään. Tarkastellaan ensimmäisenä Shostack (1984) esittelemää mallia, jossa neljässä eri vaiheessa tarkastellaan palvelua prosessina (kuva 15). Shostack (1984) mukaan palvelun tuotteistamisessa on useita eri vaiheita.



Kuva 15: Palvelun tuotteistamisprosessi (Shostack 1984).

Kuten kuvasta 15 nähdään, ensimmäisenä tulisi miettiä prosesseja, jotka muodostavat palvelun. Huomioitavaa on tarkastella myös asiakkaalle näkymättömiä prosesseja, koska tällä voi olla vaikutusta siihen, kuinka asiakas näkee palvelun. Toisena tulisi tunnistaa epäonnistumispisteet ja eristää ne prosesseista. Tällä tavoin voidaan vähentää häiriöitä ja ongelmia lopullisessa palvelussa, kun virhepisteitä on mietitty jo suunnitteluvaiheessa. Kolmantena asiana tulisi määrittää aikakehys. Suunniteltu aikakehys varmistaa standardisuoritusajan, jossa tulee huomioida myös sallitut poikkeamat. Poikkeamien kohdalla on tärkeää miettiä, paljonko työmääräarvion mukaan toteutettava prosessi saa ylittyä, jotta asiakas on edelleen tyytyväinen eikä heikennä laatu-arvioita. Neljäntenä tekijänä tulisi analysoida kannattavuutta esimerkiksi suoritusaikastandardia hyödyntämällä. Suunniteltaessa palvelua tulee siis huomioida, milloin liiketoiminta ei olisi enää kannattavaa ja miten tuottavuutta ylläpidetään. Standardi auttaa suorituskyvyn mittaamisessa ja laadun hallinnassa sekä erottaa kustannustehottoman toiminnan. (Shostack 1984)

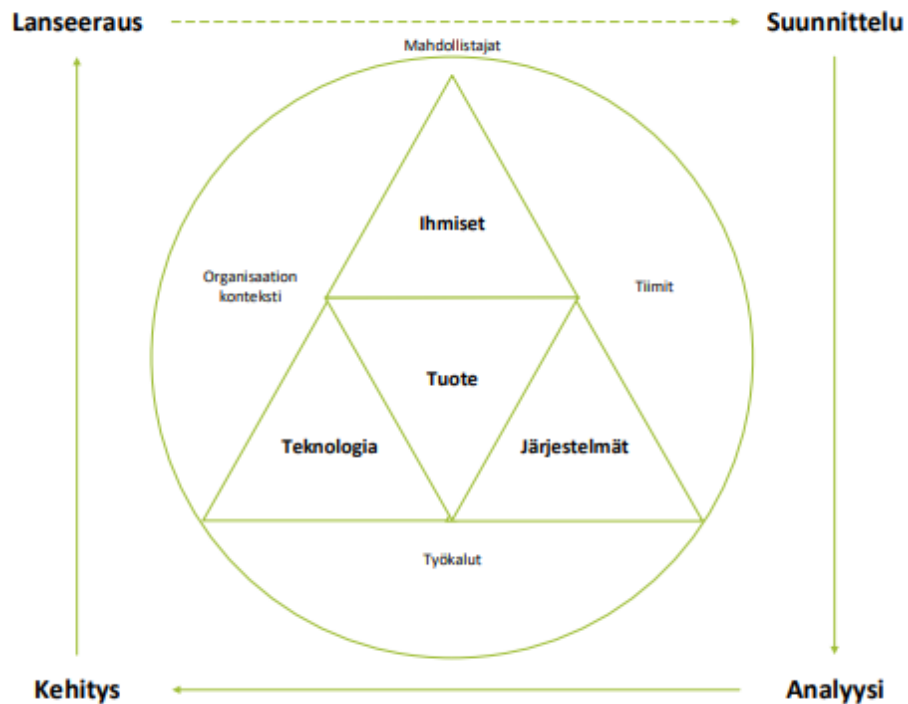
Trott (2008) esittelee kaksi eri palvelukehitysmallia: Scheuing and Johnson (1989) kehittämän palveluinnovaatioprosessin (kuva 16), jossa edetään vaiheittain sekä Johnson et al. (2000) kehittämän iteratiivisen kehitysmallin (kuva 17). Ominaista vaiheittaiselle mallille on toimintojen lineaarinen eteneminen, kun taas iteratiivisessa mallissa on mahdollista kerätä palautetta ja kehittää niitä yhä uudelleen.



Kuva 16: *Palveluinnovaatioprosessi- vaihemalli (mukailtu lähteestä Trott 2008).*

Kuvasta 16 nähdään, että vaihemalli jakautuu kolmeen pääosaan, jotka sisältävät lisää toimenpiteitä. Ensimmäisessä vaiheessa kehitetään ideaa, kun toisessa arvioidaan sitä. Arviointi koostuu mm. konseptin kehityksestä ja testauksesta, liiketoiminta-analyysistä sekä projektin hyväksymisestä. Toteutuksessa palvelu, prosessit ja järjestelmät suunnitellaan ja testataan. Vaiheessa otetaan kantaa myös markkinointiin, koulutukseen sekä pilottikäyttöön. Lopuksi palvelu lanseerataan ja tarkastellaan prosessin jälkeistä vaihetta. Trott (2008) kritisoi vaiheittaisia lineaarisia malleja, koska ne ovat kalliita, aikaa vieviä ja byrokraattisia prosesseja. Hän korostaa ajankäytön merkitystä, jotta markkinamahdollisuus on edelleen olemassa, kun kaupallistaminen on ajankohtaista. Trott (2008) esittääkin näkemyksen, että uuden tuotteen tai palvelun kehittäminen on iteratiivinen prosessi, jonka tulisi jatkua myös sen jälkeen, kun se on tullut markkinoille.

Jotta ymmärretään, mitä iteratiivinen tuotteistamisprosessi tarkoittaa, tarkastellaan seuraavaksi Trotin (2008) esittelemää iteratiivista mallia, jonka alun perin on kehittänyt Johnson et al. (2000). Kuvassa 17 nähdään iteratiivinen malli, jossa tarkoituksena on toistaa vaiheet useaan kertaan ja se antaa mahdollisuuden myös palautteen antoon.



Kuva 17: Palvelukehitysmalli – iteratiivinen malli (mukailtu lähteestä Trott 2008).

Johnson et al. (2000) ehdottaa, että uusi palvelukehitys tulisi olla iteratiivisia sekä epälineaarisia, sillä todellisuudessa prosessi ei edusta lineaarista etenemistä. Kuvassa 17 suunnittelu ja analyysi kuvaavat prosessin suunnitteluvaihetta, kun taas kehitys ja lanseeraus toteutusvaihetta. Suunnitteluvaiheessa tarkoituksena on muotoilla uudet palvelutavoitteet, luoda ja seuloa ideoita sekä kehittää ja testata konsepteja. Suunnitteluvaiheessa tulisi suorittaa myös liiketoiminta-analyysi ja valtuutus. Iteratiivisen mallin toteutusvaiheessa tavoitteena on suunnitella ja testata palvelua, prosessia ja järjestelmää. Toisaalta toteutusvaiheessa tulisi suorittaa myös palvelutestaus, pilottiajo ja testimarkkinointi. Syklin viimeisessä vaiheessa suoritetaan täysimittainen lanseeraus sekä lanseerauksen jälkeinen arviointi. Tämän jälkeen prosessi voidaan iteroida uudelleen. (Johnson et al. 2000)

Kuten kuvista nähdään mallit ovat hyvin erilaisia: toisessa oletetaan, että vaiheiden jälkeen palvelu on valmis ja hyvä, kun iteratiivisessa mallissa usean syklin avulla varmistetaan, että palvelu vastaa asiakkaiden tarpeisiin. Iteratiivisessa mallissa on tunnistettu, että ihmiset, teknologia ja järjestelmät ovat kehityksen keskiössä (Trott 2008). Trotin (2008) esittämät mallit ovat yksityiskohtaisempia ja ne ottavat tarkemmin kantaa palvelun kehityksessä, kun Shostackin (1984) malli voidaan nähdä yksinkertaisempaan.

4.4 Palveluiden paketointi

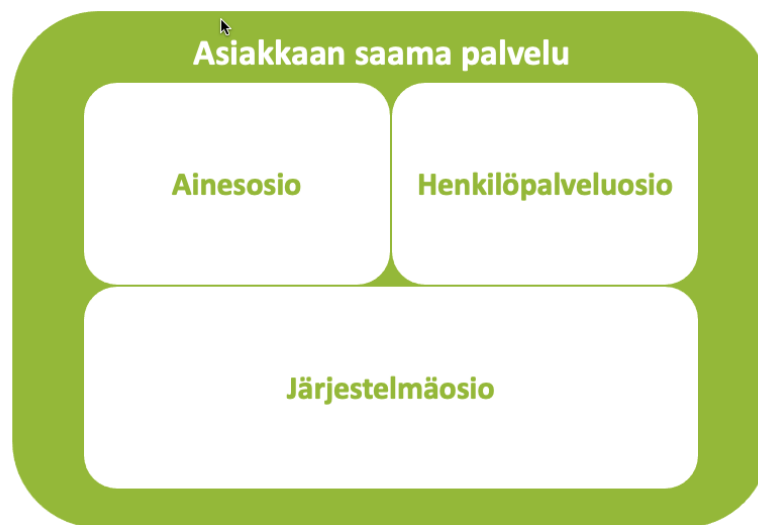
Monimutkaisuus on keskeinen teema, joka palveluiden ympärillä on tunnistettu. Tämä huomio on tehty etenkin, kun palveluita verrataan tuotteisiin. On kuitenkin tunnistettu, että palveluiden paketointi on yksi keino vähentää monimutkaisuutta. (Hyötyläinen & Möller 2007) Myös Jaakkola (2011) esittelee tutkimuksessaan näkökulmaa, jossa on tunnistettu, että palvelun vaihtelua ja monimutkaisuutta voidaan vähentää jakamalla palvelu pienempiin osiin. Tällä tarkoitetaan palvelumoduulien tai -pakettien kehittämistä. Palveluiden paketointi mahdollistaa konkreettisemmän, jopa tuotemaisen tunteen ostajalle. Tällöin asiakas tietää mitä ostettu palvelu sisältää, milloin sen saa ja mitä se maksaa. Prosessien ja menetelmien standardisoinnin yksi keskeinen tavoite nähdään liittyvän tehokkuuteen ja tuottavuuteen, esimerkkinä tarjousten tekeminen, joka noudattaa pitkälti samaa kaavaa joka kerta. (Jaakkola 2011)

Sipilä (1995) mukaan palvelupaketilla tarkoitetaan ydin- ja tukipalveluista koostuvaa kokonaisuutta. Grönroos (1998) kuvailee tuotteen omaista palvelua palvelupaketiksi, joka koostuu erilaisia konkreettisista tai aineettomista palveluista. Palvelupaketit koostuvat kahdesta osasta ydinpalveluista sekä tukipalveluista (Sipilä 1995). Grönroos (1998) jakaa peruspalvelupaketin kolmeen osaan johtamissyistä: ydinpalveluihin, avustaviin palveluihin sekä tukipalveluihin. Avustavat palvelut helpottavat ydinpalvelun käyttöä, ja ovatkin yleensä vaatimus, jotta ydinpalvelua käytetään. Tukipalveluita käytetään arvon lisäämiseksi tai niitä on, jotta pystytään erottumaan kilpailijoista edukseen. Suurin ero siis avustavilla- ja tukipalveluilla on, että ilman avustavia palveluita ydinpalvelua ei voida käyttää, kun ilman tukipalvelua ydinpalvelun käyttö on mahdollista. (Grönroos 1998) Palvelupaketit jaetaan yleensä kolmeen eri tarkoitukseen: riisuttu palvelupaketti, joka on tarkoitettu pienasiakkaille, keskikokoinen palvelupaketti, jota myydään kaikille asiakkaille sekä erikoispaketti, jossa on räätälöityjä ominaisuuksia asiakkaiden tarpeisiin (Sipilä 1995).

Jaakkola (2011) esittelee asiakaslähtöisemmän näkökulman palveluiden paketoinnille, jossa palvelu tulisi jakaa palveluversioihin. Tutkimuksesta havaitaan, että palvelu voidaan standardoida sisällön, prosessin ja hinnan osilta. Tämä tarkoittaa, että palvelu tuotetaan ennalta määritetyllä tavalla, kuitenkin jokaisen asiakkaan oman tarpeen mukaan. Vertauskuvana tutkimuksessa käytetään värityskirjaa: linjat ovat samat, mutta jokaiselle asiakkaalle käytetään omia värejä. Toisaalta nähdään myös, että palvelua ei voida kokonaan standardoida, mikäli se halutaan nähdä asiakaslähtöisenä. Voidaan kehittää palvelumoduuleja, joita voidaan yhdistää joustavasti. Tutkimuksessa esiin nousi myös, että tuotemaisuus palveluissa helpottaa kommunikointia asiakkaan kanssa. (Jaakkola 2011)

Tutkijoiden mukaan suuret räätälöinnit sekä heterogeenisuus aiheuttavat asiantuntijapalveluiden hallinnassa haasteita, etenkin palvelujen viestinnän, edistämisen ja hinnoittelun kannalta. (Jaakkola 2011) Tarkastellaan seuraavaksi kirjallisuuden tuomia näkökulmia palveluiden hinnoitteluun, edistämiseen (työmäärät) sekä viestintään.

Laitisen (2007) mukaan palveluiden hinta perustuu palvelun asiakkaalle tuomaan arvoon ja sen on oltava kilpailukykyinen. Anttonen (2003) tukee Laitisen (2007) tuomaa näkemystä toteamalla, että asiakas punnitsee hintaa lopputuloksen hyötyjä tarkastellen, eikä toimittajan tuotantokustannuksia vasten. Asiantuntijapalveluiden hinnoittelu sekä hinnan arviointi nähdään haastavana. (Wirtz et al. 2021) Selkeästi ilmaistu ja kommunikoitu hinnoittelu on tärkeä osa tuotteistettua palvelua. Palvelujen hinnoittelu perustuu erilaisiin tavoitteisiin esimerkkinä markkinaosuuden kasvu tai pitkän aikavälin kilpailukyky. (Wirtz et al. 2021) Nähdään, että on erittäin tärkeää kuvailla jo projektin neuvotteluvaiheessa hinnoittelua sekä asiakkaan esitellä budjettia, jotta hinnoittelun osalta ei tule haasteita projektin edetessä. (Sipilä 1995) Sipilä (2003) esittelee karkean jaottelun, mistä erilaisissa palveluissa maksetaan (kuva 18).



Kuva 18: Asiakkaan saaman palvelun jaottelu (mukaillen Sipilä 2003).

Kuten kuvasta 18 nähdään, palvelu voidaan jakaa kolmeen osioon. Ainesosiolla tarkoitetaan paljonko asiakas saa hinnassa ainetta tai tavaraa. Henkilöpalveluosiossa otetaan kantaa paljonko asiakas saa hinnassa palvelua henkilöltä. Järjestelmäosio osoittaa paljonko asiakkaan maksamasta hinnasta menee palvelujärjestelmän ylläpitoon. Sipilän (2003) esittelemässä mallissa eri palveluissa osioiden suuruus vaihtelee. Esimerkkinä hän esittelee asianajotoimiston, jossa 100 % jakautuu seuraavasti: henkilöpalveluosio 55 % ja järjestelmäosio 45 %. Laitisen (2007) mukaan palvelujen hinnat joko asetetaan tai neuvotellaan. Mikäli palvelu sisältää paljon materiaalia, tulee hinnan perustua selkeämmin kustannuksiin. Toisaalta, jos palvelu ei

sisällä materiaalia, tulee asiakkaan näkemys ottaa vahvemmin huomioon. Anttonen (2003) puolestaan esittää, että työn hinnoittelu perustuu yleensä aikaan. Henkilökustannukset toimivat pohjana hinnoittelulle, kun toimittaja arvioi kustannuksia.

Lähtökohtana Anttonen (2003) näkee, että asiakas haluaisi hinnoitella työn kiinteäksi ja toimittaja veloittaa kaikki tunnit. Nähdään kuitenkin, että sekä ostaja, että myyjä haluavat minimoida epävarmuutta (Anttonen 2003). Esitelläänkin seuraavaksi eri hinnoittelumalleja ja niiden sopivia käyttökohteita, jotta löydetään tämän tutkimuksen kannalta selkeä suunta palvelupaketin hinnoittelutavalle niin, että se palvelisi jokaista osapuolta parhaalla mahdollisella tavalla. **Aikaveloitus** sopii hinnoittelutavaksi, kun projektissa on paljon epäselviä tekijöitä, eikä ennakkotapauksia ole riittävästi. **Urakkahinnoittelussa** tyypillistä on, että sitä hyödynnetään projekteissa, joissa on paljon aiemmin tunnettuja elementtejä. Projektissa voidaan olettaa muutokset vähäisiksi. **Tavoitehinnoittelu** on aikaveloituksen ja kiinteän hinnoittelun välimuoto, jossa sovitaan perustyömäärä sekä aikaperusteinen veloitus. Mikäli tavoite ylitetään, voidaan hinnoittelu määrittellä sopimuksessa asiakkaan hyväksi esimerkiksi yli 10 % ylityksessä asiakas saa 50 % alennuksen työn tuntihinnasta. Tavoitehinnoittelussa asiakkaalla ja toimittajalla on selkeä yhteinen tavoite projektin aikataulun ja kustannusten osalta. (Anttonen 2003)

Tarkastellaan seuraavaksi kirjallisuuden tuomia huomioita työmääräarviointeihin. Anttonen (2003) tunnistaa, että työmääräarviointit nähdään haasteellisena etenkin uusilla tekniikoilla toteutetuissa tietojärjestelmäprojekteissa. Tavoitteena on kuitenkin, että karkea arvio pystytään määrittelemään, joka tarkentuu työn edetessä. Myös työmääräarvioiden kohdalla, kuten hinnoittelussakin, historiadatalla on vaikutusta. Mikäli esimerkiksi tekniikka, millä tietojärjestelmä tehdään, on uutta, voidaan pohjana pitää jotakin samankaltaista tai suuruista toteutusta, jonka avulla työmäärä voidaan pyrkiä määrittelemään. Karkeaan työmäärän tekemiseen voidaan ottaa avuksi jaottelu, jossa tehtävät jaetaan pienempiin osiin. Tämän avulla voidaan esimerkiksi viikkotasolla arvioida kauanko toteutuksen tekeminen kestää ottamatta kantaa sen enempää lukuihin. (Anttonen 2003)

Kommunikaatio on tärkeä tekijä ohjelmistokehityksessä. Säännöllinen viestintä on keino rakentaa luottamusta tiimeihin, mikä edesauttaa ohjelmistokehityksen tehostamisessa. Aiemmista tutkimuksista on tunnistettu, että samoissa tiloissa työskentelevät ohjelmistotiimit saavuttavat huomattavasti korkeamman tuottavuuden, kuin ne tiimit, jotka eivät ole samassa paikassa. Tämän on selitetty johtuvan jatkuvasta interaktiivisesta viestinnästä. Viestintä voidaan jakaa kahteen tyyppiin: epämuodolliseen ja muodolliseen. Epämuodollisella viestinnällä tarkoitetaan yritysten työntekijöiden välisiä

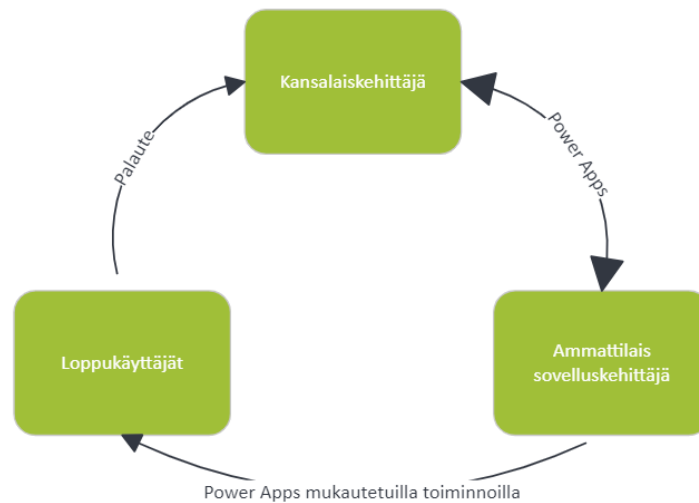
keskusteluita kuten puhelut ja sähköpostiviestit. Muodollisella viestinnällä tarkoitetaan dokumentteja sekä virallisia kokouksia. On kuitenkin tunnistettu, että ohjelmistokehitysprojektien sekä sidosryhmien välinen viestintä on haaste. Esimerkkinä eri toimijoiden tarpeet kuten loppukäyttäjä, kehitystiimi ja ylläpitotiimi. Kommunikointi tapahtuu eri näkökulmista, joka saattaa tuottaa haasteita. (Pikkarainen et al. 2008) Anttonen (2003) toteaa, että vuorovaikutus voidaan nähdä projektin voimalähteenä. Myös Pikkarainen et al. (2008) tukee tätä ajatusta mainitsemalla, että kommunikaatio nähdään yhtenä menestystekijänä. Edellä esitettyjen perusteluiden perusteella myös tutkimuksen empiirisessä osuudessa tullaan selvittämään viestinnän näkökulmaa.

5. YHTEISTYÖKEHITTÄMINEN LOW-CODELLA JA SEN HYÖDYNTÄMINEN PALVELUN TUOTTEISTAMISESSA

Tässä luvussa esitellään keskeisiä havaintoja, joita kirjallisuuden perusteella aiheen ympäriltä löydettiin. Luvussa esitellään myös kirjallisuuden vastaukset tutkimuskysymyksiin.

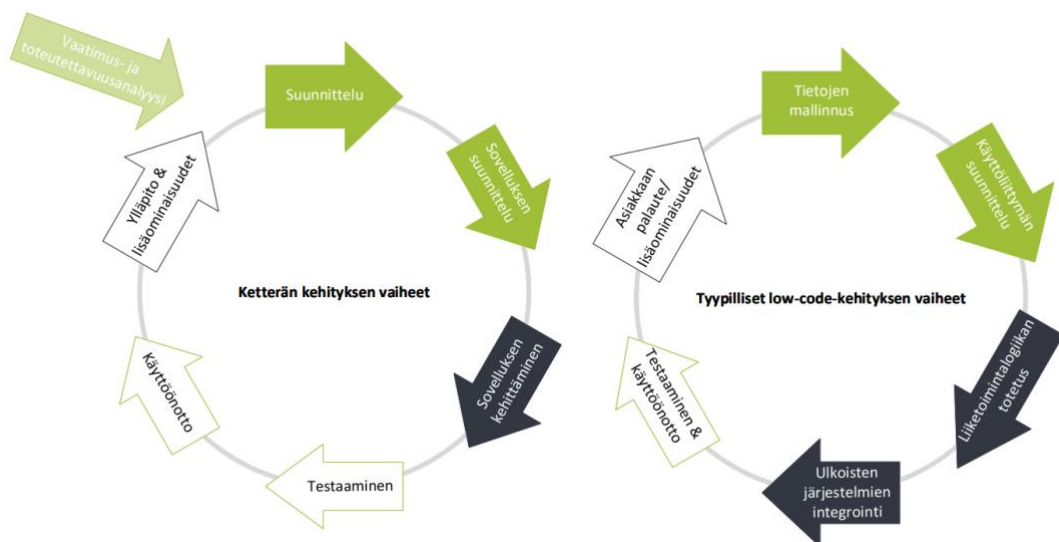
5.1 Havaintoja kirjallisuudesta

Tässä luvussa kootaan yhteenveto teorialuvuista 3 ja 4, jotka toteutettiin kirjallisuuskatsauksena aiheen ympäriltä. Tässä luvussa pyritään löytämään vastaus teorian pohjalta aiemmin määriteltyihin tutkimuskysymyksiin. Teoriaosuuden tarkoituksena oli tunnistaa kirjallisuuden tieto aiheesta ja luoda pohjaa empiiriselle tutkimukselle. Teorialuvuissa on tunnistettu, että yhteistyökehittäminen low-code -teknologian kontekstissa on relevantti tutkimuskohde. Yhteistyökehittämistä sekä sen hyödyntämistä low-code -ratkaisujen yhteydessä on tutkittu ja aiheen ympäriltä on löydetty selkeitä toimintamalleja. Yhteistyökehittämiselle käytetään kirjallisuudessa monia eri termejä kuten yhteisluominen, yhteistuotanto sekä fuusiokehittäminen. Jokaisessa tapauksessa kyseessä on yhteistyössä tapahtuva kehitystyö eri liiketoiminnan osajien kesken. Boyer et al. (2021) on luonut yhteistyökehittämiselle mallin, jossa otetaan kantaa roolitukseen sekä vuorovaikutukseen eri tiimiläisten välillä (kuva 19).



Kuva 19: Fuusiokehittämisen malli (mukailtu lähteestä Boyer et al. 2021).

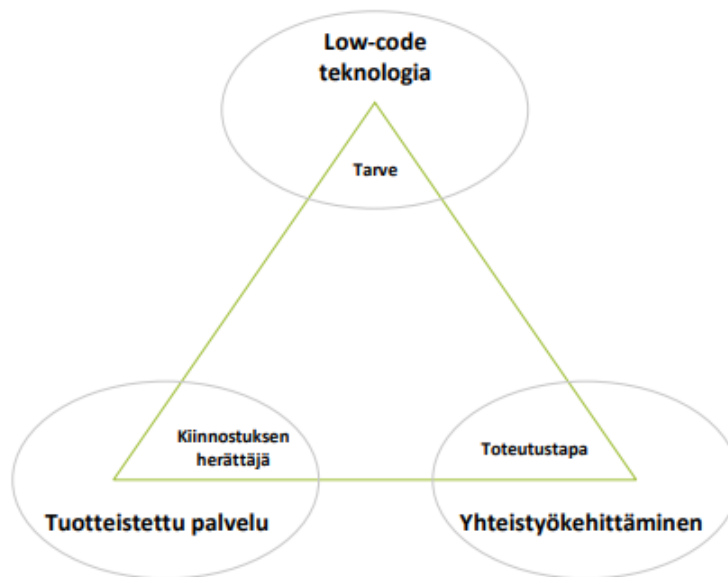
Teorian luvussa 3 tutkittiin myös ketterää ohjelmistokehitystä low-code -teknologian näkökulmasta ja havaittiin, että yhteistyökehittämisen malleilla on samoja piirteitä ketterän kehityksen prosessimallien kanssa. Al Alamin et al. (2021) esittämässä kuvauksessa vertaillaan perinteisiä ketterän ohjelmistokehityksen vaiheita low-code -kehityksen kanssa ja tästä havaitaan, että yhtäläisyyksiä on paljon (kuva 20).



Kuva 20: Ketterä kehitys vs. low-code -kehitys (mukailtu lähteestä Al Alamin et al. 2021).

Teoriassa käsitellään myös arvon yhteisluontia, jolla nähdään olevan selkeä yhteys yhteistyökehittämisen kanssa. Kirjallisuudesta tunnistettiin, että kuvissa 19 ja 20 esitetyt mallit nähdään lähtökohtana yhteistyökehittämisen mallin luomiselle, joka on yksi tutkimuksen keskeisin tavoite.

Toinen osa teoriasta eli luku 4 keskittyy palveluiden ja tuotteistamisen maailmaan. Kirjallisuudesta havaitaan, että palveluilla sekä niiden tuotteistamisella on vaikutusta yrityksen toimintaan. Kilpailuetu on vahva ilmiö, mikä kirjallisuudessa korostuu hyvän, laadukkaan ja tuotteistetun palvelun ympärillä. Luvussa 4 esitellään myös, mitä palvelupaketointi tarkoittaa ja mikä on hinnoittelun merkitys palveluiden tuotteistamisessa. Kuvaan 21 on tiivistetty tutkimuksen aihealueet sekä kirjallisuuden perusteella tunnistettu yhteys eri tekijöiden välillä.



Kuva 21: Kirjallisuudesta tunnistetut tekijät ja yhteydet tutkittavien aiheiden ympäriltä.

Kuvassa 21 esitetään tutkijan tekemiä huomioita tutkimuksen aihealueiden ympäriltä sekä yhteys, jonka tutkija aiheiden välillä on tunnistanut. Low-code -teknologian, palveluiden tuotteistamisen sekä yhteistyökehittämisen ympäriltä on löydetty selkeä yhteys. Low-code -teknologia itsessään mahdollistaa uusia, innovatiivisia ratkaisuja nopealla toteutuksella. Kun asiakas ymmärtää tämän tarpeen, etsii hän toteuttajaa ratkaisulle. Tuotteistettu palvelu herättää asiakkaan mielenkiinnon, joka tuo myös kilpailuetua palveluntarjoajalle. Kirjallisuudessa tunnistettiin, että erilaisia ketteriä kehitysmalleja low-code -teknologialle on muodostettu (kuva 19 ja 20). Yhteistyökehittämisen mallia voidaan pitää siis perusteltuna mallina toteuttaa low-code -kehitystä. Palvelulla ja toteutustavalla on merkitystä siihen, minkälaista arvoa saadaan luotua yhteistyössä asiakkaan kanssa sekä mikä on koettu arvo. Täten voidaan todeta myös, että tuotteistetulla palvelulla ja yhteistyökehittämisellä on vaikutusta siihen, minkälainen mielikuva teknologiasta asiakkaalle annetaan.

5.2 Kirjallisuuden vastaukset tutkimuskysymyksiin

Kun mietitään tutkimuksen päätutkimuskysymystä, ”**miten low-code -palveluja voidaan tuotteistaa yhteistyökehittämisen mallin avulla?**”, tunnistetaan, että kuva 21 vastaa tutkimuskysymykseen suurilta osin. Ensinnäkin kirjallisuuden pohjalta voidaan todeta, että low-code -teknologiaan liittyviä palveluja voidaan tuotteistaa. Teoriasta tunnistettiin, että tuotteistamisen lähtökohtana on toistettava prosessi, joka voidaan mukauttaa kasvavaan tuotannon määrään. Palveluiden tuotteistaminen mahdollistaa asiakastarjonnan ja liiketoiminnan tehostamisen entisestään. (Harkonen 2021; Hempe et al. 2015) Yhteistyökehittämisen malli antaa tähän erinomaisen viitekehyksen. Luomalla yhteistyökehittämisen mallin low-code -teknologialle voidaan määritellä sisäisesti, kuka tekee low-code -teknologialla tuotettuun ratkaisuun mitä, missä vaiheessa ja miten. Boyer et al. (2021) kehittämällä mallilla voidaan vastata roolitukseen liittyviin kysymyksiin. Al Alamin et al. (2021) esittämällä mallilla pystytään ottamaan kantaa vaiheisiin. Molemmista malleista puuttuu varsinaiset vastuunjaot, joka on yksi case-organisaatiossa tunnistettu haaste. Empiirisen osuuden tavoitteena on siis tuoda etenkin vastuunjaon näkökulmista malleihin kehitystä sekä vastauksia kysymyksiin kuka missä vaiheessa ja miten. Tuotteistettu yhteistyökehittämisen malli antaa myös selkeää suuntaa siihen, mitä asiakkaalle kerrotaan aikataulusta, hinnasta ja odotuksista, joita asiakkaalta oletetaan. Kun organisaatioon on luotu yhteistyökehittämisen malli, voidaan palvelu tuotteistaa sen avulla. Tuotteistamisessa voidaan hyödyntää tuotteistamisen prosessimalleja, kuten kappaleessa 4.3 esitellään. Low-code -teknologia itsessään antaa mahdollisuuden kappaleessa 3.1 kuvan 5 esitettyjen ominaisuuksien puolesta erinomaisen pohjan ja raamit tuotteistamiselle (Pinho et al. 2022; Rosa-Bilbao et al. 2023; Sanchis et al. 2019).

Pohditaan seuraavaksi alatutkimuskysymystä ”**mitä palveluiden tuotteistaminen tarkoittaa low-code -teknologialle?**”. Hempe et al. (2015) toteavat tutkimuksessaan, että palveluiden tuotteistamisella on vaikutusta päätöksentekoprosesseihin, kun mietitään ostoa ja innovaatiota. Heidän mukaansa yhteistyökehittäminen sekä ketterä toiminta mahdollistuu sekä innovaatiokyky kasvaa. Voidaan todeta, että case-organisaation tapauksessa on erittäin tärkeää ymmärtää low-code -palveluiden tuotteistamisen merkitys, koska sillä on vaikutusta liiketoiminnan kasvuun sekä low-code -teknologialla tehtäviin ratkaisuihin. Kirjallisuudesta havaitaan, että palveluita voidaan tuotteistaa erilaisten prosessimallien kautta. Monissa tuotteistamismalleissa havaitaan samoja ominaisuuksia, mutta vaiheita voi olla useampi sekä mallien keskiöissä voidaan nähdä eri toimijoita. Kappaleessa 4.3 esitellään kolme erilaista palvelukehitysmallia:

Shostackin (1984) sekä Scheuingin and Johnsonin (1989) kehittämät vaihemallit sekä Johnson et al. (2000) iteratiivinen malli. Malleissa nähdään paljon yhtäläisyyksiä. Johnson et al. (2000) kehittämän iteratiivisen mallin (kuva 17) etuna on kuitenkin useamman tuotteistamiskierroksen tuoma varmuus. Palvelutuotteessa on todennäköisemmin osattu huomioida eri asioita, kun niitä on mietitty useampaan kertaan. Kun mietitään low-code -teknologian kontekstissa palvelun tuotteistamista voidaan nähdä, että iteratiivisella mallilla on paljon potentiaalia. Kirjallisuudessa tunnistettiin, että low-code -kehitys nähdään hyvin ketteränä ja iteratiivisena prosessina itsessään, jolloin myös siihen liittyvät palvelut tulisi olla ketteriä. On kuitenkin tärkeä muistaa, että ketteryys ei tarkoita, että eri prosessit tulisi unohtaa. Tämä on hyvä tekijä muistaa myös palveluiden kohdalla. Sillä jos huomataan, että jokin osa palvelutuotteesta ei ole toimiva, on se hyvä muuttaa prosessimallin mukaan, jotta sitä osataan analysoida ja tehdä siitä parempi. Palveluiden tuotteistaminen mahdollistaa systemaattisten prosessien toteuttamisen low-code -teknologialla, joka on toistettavissa. Tuotteistamisella on vaikutusta niin case-organisaation sisäisiin kuin ulkoisiin prosesseihin, kuten viestintään ja hinnoitteluun. Low-code -palveluiden tuotteistaminen tarkoittaa parhaassa tapauksessa Harkonen (2021) esittelemiä tuotteistamisen hyötyjä (kappale 4.2, taulukko 5).

Toinen alatutkimuskysymys ottaa kantaa yhteistyökehittämiseen: ”**miten yhteistyökehittämisen mallia voidaan hyödyntää tuotteistamisessa?**”. Kuten aiemmin todettiin, yhteistyökehittämisellä nähdään selkeä merkitys palvelun tuotteistamisessa. Yhteistyökehittämisen malli mahdollistaa palveluiden tuotteistamisen ja toimii jo itsessään tuotteistettuna mallina. Mallin avulla voidaan kuvata tarkasti low-code -teknologialla tehtyjen ratkaisujen toteutustapa. Yhteistyökehittämisen mallia voidaan hyödyntää tukena palveluiden tuotteistamisprosessin kanssa. Kirjallisuuden pohjalta voidaan ehdottaa, että yhteistyökehittämisen mallin ja palvelumallin tulisi syntyä ja kehittyä yhdessä. Mikäli yhteistyökehittämisen mallia muotoillaan uudelleen, tulee myös palvelutuotetta tarkastella, jotta ne eivät ole ristiriidassa keskenään. Yhteistyökehittämisen malli mahdollistaa innovatiivisen ja ketterän toiminnan. Tämän pohjalta ehdotetaan, että palvelumalli olisi ketterä, jolloin arvoa voidaan luoda yhdessä asiakkaan ja palvelutarjoajan kanssa Grönroos-Voima -mallin mukaisesti (kappale 3.5, kuva 11). Kirjallisuuden pohjalta yhteistyökehittämisen malli ottaa kantaa rooleihin low-code -kehityksessä sekä eri vaiheisiin, jota low-code -kehityksessä on tunnistettu. Kirjallisuuden perusteella nähdään, että yhteistyökehittämisen mallin rooli palveluiden tuotteistamisessa on ottaa kantaa etenkin sisäiseen tuotteistamiseen. Tavoitteena on, että empiirisessä osuudessa saadaan myös joitakin näkökulmia ulkoiseen

tuotteistamiseen, mutta tässä kohtaa tutkimusta havaitaan, että vastaukset tuotteistamisen osalta liittyvät vahvemmin sisäiseen tuotteistamiseen ja antavat jatkotutkimukselle pohjan keskittyä myös ulkoiseen tuotteistamiseen.

6. EMPIIRISEN TUTKIMUKSEN TULOKSET

Tässä luvussa käsitellään haastatteluiden tuloksia. Haastatteluiden tulosten käsittely on jaettu kahteen eri ryhmään: asiakashaastattelut sekä palveluntarjoajan haastattelut. Molemmille ryhmille oli omat haastattelurungot, kuitenkin samoja teemoja noudattaen. Taulukkoon 6 on koottu haastateltavat koodattuina. Sama taulukko esitellään jo aiemmin teorian yhteydessä kappaleessa 2.5.

Taulukko 6: Haastateltavat koodattuina.

Haastateltava	Yritys
H1	A
H2	B
H3	C
H4	D
H5	A
H6	E
H7	F
H8	F

Koodauksilla H1-H6 tarkoitetaan asiakkaan edustajia, joiden empiirisen tutkimuksen tuloksia esitellään tarkemmin kappaleessa 6.1. Koodaukset H7 ja H8 edustaa palveluntarjoajan työntekijöitä, joiden haastatteluiden tuloksia käsitellään tarkemmin kappaleessa 6.2. Luvun viimeisessä kappaleessa esitellään yhteenvetona tunnistettuja teemoja ja alateemoja sekä asiakkaiden, että palveluntarjoajien tuloksista.

6.1 Asiakashaastattelut

Kaikilla kuudella asiakkaalla, joita haastateltiin, oli aiempaa kokemusta tai tietoa low-code -teknologiasta. Osalla low-code -ratkaisu oli jo käytössä, osalla kehitysvaiheessa ja osalla tulossa kehitykseen. Haastattelut oli rakennettu neljän pääteeman ympärille: low-code -teknologia, yhteistyökehittäminen, palvelut ja palveluiden tuotteistaminen sekä ratkaisu. Ratkaisulla tarkoitetaan low-code -teknologialla tehtyä ohjelmistoa esimerkiksi sovellusta. Asiakkaille, joille low-code -ratkaisua ei ollut vielä haastatteluvaiheessa tehty tai se oli kesken, jätettiin teeman *ratkaisu ja muita huomioita* kysymykset kysymättä. Kysymykset liittyivät jo toteutettuun low-code -ratkaisuun, joten tämän vuoksi oli perusteltua jättää kysymykset kysymättä.

Kuten kappaleessa 2.6 esiteltiin, haastattelut analysoidaan teemoittelua hyödyntäen. Teemat mukailivat hyvin teoriasta haastattelurunkoon johdettuja neljää pääteemaa, jotka esiteltiin kappaleessa jo aiemmin. Tarkastellaan seuraavaksi haastatteluilla kerättyä aineistoa sekä sieltä löydettyjä teemoja sekä alateemoja.

Low-code -teknologia ja arvonluonti

Tämän teeman tarkoituksena on esittää, minkälaista ymmärrystä asiakkailta on low-code -teknologiasta. Toisaalta tarkoituksena on perehtyä myös arvonluontiin low-coden kontekstissa sekä perehtyä haastateltavien rooleihin sekä ratkaisuihin, joita heidän organisaatioihin on jo toteutettu. Näillä pohjatiedoilla oletetaan olevan vaikutusta siihen, minkälaisia vastaukset ovat ja mistä näkökulmista haastateltavat kysymyksiä ajattelevat.

Haastatteluissa ilmeni, että low-code -teknologia ymmärrettiin käsitteenä hyvin samalla tavalla kuin kirjallisuudessa luvussa 3.1 esiteltiin (H1, H2, H3, H5, H6). Erityisesti teknologian ominaisuuksista nousi esiin ohjelmoinnin helppous, nopeus, ratkaisuiden käytännöllisyys sekä vähäinen koodin määrä. Yhdessä haastattelussa nousi esiin myös ymmärrys siitä, että tämänkaltaista avustavaa teknologiaa on ollut jo 1980-luvulla (H2). Haastateltava tunnisti low-code -teknologiasta hyvin samoja piirteitä, kuten valmiit moduulit ja mallinnus.

”Tuo teknisten ratkaisuiden tekemisen lähemmäksi yritystä.” (H3)

”Tuonut mahdollisuuden tehdä nopeita käytännönläheisiä ratkaisuja yritykseen.” (H3)

”Ei tarvitse kovinkaan perusteellisesti tietää tai tajuta ohjelmointikieltä tai erilaisia ohjelmointikieliä, vaan alusta tarjoaa ohjelmoinnin mahdollisimman helposti ja tietyllä tavalla yksinkertaisesti. Totta kai jotain kooditusta pitää ymmärtää, kun puhutaan low-codesta, mutta ei kauhean syvällisesti.” (H6)

Low-code -teknologian tuoma arvo jakautui vastaajien kesken hyvin erilaisiin näkemyksiin, johon todennäköisesti vaikutti yritykselle tehdyt ratkaisut sekä missä roolissa haastateltava työskentelee. Yksi arvoa tuova tekijä, joka mainittiin, oli kehityksen nopeus (H2). Haastatteluissa tunnistettiin myös kustannussäästöjen mahdollisuus, järjestelmien valvonta-/hälytystoiminta, henkilökohtaisen työn tehostuminen sekä kansalaiskehittäjien mahdollisuus. Pääteemana arvonluonnin näkökulmasta nousi esiin se, että low-code -teknologia mahdollistaa erilaisia liiketoimintaprosesseja digitalisaation näkökulmasta (H3, H5).

”Kustannussäästöä, että pysytään itse tekemään toteutuksia ICT-puolella. Tärkein anti on, että saadaan prosessit sujumaan.” (H3)

”Se, että ostetaan kolmannen osapuolen valmiita ratkaisuita, joudutaan yleensä sovittamaan oma tekeminen olemassa olevan järjestelmän ehdoilla, kun taas tässä tapauksessa me pystytään tekemään järjestelmä meidän ehdoillamme.” (H3)

”Kehityksen nopeus, pääsee nopeasti tekemään tietyn asteisia protoja, josta päästään arvioimaan, että onko oikea suunta vai ei.” (H2)

Yhteistyökehittäminen ja low-code -teknologia

Tämän teeman tarkoituksena on pohtia yhteistyökehittämistä kokonaisuutena. Minkälaisessa roolissa asiakkaat mieluiten toimisivat, mitä case-organisaatiolta odotetaan sekä onko ketterän kehityksen piirteet sopiva lähtökohta yhteistyökehittämisen mallille. Teemassa kartoitettiin myös sitä, miten yhteistyökehittäminen ymmärretään ja mitä se tarkoittaa asiakkaille, jotta case-organisaatiossa voidaan ymmärtää yhteistyötä ja kumppanuutta yhä paremmin.

Yhteistyökehittämisen teemassa tunnistettiin, että asiakkaat korostivat ohjaamista, apua, tukea sekä asiantuntijuutta, joita odotetaan palveluntarjoajalta (H2, H4, H6). Usea haastateltava tunnisti, että yhteistyökehittämisen lähtökohta on asiakkaan tarve low-code -ratkaisulle (H2, H4, H5, H6). Kun asiakas tunnistaa tarpeen, esittää se idean palveluntarjoajalle, jonka tehtävänä on jalostaa, kehittää ja iteroida ideaa yhdessä asiakkaan kanssa eteenpäin. Yhteistyökehittämisessä tärkeänä nähtiin myös pitkäjänteisyys ja vuorovaikutus, joka mahdollistaa kumppanuuden palveluntarjoajan kanssa.

”Asiakkaan tarve on se mihin haetaan ratkaisua.” (H2)

”Lähdetään tekemään yhdessä, kun meillä on tarpeita asiakkaan näkökulmasta, niin teiltä (palveluntarjoajalta) pitäisi tulla asiantuntemusta ja kokemuksen kautta sitä, että miten tämänkaltaista toteutusta kannattaisi lähteä viemään eteenpäin, millä teknologialla ja millä tavalla. Tämä on jatkuvaa vuorovaikutusta.” (H2)

”Yhteiskehittäminen on ehdoton lähtökohta sille, että pystytään saamaan meidän liiketoimintaamme parhaiten palveleva osakokonaisuus tai kokonaisuus.” (H2)

Yhteistyökehittämisen teemassa pyrittiin selvittämään asiakkaiden näkemyksiä työskentelytavoista low-code -teknologian kontekstissa. Koska kirjallisuudessa oli tunnistettu ketterän kehityksen ja low-code -kehityksen yhteys, kysyttiin tätä myös asiakashaastatteluissa. Ketteräkehitys koettiin jokaisessa haastattelussa perusteltuna tapana myös low-code -teknologian kohdalla. Huomattavaa oli kuitenkin yhden haastateltavan tuoma näkökulma, jossa painotettiin ketterän kehityksen hyödyntämistä osittain (H5). Tässä nousi esiin se, että esimerkiksi Scrum-menetelmässä on Scrum

master, jota low-coden kontekstissa haastateltava ei näe tarpeellisena. Toisaalta kappaleessa 3.4 kuvassa 9 esitetty vertailu ketterän kehityksen ja low-code -kehityksen vaiheista tukee myös tätä näkökulmaa.

"Ketterä kehitys toimii erittäin hyvin, toimii isommissa ja pienemmissä projekteissa hyvin. Näen sen hyvänä myös tässä, saadaan nopeasti demoitavaa. Enemmän niin, että saadaan nopeasti jotain esitettävää ja jatkokehitysideoita, että ei mennä hirveän pitkälle väärään suuntaan." (H1)

Haastatteluissa pyrittiin selvittämään, missä roolissa asiakkaan tulisi olla yhteistyökehittämisen mallissa. Jokaisella haastateltavalla nousi asiakkaan roolituksessa selkeästi esiin se, että asiakkaan rooli yhteistyökehittämisen mallissa nähdään määrittelemässä sekä testaamassa ratkaisuita. Toisaalta kaksi haastateltavaa toi esiin myös näkökulman, joissa liiketoiminnan ydinosaaja asiakkaan puolelta voisi olla kansalaiskehittäjä (H3, H5). Tässä roolituksessa korostettiin kuitenkin sitä, että tällöin organisaatiossa tulee olla tarpeeksi resursseja sekä osaamista, jotta tällä mallilla voidaan toteuttaa yhteistyökehittämistä.

"Asiakkaan rooli riippuu vahvasti asiakkaan tietotaidosta ja kyvykkyydestä." (H5)

"Asiakkaan rooli on määrittely ja testaaminen." (H2)

"Se tuo hirveästi lisäarvoa, että yrityksessä on jonkin verran osaamista, jotta osataan paremmin kommunikoida ja keskustella ammattiorganisaation kanssa. Ja myöskään ei tarvitse jokaista pikku muutosta tuoda palveluntarjoajalle, tällä tavoin pystytään järkevöittämään työtä. Mä en usko, että kaikissa yrityksissä on low-code -osaamista, jolloin uskon, että ammattilaisohjelmisto-organisaatiolla on iso rooli, että he pystyvät tekemään kaiken ja yrittävät ymmärtää asiakkaan tarpeita. Tässä on erilaisia yhteistyömalleja ja molemmat ovat oleellisia." (H3)

"Asiakkaana pitäisin enemmän määrittelystä ja testaamisesta, kuin varsinaisesta ohjelmoinnin tekemisestä." (H6)

Palvelut ja palveluiden tuotteistaminen

Teeman avulla analysoitiin konkreettisia tekijöitä, joilla on vaikutusta palveluihin ja miten palvelut koetaan. Teemassa keskityttiin erityisesti työmääräarvioihin, viestintään sekä palautteeseen. Tarkoituksena oli ymmärtää, miten asiakkaat toivoisivat, että nämä tekijät otettaisiin huomioon yhteistyökehittämisen mallissa ja case-organisaation low-code -palveluissa.

Haastatteluissa selvitettiin low-code -teknologiaan liittyvien työmääräarvioiden tarpeita. Minkälaista tietoa asiakkaat haluaisivat nähdä, sillä lähtökohtana nähdään, että tällä on

suuri merkitys tuotteistamiseen. Työmääräarvioiden kohdalla nousi esiin selkeitä asioita ja toisaalta, joltakin osin siihen ei osattu ottaa kantaa. Kyky arvioida kustannuksia työmääräarvion kautta nähtiin asiakkaiden puolelta tärkeänä (H1, H5). Erityisesti nostettiin hankintakustannukset sekä juoksevat kustannukset. Toinen selkeä asia, mikä nähtiin tärkeänä työmääräarvioissa, on eri osakokonaisuudet (H2, H3, H5). Haastateltavien mukaan tässä mallissa erottelu tulisi tehdä ominaisuuksien mukaan mitä ratkaisun tulisi sisältää sekä erottelua datalähteen ja low-coden kanssa. Osa haastateltavista ei ottanut tarkemmin kantaa, mitä haluaisi nähdä työmääräarviossa. Heidän mukaansa raaka-arvio tunneista suurin piirtein riittää (H4, H6).

Viestinnässä ja viestintätavoissa nousi esiin interaktiivisuus. Jokainen haastateltava näki tarpeen palavereille esim. Microsoft Teamsin välityksellä. Haastatteluissa nousi esiin myös livetapaamisten merkitys (H1, H2). Livetapaamista perusteltiin kommunikoinnin ja yhteistyön helpottumisella projektin eri vaiheissa, kuten määrittely- ja käyttöönottovaiheessa. Useamman haastateltavan kohdalla nousi esiin voimakkaita negatiivisia ilmauksia sähköpostin käyttämisestä tämänkaltaisessa projektissa pääviestintätyökaluna (H2, H5, H6). Perusteluita tuli useampia erilaisia, mutta etenkin esiin nousi sähköpostin kankeus ja hitaus. Asiakkaat (H1, H3) kokivat, että projektin jälkeen, kun ratkaisu on todettu toimivaksi, muutostyöt sekä pienet kehitystyöt voidaan pyytää sähköpostin välityksellä. Yhden asiakkaan (H4) mukaan sähköposti toimii pääasiassa hyvin, mutta myös tässä tapauksessa alussa sekä mahdollisesti testauksessa etäpalaverit nähtiin tarpeellisena.

”Mahdollisimman interaktiivinen, eli olisi se esimerkiksi Teams, kommentointi tulee nopeasti, pystyy palottelemaan puolin ja toisin. Eniten inhoan sähköpostia, joka on aina hyvin staattinen.” (H6)

”Varmaan käyttöönotto ja määrittely, jos on yhtään isompia asioita niin hyvä ottaa live tai Teams-miitti, sen jälkeen Teamsissä viesti tai sähköpostiviesti on riittävä: nyt sinne on tehty tällainen ja tällainen muutos, testaatteko. Jos tulee jotain isompaa, tuntuu että ei asia mahdu enää viestiin, sitten Teams-palaveri, että katsotaan yhdessä, että päästään saman ruudun ääreen ja pääsee sen demoamaan.” (H1)

Palautteenannossa nousi esiin juokseva palautteenantaminen ja jatkuva vuorovaikuttaminen (H1, H2, H3, H5). Näillä nähtiin olevan suora yhteys ketterään yhteistyökehittämiseen. Palautteenannossa nähtiin tärkeänä, että kyseessä on kumppanuus palveluntarjoajan kanssa, jolloin palautetta toivotaan molempiin suuntiin (H5, H6). Palautteenannossa korostettiin, että palauteprosessi voi olla tarpeen, etenkin

projektin lopuksi, mutta jatkuva, ketterä palautteen antaminen korostui kuitenkin selvästi tärkeämmäksi.

Haastatteluissa kartoitettiin myös low-codeen liittyvien jatkuvien palveluiden ja tuen tarpeita. Asiakkaan (H2) mukaan jatkuvissa palveluissa palveluntarjoajan puolelta tulisi olla pääkäyttäjää tyyppinen henkilö, joka ymmärtää low-codesta ja toteutetuista ratkaisuista. Tällä varmistetaan kohdennettu ja nopea tuki. Toisaalta toivottiin jatkuvaa tukea teknologian toimivuutta varmistamalla esimerkiksi alustan päivitysten ja muutosten osilta (H6). Kun tulee suurempia ongelmia low-code -ratkaisun toiminnan kanssa, mitä sisäisesti ei pystytä selvittämään, tulisi tässä kohtaa jatkuvien palvelujen ja tuen rooli esiin (H2, H3, H4, H5). Haastatteluista tunnistettiin se, että useampi koki pystyvänsä ratkomaan helppoja ja perinteisiä ongelmia itsenäisesti.

Low-code -palveluihin liittyen asiakas (H1) tunnisti ylipäätään sen, että teknologiasta ja low-code -palveluista tulisi saada yleisnäkyä. Tämä tarkoittaa esimerkiksi hyötyjä, elinkaarta sekä kustannuksia. Myös teknologian osaajien määrällä on merkitystä siihen, halutaanko teknologiaa alkaa hyödyntämään.

Ratkaisu ja muita huomioita

Tämän teeman analyysissä pyritään ymmärtämään, mitä case-organisaation low-code -kehityksessä tehdään jo hyvin ja missä osa-alueissa on kehitettävää. Vastatusten avulla pyritään ymmärtämään, mitä toiminnassa ei tarvitse muuttaa, jotta kehitystä niiltä osin ei tehdä turhaan. Toisaalta pyritään myös ymmärtämään, mitkä ovat case-organisaation kilpailuedut, jota voidaan mahdollisesti hyödyntää palvelun tuotteistamisen tarkoitukseen.

Asiakkaat tunnistivat, että case-organisaation tapauksessa toiminta on ollut suoraviivaista ja nopeaa (H1, H2, H3). Korostettiin myös sitä, että kokonaisuudessaan toiminta on toiminut ja yhteistyö sekä kehittäminen sujuu (H3, H5). Asiakkaan (H2) mukaan reagointi palautteeseen on ollut hyvää ja toiminta on ollut palautteen mukaista.

Asiakkaat tunnistivat kehityskohteita, mutta yhtäläisiä teemoja ei vastauksista pystytty yhdistämään. Oletettavaa on, että myös tässä kohtaa asiakkaan vastauksiin vaikuttaa rooli sekä missä vaiheessa ratkaisu tällä hetkellä on. Kaksi asiakasta ei ottanut kantaa ratkaisuun liittyviin kysymyksiin, sillä joko ratkaisua ei ollut vielä toteutettu tai se oli niin alkuvaiheessa, että kysymyksiin ei voitu ottaa kantaa.

Haastatteluissa nousi esiin myös muita huomioita, joista yksi selkeä tema oli markkinointi low-code -teknologian ympärillä. Asiakkaat (H2, H4, H5) toivoivat palveluntarjoajalta enemmän markkinointia teknologian ympäriltä, mitä sillä voisi tehdä sekä esittelyä ratkaisuista, joita case-organisaatio on jo tehnyt. Erityisesti toivottiin case-

organisaation aktiivista tiedonjakoa, miten jo tehdyillä ratkaisilla voitaisiin helpottaa myös muiden yritysten toimintaa sekä lisätä tietoa low-code -teknologiasta, jotta ymmärretään paremmin, miten teknologiaa voidaan hyödyntää.

6.2 Palveluntarjoajan haastattelut

Case-organisaatiosta haastateltiin kahta ohjelmistokehittäjää, joilla on aiempaa kokemusta low-code -kehityksestä. Molemmilla haastateltavilla on kokemusta useasta eri low-code -ratkaisusta sekä ERP-liiketoiminnasta niin konsultoinnin kuin ohjelmistokehityksen kautta.

Low-code -teknologia ja arvonluonti

Tämän teeman tarkoituksena on selvittää, minkälainen näkemys ja kokemus case-organisaation työntekijöillä on low-code -teknologiasta. Teemassa perehdytään tarkemmin low-code -teknologian tuomaan arvoon ja sen tulevaisuuteen. Näiden avulla voidaan pohtia low-code -teknologian mahdollisuuksia tulevaisuudessa sekä perustella miksi low-code -liiketoimintaan kannattaa käyttää resursseja.

Haastateltavat tunnistivat, että low-code -ratkaisulla voidaan mahdollistaa uusia toimintoja jo olemassa oleviin järjestelmiin, joita niillä ei pystytä tekemään (H7, H8). Low-code -sovellukset tuovat myös mahdollisuuden tehdä puhelimella toimintoja, mikä mahdollistaa eri toimintojen tekemisen missä ja milloin vain (H7).

Low-code -teknologian tulevaisuus nähtiin haastateltavien (H7, H8) mukaan positiivisena. Molempien mukaan low-code -liiketoiminnan voidaan olettaa kasvavan, asiakkailta nähdään selkeästi uusia tarpeita. Haastateltava (H7) nosti kuitenkin markkinoinnin sekä tiedonjaon merkityksen.

”Mä uskon, että low-code -teknologia kasvaa. Varsinkin jos osataan tuoda asiakkaille esiin, että on tämmöisiä mahdollisuuksia. Se, että pystytään toteuttamaan tällaisia yksinkertaisia mobiilisovelluksia, kun tuodaan sitä esille ja markkinoidaan osaamista. Tällöin asiakkaatkin voivat keksiä, että toi voisi täyttää meidän tarpeemme.” (H7)

Yhteistyökehittäminen ja low-code -teknologia

Tämän teeman tarkoituksena on pohtia yhteistyökehittämistä kokonaisuutena. Minkälaisessa roolissa palveluntarjoajan sekä asiakkaiden tulisi toimia, mitä asiakkaalta odotetaan sekä onko ketterän kehityksen piirteet sopiva lähtökohta yhteistyökehittämisen mallille. Teemassa kartoitettiin myös sitä, miten yhteistyökehittäminen ymmärretään ja mitä se tarkoittaa case-organisaation työntekijöille.

Case-organisaation työntekijöiden vastauksista tunnistettiin selkeä yhteinen linja, miten yhteistyökehittäminen ymmärretään. Haastateltavien mukaan yhteistyökehittäminen nähdään asiakkaan osallistamisena koko kehitysprosessiin (H7, H8). Yhteistyökehittämisellä nähdään olevan vaikutusta lopputulokseen positiivisessa mielessä (H8). Myös ketterän kehityksen hyödyntämisestä low-code -kehityksestä ollaan samaa mieltä. Molemmat haastateltavat toteavat, että ketterä kehitys on heidän mukaansa ehdottomasti keino tehdä low-code -ratkaisuja. Haastateltavan (H7) mukaan low-code -alustat myös tukevat ja ohjaavat tähän suuntaan esimerkkinä alustan mahdollistama helppo demoaminen asiakkaille.

”Asiakas otetaan koko kehitysprosessiin mukaan jatkuvasti. Ei niin, että asiakkaalta tulee määräys ja lopputulos näytetään asiakkaalle. Iteratiivisesti kehitetään ja asiakas pääsee koko ajan testaukseen mukaan. Päästään lopputulokseen ehkä helpommin ja nopeammin ja toivottavasti tarkemmalla tasolla siihen, mitä asiakas oikeasti haluaa.” (H7)

Yhteistyökehittämisen roolituksessa molemmat haastateltavat näkivät roolitukset hyvin samalla tavalla. Haastateltavan (H8) mukaan palveluntarjoajan puolelta tarvitaan 1-3 henkilöä. Toisen haastateltavan (H7) mukaan palveluntarjoajan puolelta olisi hyvä olla useampi henkilö mukana projektissa etenkin, jos asiakas ei ole low-code -kehityksessä teknisesti mukana. Haastatteluista tunnistetaan, että roolit palveluntarjoajalta määritellään kehittäjäksi sekä konsultiksi (H7, H8). Kehittäjä voidaan jakaa vielä datalähteen puolen kehittäjäksi esimerkkinä case-organisaation tapauksessa ERP sekä low-code -kehittäjäksi (H8). Konsultti toimii asiakkaan pääyhteyshenkilönä, jonka tehtävänä olisi miettiä liiketoiminnan näkökulmaa sekä tarpeita (H7, H8). Molemmat kuitenkin tunnistavat, että kehittäjä voi olla myös suoraan asiakkaan yhteyshenkilö. Haastateltava (H8) tunnistaa, että asiakkaalla tulisi olla yhteistyökehittämisen mallissa 1-2 roolia: puolitekninen sekä loppukäyttäjätyyppinen, jonka vastuulla on testaaminen. Haastateltava (H7) tuo esiin näkökulman, että liiketoiminnan tarpeita tulisi olla miettimässä myös case-organisaation puolelta henkilö, jotta ymmärretään varmasti liiketoiminnan näkökulma.

Asiakkaan puolelta toivotaan selkeää määrittystä (H7, H8). Haastateltava (H8) lisää myös yhteistyön visuaalisessa suunnittelussa sekä testausvastuun, joka lopussa tulisi olla asiakkaalla. Asiakkaalta toivotaan myös teknistä valvetuneisuutta, sillä projektin aikana voidaan joutua selittämään asioita hyvinkin teknisellä tasolla (H7). Arvonluonnin näkökulmasta yhteistyökehittäminen nähdään osallistavana ja testauksen näkökulmasta virheiden tunnistaminen aikaistuu.

Palvelut ja palveluiden tuotteistaminen

Teeman avulla analysoitiin konkreettisia tekijöitä, joilla on vaikutusta palveluihin. Teemassa keskityttiin erityisesti työmääräarvioihin, viestintään sekä koulutukseen. Tarkoituksena oli ymmärtää sisäisen sekä ulkoisen tuotteistamisen näkökulmista aihealuetta.

Low-code -palveluiden ympärillä tunnistettiin, että palveluntarjoajalla tulisi olla tuotteistettu kokonaisratkaisu, jota voidaan tarjota asiakkaille (H7, H8). Haastatteluissa tunnistettiin myös se, että teknisestä näkökulmasta vikapalvelu pitäisi pystyä tarjoamaan (H8). Yhtenä teemana esiin nousi dokumentoinnin merkitys, jotta tiedetään mitä järjestelmää asiakas käyttää (H7, H8). Tähän liittyy tiiviisti myös versionhallinta, jota molemmat haastateltavat korostivat. Koulutuksen teemassa nousi esiin asiakkaan tietoisuuden lisääminen, mitä low-code -teknologia tarkoittaa (H7, H8). Samalla korostettiin markkinoinnin näkökulmaa. Haastateltava (H7) nosti esiin myös sen, että ratkaisun käyttökoulutus on palveluntarjoajan tehtävä.

Viestinnästä ja viestintätyökaluista haastateltavilla oli hyvin yhtenäiset näkemykset. Teams toimii pääasiallisena työkaluna, jossa voidaan demota ratkaisua eri vaiheissa sekä iteratiivisesti kehittää ja kommunikoida asiakkaan kanssa. Sisäisesti työtehtävien jako tulee toteuttaa tehtävähallintaohjelmistolla. Myös työmääräarviossa haastateltavat näkivät yhtenäisesti toteutuksen. Molempien mukaan työmääräarviossa tulisi näkyä datalähteestä tarvittavat asiat ja käytettävät resurssit, low-code -alustaan liittyvät toimenpiteet sekä testaamiseen käytettävät resurssit (H7, H8).

”Varmaan joku Teams-kanava olisi helppo nopeaan viestintää, kun sitä iteratiivisesti kehitetään, jos siellä on useampi henkilö. Kyllä se olisi parempi kuin edestakainen sähköpostien pallottelu. Kyllä jotkut viikoittaiset tai joka toinen viikko tai mikä se sykli olisikin niin ottaa yhteinen Teams-palaveri, just sen takia, että siinä pystyy demoamaan sitä toiminnallisuutta.” (H7)

Ratkaisu ja muita huomioita

Tämän teeman alla pyrittiin ymmärtämään erityisesti, mitä palveluntarjoaja voisi kehittää ja mitä toisaalta ei tarvitse kehittää. Tässäkin teemassa nousi samoja havaintoja molempien haastateltavien toimesta. Molemmat ovat olleet tyytyväisiä toteutettujen ratkaisujen yhteistyökehittämiseen sekä tarpeisiin on pystytty vastaamaan nopeasti ja asiakkaat ovat olleet tyytyväisiä ratkaisuihin. Haasteena on ollut se, että kehitystyö on ollut nuorta eikä ratkaisuista ole ollut aiempaa kokemusta.

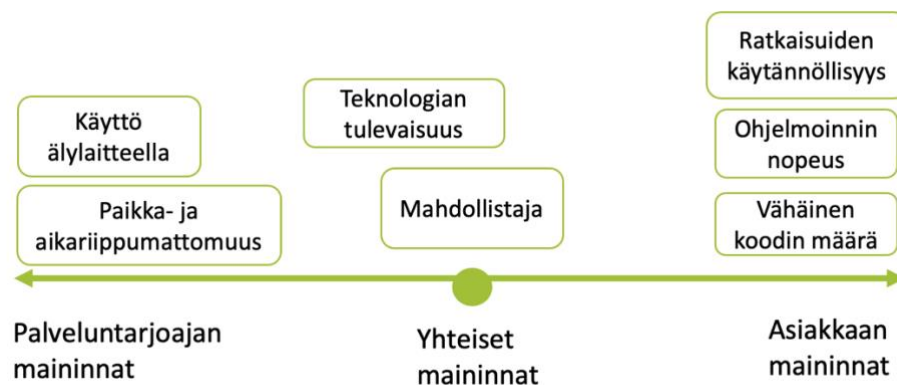
Haastateltava (H7) korostaa vielä erityisesti low-code -alustaan liittyvää ylläpitoon ja hallintaan panostamista. Tällä tarkoitetaan käyttöoikeuksiin sekä lisensseihin

perehtymistä, jotta sovellusten testaaminen ja käyttö näyttäytyy myös asiakkaille selkeältä. Myös dokumentaation merkitystä korostetaan vielä uudelleen, kuten jo aiemmin tuotiin esille.

6.3 Yhteenveto haastatteluista

Tässä kappaleessa esitetään yhteenvetona tunnistettuja teemoja sekä alateemoja, joita asiakkaiden sekä palveluntarjoajien haastatteluista löydettiin. Kappaleen tarkoituksena on helpottaa lukijaa ymmärtämään molempien näkökulmia ja haastatteluista tunnistettuja yhtäläisyyksiä. Toisaalta tällä kappaleella helpotetaan myös tutkijaa muodostamaan päätelmiä kirjallisuuden sekä empirian väliltä.

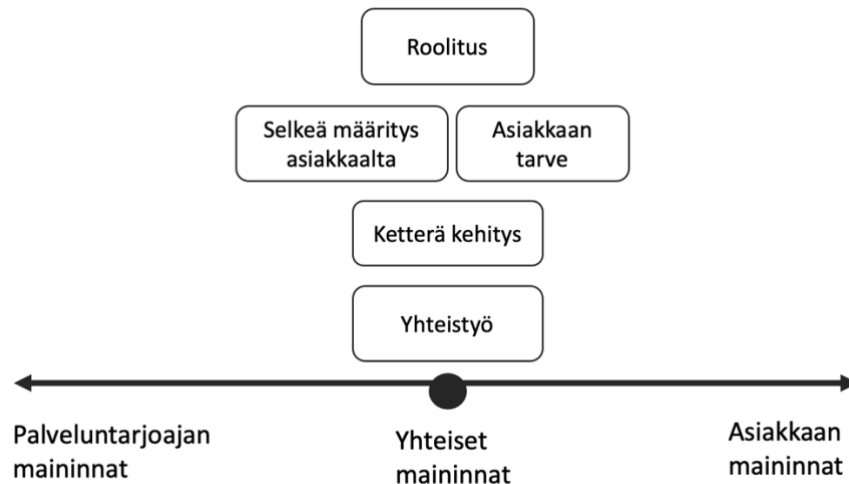
Ensimmäisen teeman eli low-code -teknologian ympäriltä tunnistettiin, että teknologian tulevaisuus mainittiin sekä asiakkaiden, että case-organisaation työntekijöiden puolesta (H3, H7, H8). Myös muiden asiakkaiden osalta voidaan olettaa heidän uskovan teknologian tulevaisuuteen, koska he ovat investoineet low-code -ratkaisuihin. Kuvaan 22 on koottu mainintoja alateemoista.



Kuva 22: Haastatteluista tunnistetut alateemat low-coden ja arvonluonnin teemasta.

Kuten kuvasta 22 huomataan low-coden ympäriltä tunnistettiin selkeitä alateemoja. Yhteisiä mainintoja esiintyi teknologian tulevaisuuden sekä mahdollistajan osilta. Low-code -teknologian tunnistettiin mahdollistajaksi, sillä sen avulla on sekä asiakkaiden, että case-organisaation työntekijöiden mukaan pystytty kehittämään erilaisiin liiketoimintaprosesseihin ratkaisuita. Toisaalta tunnistettiin myös selkeitä alateemoja, joita mainittiin vain toisen haastatteluryhmän puolesta. Kuvassa 22 vasemmalla puolella on kuvattu alateemat, joita mainittiin palveluntarjoajan näkökulmasta sekä oikealla puolella, jota mainittiin asiakkaiden haastatteluissa. Eri haastatteluryhmille oli hieman eri haastattelurunko, jolla voidaan olettaa olevan vaikutusta mainittuihin asioihin.

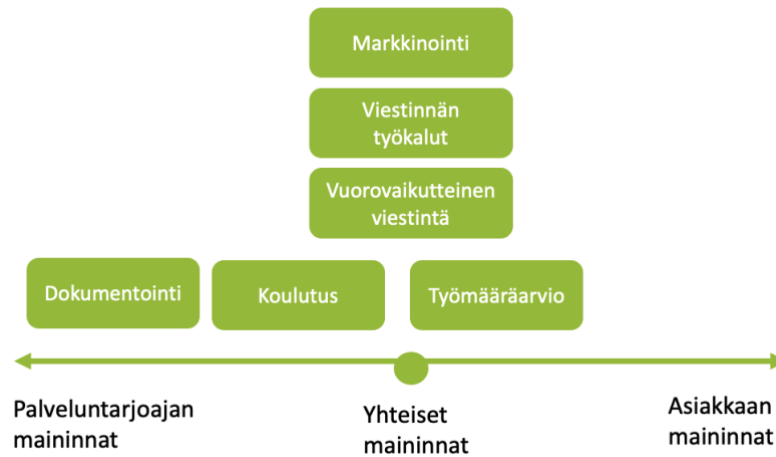
Yhteistyökehittämisen teemasta esiin nousi hyvin samankaltaisia asioita eri haastateltavien puolesta. Kuvaan 23 on koottu sekä case-organisaation työntekijöiden, että asiakkaiden mainintoja, jolla kuvataan näkemyksiä siitä, miten yhteistyökehittäminen ymmärretään ja minkälaista sen tulisi olla.



Kuva 23: Haastatteluista tunnistetut alateemat yhteistyökehittämisen ja low-code -teknologian teemasta.

Kuvasta huomataan, että maininnat on karkeasti koottu alateemojen alle. Case-organisaation työntekijät sekä asiakkaat ymmärsivät sekä toivoivat yhteistyökehittämiseltä hyvin pitkälti samoja asioita. Kuvaan 23 nostetut alateemat olivat selkeimmät kokonaisuudet, jotka haastatteluissa nousi esiin. Yhteistyön alateemassa korostettiin kumppanuutta, vuorovaikutusta ja asiakkaiden osallistumista prosessiin. Roolituksessa nähtiin molempien haastatteluryhmien puolesta samankaltaisuuksia. Selkeä tunnistettu alateema oli asiakkaan tarve, joita asiakkaat mainitsivat. Case-organisaation työntekijät mainitsivat selkeän määrittämisen tarpeen. Voidaan todeta, että molemmat haastatteluryhmät tukivat ajatusta siitä, että asiakkaan tarpeeseen kehitetään ratkaisua. Myös ketterän kehityksen ja yhteistyön merkitystä korostettiin molempien haastatteluryhmien puolesta.

Palveluiden ja tuotteistamisen teemasta tunnistettiin myös selkeitä alateemoja ja mainintoja, joista oltiin samaa mieltä. Kuvaan 24 on koottu yhteen selkeät alateemat, joita haastatteluista tunnistettiin.



Kuva 24: Haastatteluista tunnistetut alateemat palveluiden ja tuotteistamisen teemasta.

Kuva 24 osoittaa sen, että haastatteluissa keskityttiin konkreettisiin asioihin, joilla nähdään olevan vaikutusta palveluihin ja koettuun laatuun. Dokumentoinnin merkitys korostui case-organisaation työntekijöiden haastatteluissa. Koulutus mainittiin molempien haastatteluryhmien vastauksissa, mutta case-organisaation työntekijöiden puolesta sen merkitystä korostettiin enemmän. Osa asiakkaista korosti myös koulutuksen merkitystä. Asiakkaiden puolelta korostui koulutuksen kohdalla enemmänkin markkinointi. Koulutusta toivottiin markkinointi mielessä, jossa kerrottaisiin mitä teknologialla voi tehdä ja minkälaisia ratkaisuja on jo toteutettu. Viestintään liittyvät maininnat olivat hyvinkin yhdenmukaisia haastatteluryhmien välillä.

7. PÄÄTELMÄT

Tässä luvussa tehdään päätelmät kirjallisuudesta sekä empiirisestä osuudesta tutkimuksen tavoitteiden mukaisesti. Kappaleessa 7.1 luodaan yhteistyökehittämisen malli low-code -teknologialle kirjallisuuden ja empirian perusteella. Kappaleessa 7.2 arvioidaan yhteistyökehittämisen mallin hyödyntämistä low-code -palveluiden tuotteistamisessa. Kappaleessa otetaan kantaa konkreettisiin käytännön asioihin, joilla nähdään olevan vaikutusta palveluihin ja palveluiden laatuun. Lopuksi käsitellään heikon markkinatestin tuloksia ja tehdään päätelmiä, kuinka mallin luonnissa onnistuttiin sekä nousiko testissä jatkokehitystarpeita.

7.1 Yhteistyökehittämisen malli low-code -teknologialle

Yksi tutkimuksen tavoite oli selkeyttää case-organisaation sisäistä toimintaa low-code -kehityksen parissa. Tutkimusta lähdettiin tekemään, koska oli tunnistettu, että toimintatavoissa oli kehityskohteita. Case-organisaatiolla ei ollut selkeää dokumentointia siitä, millaisia eri rooleja low-code -kehitykseen tulisi osallistua. Myöskään vastuunjako sekä tiimin jäsenten osallistuminen eri kehitysvaiheisiin ei ole ollut dokumentoitu. Tästä syystä kirjallisuuden sekä empiirisen tutkimuksen avulla pyrittiin selvittämään vastuita ja rooleja eri low-code -kehityksessä. Tässä tutkimuksessa muodostetulla yhteistyökehittämisen mallilla otetaan kantaa siihen, millaisella roolituksella yhteistyökehittämistä tulisi tehdä sekä mitä vaiheita low-code -kehitys sisältää. Malli on muodostettu kirjallisuuden ja empirian vuoropuhelun kautta.

Kirjallisuudesta tunnistettiin kolme eri roolia, jotka ovat kansalaiskehittäjä, sovelluskehityksen ammattilainen sekä loppukäyttäjä. Iho et al. (2021) määrittelee kansalaiskehittäjät LCDP-kehittäjiksi (low-code development platform). Kappaleessa 3.3 esiteltiin Boyer et al. (2021) kehittämä yhteistyötiimin malli. Empiirisen tutkimuksen pohjalta tässä mallissa tunnistettiin kuitenkin puutteita, joiden pohjalta rooleja ja vastuita kehitettiin. Empiirisessä osuudessa tunnistettiin myös neljäs rooli eli konsultti. Yhdessä kirjallisuuden ja empiirisen tutkimuksen myötä kehitettiin malli, jossa kuvataan yhteistyötimiin kuuluvia ammattilaisia (kuva 25). Mallissa otetaan myös kantaa siihen, onko roolin henkilö asiakkaan vai palveluntarjoajan työntekijä.

Taulukko 7: Yhteistyötiimin kokoonpanon eri mahdollisuudet.

		Sovelluskehityksen				Työntekijöiden määrä (kpl)
		Kansalaiskehittäjä	ammattilainen	Konsultti	Loppukäyttäjä	
Vaihtoehto 1	Asiakas	x			x	1
	Palveluntarjoaja		x	x		1
Vaihtoehto 2	Asiakas				x	1
	Palveluntarjoaja	x	x	x		1
Vaihtoehto 3	Asiakas				x	1
	Palveluntarjoaja	x	x	x		2
Vaihtoehto 4	Asiakas				x	1
	Palveluntarjoaja	x	x	x		3
Vaihtoehto 5	Asiakas	x			x	2
	Palveluntarjoaja	x	x	x		3

Yhteistyötiimin kokoonpanot on jaettu viiteen eri vaihtoehtoon taulukon 7 mukaisesti, joka on koottu empiirisen tutkimuksen perusteella. **Vaihtoehdossa 1** tiimissä on kaksi jäsentä, yksi asiakkaalta ja yksi palveluntarjoajalta. Asiakkaan rooli on toimia kansalaiskehittäjänä sekä loppukäyttäjän rooleissa. Tässä tapauksessa palveluntarjoajalta yksi työntekijä toimii sekä konsulttina, että sovelluskehittäjän rooleissa. Vaihtoehdossa 1 palveluntarjoajan rooli on olla tukemassa niin tietojen mallinnukseen kuin low-code -ratkaisun tekniseen näkökulmaan liittyvissä asioissa. Sovelluskehityksen ammattilaisen tehtävänä on mahdollistaa low-code -kehitys esimerkiksi huolehtimalla ulkoisten järjestelmien integroinnista. Case-organisaation tapauksessa tämä voisi tarkoittaa ERP-järjestelmän integroimista low-code -ratkaisuun.

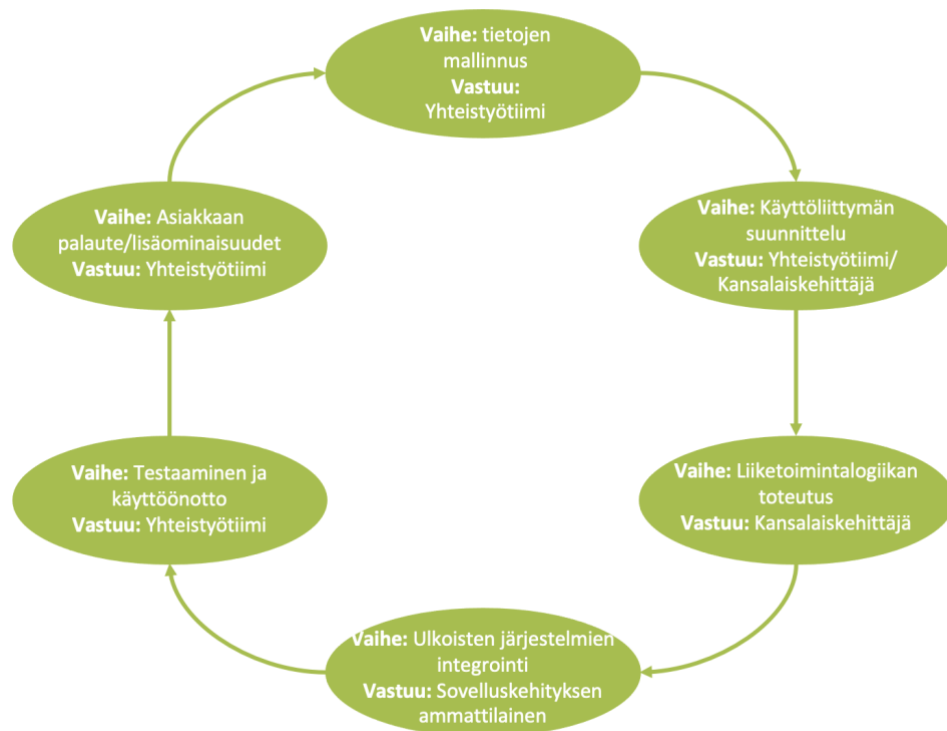
Vaihtoehdossa 2 palveluntarjoaja tarjoaa kaiken teknisen tuen ja toteutuksen, kun asiakkaan vastuulle jää määrittellä tarve sekä testata ratkaisua. Myös tässä vaihtoehdossa yhteistyötiimi koostuu kahdesta henkilöstä. Palveluntarjoajan puolelta yksi henkilö toimii sovelluskehittäjän, konsultin sekä kansalaiskehittäjän rooleissa. Henkilöllä on tällöin suuri vastuu paitsi teknisestä toteutuksesta, myös asiakasviestinnästä, määrittelystä sekä liiketoimintaymmärryksestä.

Vaihtoehdossa 3 tekninen toteutus on edelleen täysin palveluntarjoajan vastuulla, kun asiakkaan rooli on määrittellä tarve sekä testata. Tässä vaihtoehdossa yhteistyötiimi koostuu kolmesta henkilöstä, mikä tarkoittaa, että palveluntarjoajan puolelta tiimiin osallistuu kaksi henkilöä. Palveluntarjoajan puolelta yksi henkilö on joko konsultti sekä kansalaiskehittäjä, kun toisen rooli on toimia sovelluskehittäjän roolissa. Toinen vaihtoehto on, että sovelluskehittäjä sekä kansalaiskehittäjä ovat sama työntekijä. Boyer et al. (2021) mukaan lähtökohtana voidaan pitää sitä, että kansalaiskehittäjät eli yrityksen liiketoiminnan keskiössä työskentelevät työntekijät tekevät sovellusratkaisut. Koska kirjallisuus tukee näkökulmaa, jossa liiketoiminnan osaaja toimii kansalaiskehittäjän roolissa, voidaan suositella vaihtoehtoa, jossa konsultti toimii samanaikaisesti kansalaiskehittäjänä.

Vaihtoehto 4 eroaa edellä esitellystä vaihtoehdosta 3 niin, että yhteistyötiimi koostuu neljästä henkilöstä, joista kolme osallistuu palveluntarjoajan puolelta. Asiakkaan rooli ei eroa vaihtoehdosta 3, mutta palveluntarjoajalta on jokaiseen rooliin oma työntekijä.

Vaihtoehdossa 5 yhteistyötiimi koostuu viidestä henkilöstä, joka on kaikista suurin kooltaan. Tämä eroaa vaihtoehdosta 4 niin, että asiakkaan puolelta osallistuu yhden henkilön sijaan kaksi. Tämä tarkoittaa, että asiakas tuo mukaan myös teknistä osaamista, joko osallistuen myös low-code -kehitykseen tai teknisemmästä näkökulmasta määrittelyyn ja testaukseen.

Kuvassa 25 ja taulukossa 7 otettiin jo kantaa siihen, mitä rooleja tiimissä tulisi olla. Tarkastellaan siis seuraavaksi low-code -kehityksen vaiheita ja minkälaiset vastuut eri vaiheissa kenelläkin yhteistyötiimin jäsenellä tulisi olla. Vastuunjakojen sekä roolien kohdalla oli case-organisaatiossa tunnistettu haaste. Ei ole ollut selkeitä vastuunjakoja kenen osaajan tulisi tehdä mitään sekä osallistua mihinkin vaiheeseen. Tämän mallin tavoitteena on siis selkeyttää case-organisaation työntekijöille, keiden pitää osallistua mihinkin vaiheeseen ja toisaalta selkeyttää myös asiakkaille low-code -ratkaisujen kehitystä. Kirjallisuudesta tunnistetaan Al Alamin et al. (2021) esittelemä malli, joka kuvaa low-code -kehityksen vaiheet. Tämän mallin käyttämistä tukee myös empiriassa tunnistetut huomiot, sillä jokainen haastateltava puolsi ketterää kehitysmenetelmää myös low-code -kehityksessä. Kuvassa 26 kuvataan yhteistyötiimin roolien vastuut eri low-code -kehityksen vaiheissa. Kuvassa on hyödynnetty kappaleessa 3.4 esiteltyä mallia, joka on mukailtu lähteestä Al Alamin et al. (2021). Kuvan 26 tavoitteena on esittää selkeät vastuujat eri rooleille sekä milloin yhteistyötiimin tulee toimia yhdessä. Tutkimuksessa pyrittiin selvittämään myös työkaluja, miten tiimin tulee mm. viestiä. Nämä päätelmät esitellään myöhemmin luvussa 7.2.



Kuva 26: Yhteistyötiimin roolien vastuut low-code -kehityksen eri vaiheissa.

Kuten kuvasta 26 nähdään, yhteistyötiimillä on suuri rooli eri low-code -kehityksen vaiheissa. Kirjallisuuden ja empiirisen osuuden perusteella tunnistettiin, että tietojen mallinnuksessa tarvitaan tiimin osaamista. Tietojen mallinnuksella tarkoitetaan entiteettien, suhteiden, rajoitusten ja riippuvuuksien luomista yrityksen tietojen välille (Sahay et al. 2020). Lähtökohtana on, että asiakkaan tarpeen mukaan lähdetään ideoimaan ja toteuttamaan ratkaisua niin, että palveluntarjoaja voi tukea, ohjata sekä auttaa ratkaisun toteutuksessa. Empiirisen tutkimuksen tuloksena tunnistettiin, että käyttöliittymän tekemisen vastuu on yhteistyötiimillä sekä kansalaiskehittäjällä. Sen suunnittelu tulisi hoitaa yhteistyössä, mutta varsinaisen toteutuksen tekee kansalaiskehittäjä eli low-code -kehittäjä. Empiria sekä kirjallisuus tukevat näkemystä siitä, että liiketoimintalogiikan toteutuksesta vastaa kansalaiskehittäjä, kun taas ulkoisten järjestelmien integroinnista sovelluskehityksen ammattilainen. Empiirisen tutkimuksen perusteella tehtiin johtopäätös, että testaamisesta ja käyttöönotosta on vastuussa yhteistyötiimi. Palveluntarjoajan puolelta tulisi tulla tekninen näkökulma testaamiselle ja käyttöönotolle. Myös asiakkaalla on vastuu testaamisesta ja ratkaisun toimivuuden hyväksymisestä. Tässä kohdassa kuitenkin testaamisen ei tarvitse tapahtua ns. saman pöydän ääressä vaan se voidaan toteuttaa erikseen. Huomiot sekä testaamisen ja

käyttöönoton tuki tulee kuitenkin suorittaa yhteistyössä. Tällä tarkoitetaan sitä, että tarvittaessa palveluntarjoajalla tulee olla valmius asiakastestauksen tukemiseen. Palautteen ja lisäominaisuuksien vaiheessa nähdään, että yhteistyötiimin tulisi olla koolla. Tässä kohtaa voidaan jatkaa ratkaisun sekä yhteistyön kehittämistä palautteen mukaisesti tai todeta ratkaisu valmiiksi. Empiirisessä osuudessa korostettiin jatkuvaa palautetta ja palautteenanto nähtiin ylipäätään erittäin tärkeänä. Täten mallissa halutaankin korostaa palautteenannon mahdollisuus muissakin vaiheissa kuin palautteen ja lisäominaisuuksien kohdalla.

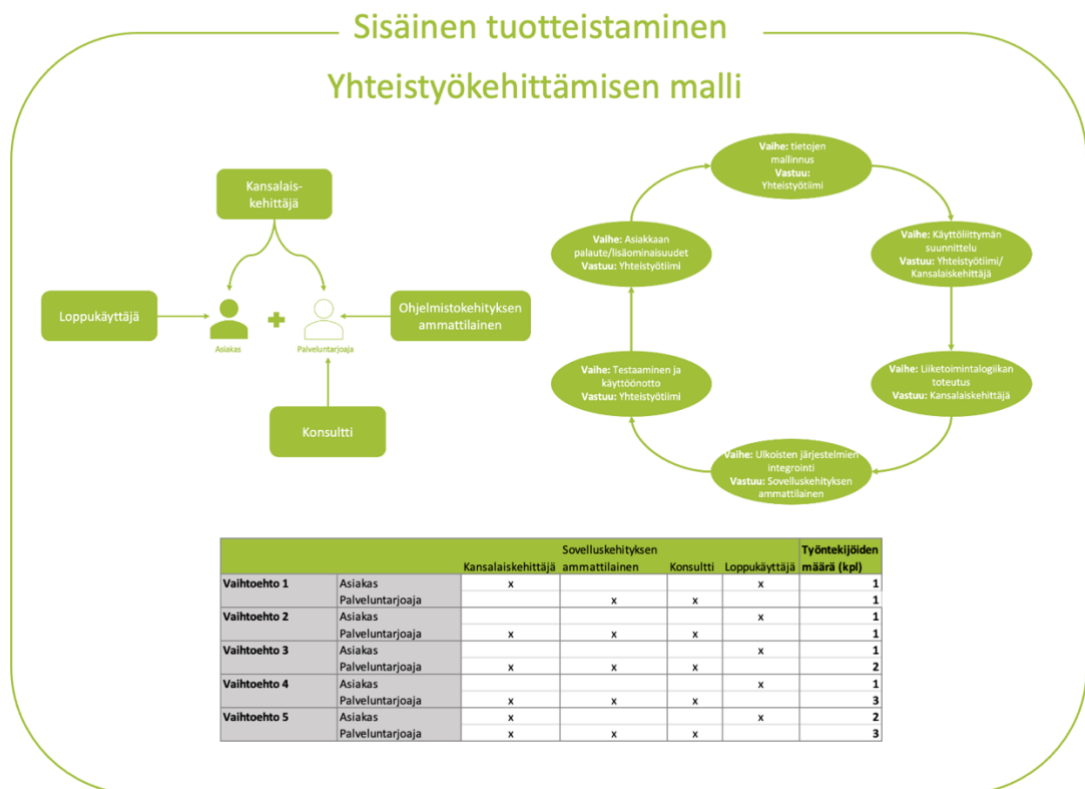
Pohditaan seuraavaksi esiteltyä yhteistyökehittämisen mallia arvon yhteisluonnin näkökulmasta. Kirjallisuudesta tunnistettiin kuvassa 11 esitetty Grönroos-Voima -malli, jossa kuvataan arvon yhteisluonnin eri alueita sekä asiakkaan ja toimittajan rooleja (Grönroos 2017). Tässä kappaleessa esitetyssä kuvassa 25 nähdään, että yhteistyötiimi toimii vahvasti yhteisellä alueella luoden todellista arvoa. Arvoa syntyy yhteistyön myötä ja asiakas osallistuu yhteistuottajana prosessiin. Taulukossa 7 on esitetty eri vaihtoehtoja, miten yhteistyötiimi voidaan koota. Vaihtoehtoissa osassa toimitaan enemmän asiakkaan alueella ja osassa enemmän toimittajan alueella. Jokaisessa vaihtoehdossa työskennellään kuitenkin myös yhteisellä alueella eli toiminta ei rajoitu ainoastaan toisen ryhmän arvonluonnin alueelle. Esimerkiksi vaihtoehdoissa 1 ja 5 toimitaan eniten asiakkaan alueella, koska kansalaiskehittäjän rooli on kokonaan tai osittain asiakkaan vastuulla. Vaihtoehtoissa 2, 3 ja 4 toimitaan enemmän yhteisellä alueella, jopa toimittajan puolelle painottuen, sillä palveluntarjoajalla on tällöin täysi vastuu teknisestä toteutuksesta.

7.2 Yhteistyökehittämisen mallin hyödyntäminen low-code -palveluiden tuotteistamisessa

Kuten kirjallisuudesta aiemmin tunnistettiin, tuotteistaminen voidaan jakaa sisäiseen sekä ulkoiseen tuotteistamiseen. Sisäinen tuotteistaminen kertoo organisaation kyvystä luoda ja ulkoinen tuotteistaminen kertoo kyvystä myydä. (Hempe et al. 2015) Kirjallisuudessa tunnistettiin tuotteistamisen asteet, joka lähtee sisäisten työmenetelmien tuotteistamisella (Sipilä 1995). Tuotteistamisen asteet esitellään tarkemmin kappaleessa 4.2 kuvassa 14. Tarkastellaan siis ensin yhteistyökehittämisen mallia ja sisäistä tuotteistamista, jonka jälkeen pohditaan mallin tuomia mahdollisuuksia myös ulkoiseen tuotteistamiseen.

Sisäinen tuotteistaminen pyrkii tehostamaan palvelun kehittämistä, hankintaa, testausta, sekä prosessin laatua (Hempe et al. 2015). Sisäinen ymmärrys kasvaa, mikä mahdollistaa asiakkaille paremman palvelun, kun työntekijällä on tiedot selittää palvelun

sisältö selkeästi (Harkonen 2021; Sipilä 1995). Sisäisessä tuotteistuksessa ydinidea on, että samaa asiaa ei tehdä useaan kertaan (Sipilä 1995). Kappaleessa 7.1 esitetyt kuvaukset luovat yhdessä yhteistyökehittämisen mallin, joka auttaa low-code -ratkaisun kehittämisessä, testauksessa sekä parantaa prosessin laatua. Mallissa määritellään yhteistyötiimin vastuut ja roolit sekä otetaan kantaa eri mahdollisuuksista luoda yhteistyötiimi. Mallissa esitellään myös kirjallisuudesta tunnistetut low-code -kehityksen vaiheet ja niihin yhdistetyt vastuut. Kuvaan 27 on tiivistetty, mitä osuuksia yhteistyökehittämisen malli sisältää ja kuinka se nähdään sisäisen tuotteistamisen lähtökohtana.



Kuva 27: Tiivistetty kuvaus yhteistyökehittämisen mallista sekä sen roolista sisäisessä tuotteistamisessa.

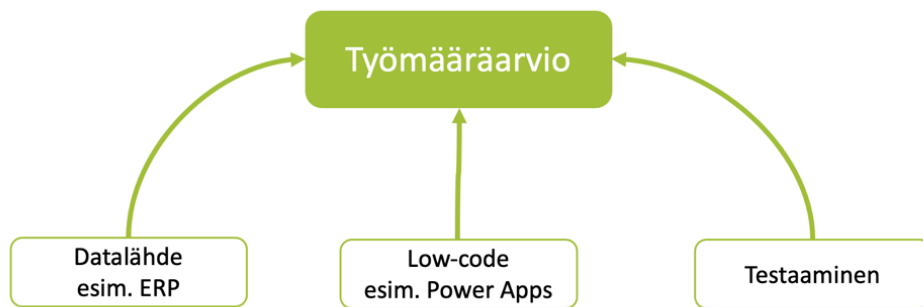
Tällä mallilla voidaan selkeyttää low-code -kehitystä, sillä rooleilla voidaan ohjata työntekijöiden tekemistä, kun jokaisella on omat vastuut ja työtehtävät tiedossa. Selkeästi määritetyt vaiheet ja niissä tarvittava osaaminen auttaa myös asiakkaiden ohjaamisessa. Tämä oli yksi haastatteluissa noussut tekijä, jota yhteistyökehittämisessä asiakkaiden puolelta toivottiin. Taulukossa 7 esitetyt eri kokoonpanot auttavat palveluntarjoajaa sekä asiakasta resursoinnissa, koska mallilla voidaan esittää selkeästi, kuinka monta työntekijää tiettyyn vastustettuun rooliin tarvitaan. Oletuksena on, että työntekijöiden tarvittavaa määrää ohjaa ratkaisun laajuus ja haastavuus. Tällöin low-

code -ratkaisun kehitykseen ei käytetä turhia resursseja niin case-organisaation kuin asiakkaidenkaan puolelta. Yhteistyökehittämisen mallin avulla voidaan myös kehittää case-organisaatiossa tunnistettua haastetta low-code -kehityksen vaiheiden ja vastuiden osilta. Tämän myötä tiedetään, kenen tulee osallistua mihinkin vaiheeseen, mikä vähentää päällekkäisyyksiä ja asioiden tekemistä useaan kertaan.

Kirjallisuudesta tunnistettiin, että asiantuntijapalveluissa voidaan nähdä haasteita viestinnän, edistämisen sekä hinnoittelun näkökulmista. Myös case-organisaatiossa ennen tutkimusta oli tunnistettu, että näihin osa-alueisiin tarvitaan selkeämpää prosessia, jotta sisäinen työnteko olisi tehokkaampaa ja asiantuntijuutta low-coden kontekstissa voitaisiin parantaa. Kirjallisuudesta tunnistettujen tekijöiden sekä case-organisaatiossa tunnistettujen tarpeiden perusteella empiirisessä osuudessa kiinnitettiin huomiota hyvin konkreettisiin tekijöihin, joilla nähdään olevan vaikutusta palvelun laatuun sekä miten kehitysprosessi koetaan.

Sipilän (1995) mukaan asiantuntijapalveluiden tuotteistamiseen keskiössä voidaan nähdä vuorovaikutusprosessi. Prosessikonsultointi eli kyky lähestyä asiakasta ja auttaa löytämään oikea diagnoosi nähdään jopa tärkeämpänä tekijänä palveluissa kuin selkeä tekninen ratkaisu ongelmaan. Tutkimuksen teorian perustella todettiin, että vuorovaikutus voidaan nähdä projektin voimalähteenä (Anttonen 2003; Pikkarainen et al. 2008). Myös Bradyn & Croninin (2001) luomassa palvelun laatumallissa korostettiin vuorovaikutuksen merkitystä (kappale 4.1, kuva 12). Tutkimuksen empiirisessä osuudessa tunnistettiin, että viestinnän näkökulmasta asiakkailta sekä palveluntarjoajalla oli hyvin samanlaiset tarpeet. Viestinnässä ja työkaluissa korostui helppous ja interaktiivisuus. Tuotteistamisen näkökulmasta nähdään, että viestinnällä on merkitystä sekä sisäisessä, että ulkoisessa tuotteistamisessa. Tutkimuksessa tehdyt havainnot viestinnän ja vuorovaikutuksen osilta auttaa case-organisaatiota sekä epämuodollisessa, että muodollisessa viestinnässä.

Tutkimuksen empiirisessä osuudessa selvitettiin sekä asiakkaiden, että palveluntarjoajan näkökulmista, minkälainen erottelu työmäärästä tulisi tehdä. Tätä selvitettiin siitä syystä, koska case-organisaatiossa sekä kirjallisuudessa on tunnistettu, että työmääräarvioiden tekeminen nähdään haasteellisena, kun historiadataa ratkaisun toteutuksen ympäriltä ei ole. Näkökulmana empiirisessä osuudessa oli seuraava: mitä asiakkaat haluavat nähdä ja toisaalta mitä työmääräarvioon on hyvä sisällyttää, joka helpottaa myös sisäisiä prosesseja. Ajatuksena oli, että hyvin tehdyllä työmääräarviolla voidaan ohjata myös palveluntarjoajan tekemistä, kuten työn organisointia ja aikataulutusta. Empiirisen osuuden perusteella luotiin malli, jota voidaan hyödyntää työmääräarvioiden tekemisessä.



Kuva 28: Malli low-code -ratkaisun työmääräarvion luomiseen.

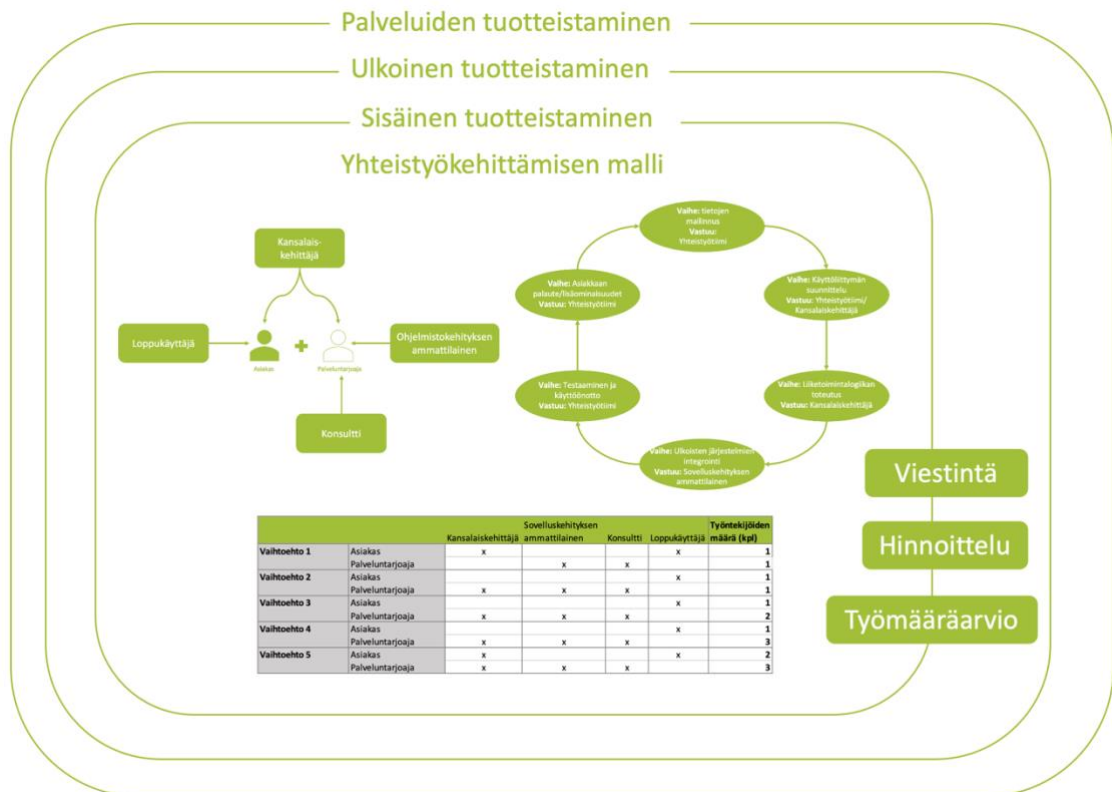
Työmääräarviossa tulisi näkyä erottelu datalähteen, low-coden sekä testaamisen välillä. Datalähteen kohdalla voidaan viitata sovelluskehityksen ammattilaisen tekemään työmäärään, low-coden kohdalla tarkoitetaan low-code -kehitykseen menevää työaika ja testaamisen kohdalla testaamiseen menevää työaika. Empiirisessä osuudessa ei käynyt tarkemmin ilmi, että mitä muuta halutaan nähdä. Low-code -kehityksen vaiheista nähdään kuitenkin, että myös konsultilliset työmäärät tulisi huomioida. Yhteistyökehittämisen malli tukee myös työmääräarvioiden luomista, sillä sen avulla pystytään määrittelemään selkeät roolit ja vastuut sekä työtavat. Kun jokaisella yhteistyötiimin jäsenellä on tiedossa omat työtehtävät, päällekkäisiltä tehtäviltä vältytään sekä aikataulus onnistuu todennäköisesti paremmin.

Tarkastellaan seuraavaksi kolmatta tunnistettua haastetta eli hinnoittelua low-coden kontekstissa. Kirjallisuudesta tunnistettiin eri hinnoittelumalleja, jotka esitellään tarkemmin kappaleessa 4.4. Empiirisessä osuudessa hinnoitteluun ei otettu kantaa, joten ehdotus perustuu kirjallisuuteen. Empiirisessä osuudessa kuitenkin ilmeni, että asiakkaalle tulisi näkyä, mitä ratkaisun hankintakustannus on ja toisaalta, mitä jatkuvia kustannuksia ratkaisusta koostuu. Low-code -ratkaisuiden kohdalla jatkuvia kustannuksia ovat esimerkiksi lisenssikustannukset. Hankintakustannuksiin vaikuttaa datalähteeseen, low-code -kehitykseen sekä testaamiseen käytettävä työaika. Kirjallisuudessa tunnistettiin myös, että on erittäin tärkeää kuvailla jo projektin neuvotteluvaiheessa hinnoittelua sekä asiakkaan esitellä budjettia, jotta hinnoittelun osilta ei tule haasteita projektin edetessä (Sipilä 1995). Low-code -teknologian kohdalla nähdään, että ratkaisut ovat joko täysin uusia, jolloin parhaana hinnoittelumallina toimii aikaveloitus. Tämä on sekä palveluntarjoajan, että asiakkaan näkökulmasta hyvä vaihtoehto, koska historiadataa ei ole saatavilla. Toisaalta palveluntarjoajalla voi olla valmiita ratkaisuja, joita palveluntarjoaja on jo kehittänyt. Tässä tapauksessa parhaina vaihtoehtoina nähdään joko urakkahinnoittelun tai tavoitehinnoittelu. Aiemman

kokemuksen perusteella voidaan arvioida paremmin työmääriä, jolloin esimerkiksi tavoitehinnoittelu helpottaa ratkaisun myymistä uudelle asiakkaalle kannattavasti. Tavoitehinnoittelu helpottaa asiakkaan puolelta palvelun ostamista, koska aikataulun ja kustannusten osalta myös palveluntarjoajalla on selkeämpi tavoite ja intressi.

Kuten aiemmin kirjallisuudesta tunnistettiin, ulkoinen tuotteistaminen kertoo kyvystä myydä (Hempe et al. 2015). Myös kirjallisuudesta tunnistettiin näkökulma, että tuotteistettu palvelu on kiinnostuksen herättäjä (kappale 5.1, kuva 21). Tämän tutkimuksen myötä low-code -ratkaisuiden ympärillä asiakkaille voidaan esittää, miten ratkaisu toteutetaan, minkälaisia resursseja se vaatii ja mitä eri vaiheita se sisältää. Toisaalta voidaan esittää erilaisia tapoja tehdä asioita eli pystytään asiakaslähtöisesti pohtimaan mikä olisi paras tapa lähteä tekemään ratkaisua ja kuinka palveluntarjoaja voi tässä auttaa. Kehitetyn yhteistyökehittämisen mallin avulla voidaan tukea, auttaa ja ohjata asiakasta tai toisaalta heille voidaan antaa enemmän vastuuta, mikäli se on toimintatapa, jota asiakas haluaa noudattaa. Nähdään, että viestinnällä, työmääräarvioilla sekä hinnoittelulla on myös selkeä vaikutus siihen, kuinka low-code -palvelut ulkopuolelta nähdään. Näillä tekijöillä on siis vaikutusta sisäiseen, että ulkoiseen tuotteistamiseen.

Tämä tutkimus ottaa hyvin pitkälti sisäiseen tuotteistamiseen kantaa, mikä kirjallisuuden ja tutkimuksen tavoitteiden kannalta on erinomainen päätelmä. Nähdään kuitenkin selkeästi, että sisäisellä tuotteistamisella on suora yhteys ulkoiseen tuotteistamiseen ja se on jopa ulkoisen tuotteistamisen mahdollistaja. Hempe et al. (2015) mukaan ulkoinen tuotteistaminen kertoo organisaation ulkopuolelle palvelun painopisteestä esimerkiksi brändäyksen, markkinoinnin ja koulutuksen avulla. Empiirisessä osuudessa painotettiin etenkin markkinointia ja koulutusta case-organisaation puolesta. Myös dokumentoinnin näkökulma ja tärkeys nostettiin case-organisaation työntekijöiden puolesta esiin, johon tässä tutkimuksessa ei tarkemmin otettu kantaa. Todetaan kuitenkin se, että dokumentoinnin merkitys ja tärkeys on tunnistettu ja sen tarkempi tutkiminen tulevaisuudessa low-coden kontekstissa nähdään perusteltuna. Kappaleessa 7.1 otettiin empiirisen tutkimuksen perusteella kantaa myös koulutuksen näkökulmaan. Markkinointiin ja brändäykseen tutkimuksessa ei perehdytä tarkemmin. Kuva 29 kuvaa tiivistetysti päätelmät, joita tutkimuksessa tehtiin palveluiden tuotteistamisen näkökulmista.



Kuva 29: Tutkimuksen päätelmät low-code -palveluiden tuotteistamisen näkökulmasta.

Kuva 29 vahvistaa, että sisäisen tuotteistamisen kysymyksiin pystyttiin vastamaan hyvin yhteistyökehittämisen mallin avulla. Viestintä, hinnoittelu ja työmääräarviot nähdään olevan osittain sisäistä tuotteistamista ja toisaalta osittain myös ulkoista tuotteistamista. Näillä kolmella tekijällä on vaikutusta sisäisiin prosesseihin ja tekemiseen, mutta ne näyttäytyvät myös organisaation ulkopuolelle. Päätelmä on, että tutkimuksessa pystyttiin hyvin rakentamaan pohja low-code -palveluiden tuotteistamiselle, jota voidaan jatkotutkimuksella laajentaa ja kehittää laajemmaksi.

Kirjallisuudesta tunnistettiin myös palvelun tuotteistamisen prosessimalleja, jotka esiteltiin kappaleessa 4.3. Kuten jo kirjallisuuden yhteenvedossa tunnistettiin, Johnson et al. (2000) kehittämä iteratiivinen palvelun kehitysmalli nähdään potentiaalisimpana tapana kehittää palveluita. Tämä malli mahdollistaa palvelun iteroinnin ja uudelleen arvioinnin yhä uudelleen, mikäli palvelulla nähdään olevan muutostarpeita. Kun tarkastellaan iteratiivista mallia (kuva 17) nähdään, että tässä tutkimuksessa noudatettiin hyvin pitkälti suunnitteluvaihetta. Koska tässä tutkimuksessa suoritetaan yhteistyökehittämisen mallin testaaminen heikolla markkinatestillä, mennään osittain myös toteutusvaiheen puolelle. Tarkoituksena on kuitenkin, että tämän tutkimuksen jälkeen palvelu jalkautetaan case-organisaatioon ja jatketaan toteutusvaihetta pidemmälle. Tämä päätös tehtiin tutkimuksen aikarajoitteiden vuoksi. Tavoitteena on,

että malli ja tuotteistettu palvelu testataan ja lanseerataan case-organisaatioon. Tämä tutkimus antaa kehyksen sille, miten case-organisaatiossa lähdetään low-code -palveluita kehittämään. Koska teknologia on uusi ja sen suosio lisääntyy yhä koko ajan, on tärkeää muistaa toteutusvaiheen jälkeen iteratiivisen mallin tuomat mahdollisuudet tarkastella palveluita yhä uudelleen ja kehittää niitä ajankohtaisiksi, kun esimerkiksi low-code -alustat päivittyvät.

Lopuksi voidaan todeta, että tuotteistamisen näkökulma low-code -teknologian kontekstissa nähdään erittäin tärkeänä ja perusteltuna tutkia. Trott (2008) esittää huomion, että palvelua kehitettäessä ei tulisi painottaa ainoastaan ydinominaisuuksia vaan myös lisäpalveluiden olemassaoloa. Case-organisaation tapauksessa low-code -ratkaisut ovat ainakin tällä hetkellä nimenomaan lisäarvoratkaisuita, eikä niiden kehitystä tule unohtaa. Myös tutkimuksen johdannossa esitettyjen low-code -teknologian tulevaisuuden lukujen perusteella voidaan olettaa, että low-code -teknologia voi jossain kohtaa olla enemmän kuin vain lisäarvoratkaisu, se voidaan nähdä yhtenä liiketoiminnan ydinpalveluna.

7.3 Yhteistyökehittämisen mallin validointi

Yhteistyökehittämisen mallille suoritettiin heikko markkinatesti, joka on tarkemmin esitelty kappaleessa 2.2. Tässä kappaleessa keskitytään heikon markkinatestin tuloksiin. Yhteistyökehittämisen malli esiteltiin kahdelle henkilölle, case-organisaation asiakkaalle sekä työntekijälle. Case-organisaation asiakkaan edustajaan viitataan lyhenteellä T1 ja case-organisaation työntekijään lyhenteellä T2.

Ensimmäisenä henkilöille esiteltiin yhteistyökehittämisen roolit ja vastuut eli kappaleen 7.1 kuva 25. Molempien näkemys oli, että roolit olivat relevantteja ja malli on selkeä sekä ymmärrettävä. Erityisesti konsultin roolin merkitys korostui tarpeellisena. Kirjallisuudesta löydettyssä Boyer et al. (2021) kehittämässä mallissa konsultin roolia ei ollut erikseen määriteltä. Voidaan siis todeta, että tämä oli merkittävä uusi huomio ja mallin kehitys. Uuden roolin tuominen malliin lisää tutkimuksen uutuusarvoa, joka vahvistettiin heikon markkinatestin avulla.

”Tämä on yksinkertainen ja nopealla katsomisella ilman sinun selitystäkään mielestäni heti ymmärsin, mistä on kysymys. Hyvännäköinen kaavio. Ja kyllä mielestäni tämä kuvaa asiaa hyvin.” (T1)

”Mun mielestä tätä ei kannata monimutkaisemmaksi lähteä tekemään, tätä voi aina soveltaa ja lisätä rooleja, jos siltä tuntuu, mutta tämä on hyvä lähtökohta.” (T1)

”Kyllä mun mielestä toi olisi tosi hyvä. Näen, että kaikilla, tietysti loppukäyttäjä on käyttäjä, mutta jos ajatellaan noita kolmea muuta roolia niillä kaikilla, on oikeasti tarpeellinen rooli olla siinä. Esimerkiksi jos ei ole erillistä konsulttia, suunnitteluvaihe ja testausvaihe voi jäädä vähän suppeaksi. Ylipäätään sellainen ideointi ja oletuksena konsultti osaa nähdä sen enemmän asiakkaan näkökulmasta valmiiksi. Ja semmoisia käytettävyyssjuttuja teknisten asioiden lisäksi, ne on tärkeitä varsinkin tällaisessa piensovelluskehityksessä.” (T2)

”Varsinkin alkuvaiheessa ennenkö lähdetään tekemään, jos se on jokin tosi monimutkainen ratkaisu, niin mun mielestä toi konsultti on todella tärkeä.” (T2)

Kansalaiskehittäjän rooliin tuotiin mielenkiintoinen näkökulma, jossa pohdittiin, että voisiko molemmilta puolilta sekä palveluntarjoajalta, että asiakkaalta tulla työntekijä, joka toimisi kansalaiskehittäjän rooleissa. Mallissa pohdittiin myös tätä näkökulmaa (taulukko 7, vaihtoehto 5), joten testaaminen vahvisti tämän vaihtoehdon olemassaoloa.

”Kansalaiskehittäjä kumpi se sit on (asiakkaan vai palveluntarjoajan työntekijä) vai voisiko olla, että tehdään yhteistyössä molemmilta puolilta, kansalaiskehittäjän roolissa sekä palveluntarjoajalta, että asiakkaalta, se riippuu siitä asiakkaan kyvyistä ja valmiuksista ja innostuksesta.” (T2)

Sitaatista huomataan myös toinen hyvä huomio eli se kuka kansalaiskehittäjä on, riippuu asiakkaan kyvyistä ja myös innostuksesta. Tämä tukee jo aiemmin tehtyä huomiota siitä, että kansalaiskehittäjän roolin pitää pystyä mukautumaan tarpeen vaatiessa asiakkaan tai palveluntarjoajan tehtäväksi. Innostukseen ei tarkemmin tutkimuksessa otettu kantaa, mutta se on myös mielenkiintoinen näkökulma, jota tässä kontekstissa voisi olla tarpeellista tulevaisuudessa tutkia.

Toiseksi henkilöille esiteltiin luvun 7.1. taulukko 7. Sen merkitystä selitettiin ja avattiin, jonka jälkeen siitä pyydettiin liitteen C mukaisesti huomioita. Myös tämä taulukko nähtiin selkeänä ja ymmärrettävänä, kuitenkin niin, että se kaipaa tuekseen idean avaavan selityksen. Testaamisesta nousi erityisesti huomiona mallin skaalautuvuus. Sitä voidaan tulevaisuudessa jatkojalostaa ja työntekijöiden määrät eivät ole lukittuja, vaan tarpeen vaatiessa näitä voidaan muokata. Perusideana taulukko koettiin hyvänä ja selkeänä sekä ideaa avaavana.

”Hyvältä näyttää, nämä ovat mahdollisia kokoonpanoja.” (T1)

”Jos oikein iso projekti, niin sitten tarvitaan kaksi sovelluskehityksen ammattilaista tai kolmekin, että sitten se vain skaalautuu.” (T1)

”Mun mielestä on ymmärrettävä ja selkeä. Kun selittää tämän auki, niin tämän ymmärtää.” (T2)

”Tämä avaa sitä, että pystyy tehdä case-kohtaisesti, mikä koetaan järkeväksi ja kuinka vaikea ratkaisu on.” (T2)

Kolmantena esiteltiin kappaleessa 7.1 esitelty kuva 26, jossa otetaan kantaa low-code -kehityksen eri vaiheisiin ja vastuisiin. Testeissä henkilöt tukivat kehitettyä mallia ja näkivät sen selkeänä ja ymmärrettävänä. Erityisesti korostettiin sitä, että malli olisi hyvä esittää myös jollakin tapaa asiakkaalle, jotta vastuut eri vaiheissa on myös heille alusta asti selkeitä. Testeissä esiintyi myös se näkökulma, että vastuut voivat tapauskohtaisesti hieman muuttua.

”Kyllä yritin tätä ajatuksissa peilata omaan projektiin, siihen nähden niin varmaan on juurikin näin. Varmasti tapauskohtaisesti voi pikkasen elää, esimerkiksi onko liiketoimintalogiikan kehityksessä sovelluskehityksen ammattilainenkin mukana, mutta jälleen kerran jostain pitää lähteä ja perusmallina tämä on oikein pätevä kaavio.” (T1)

”On ymmärrettävä ja selkeä, en keksi mitään kehitettävää. Voisin hyödyntää tätä omassa työssäni. Kun mietin projekteja, mitä olemme tähän mennessä tehnyt, olisi ollut todella hyvä, kun olisi ollut jonkin tällainen malli. Olisi hyvä puhua asiat, mitä kuva kertoo projektin alussa, jotta kaikki tietää miten siinä edetään. Monessa kohtaa olisi varmaan selkeyttänyt.” (T2)

Voidaan siis todeta, että tässä tutkimuksessa luotu yhteistyökehittämisen malli läpäisi heikon markkinatesti. Kuten aiemmin esiteltiin, testeissä ilmeni mallin mm. skaalautuvuuteen liittyvä huomioita, joten on perusteltua pohtia ratkaisun soveltamisalaa. Kappaleessa 7.1 esitetty malli luotiin teorian sekä empiirisen osuuden perusteella. Empiirisessä osuudessa haastateltiin vain case-organisaation asiakkaita sekä työntekijöitä, joten on mahdollista, että yritysten toimialalla sekä koolla on vaikutusta mallin toimivuuteen. Mallissa ei kuitenkaan nähdä esteitä miksi sitä ei voisi skaalata esimerkiksi suurten yritysten tarpeisiin. Myös nyt empiirisessä osuudessa haastateltiin usealta eri toimialalta henkilöitä, joka vahvistaa mallin toimivuutta eri ympäristöissä. Mallia tullaan jatkossa case-organisaation toimesta testaamaan käytännön tasolla. Kappaleessa 8.4 esitellään jatkotutkimusehdotuksia, jotka ottavat kantaa myös mallin vahvistamiseen ja kehittämiseen tulevaisuudessa.

8. YHTEENVETO

Tässä luvussa esitetään tutkimuksen yhteenveto sekä keskeiset havainnot. Luvussa käsitellään myös huomioita teoreettisen ja käytännön kontribuution näkökulmista. Lopuksi arvioidaan, kuinka tutkimuksessa on onnistuttu eri näkökulmista sekä ehdotetaan jatkotutkimukseen erilaisia mahdollisuuksia.

8.1 Yhteenveto ja keskeiset havainnot

Tutkimus toteutettiin kvalitatiivisena tapaustutkimuksena konstruktivisella tutkimusotteella, jossa empiirinen osuus muodostui puolistrukturoidusta teemahaastattelusta sekä ratkaisun testaamisesta heikolla markkinatestillä. Tarkoituksena oli tulkita ja ymmärtää toimijoiden näkökulmia (Hirsjärvi & Hurme 2022). Täten voidaan ymmärtää, miksi tällä hetkellä toimitaan näin ja minkä toivottaisi kehittyvän tapauksen kontekstissa. Kvantitatiivisella tutkimuksella pyritään yleisesti löytämään tietoa jonkin ilmiön laajuudesta ja voimakkuudesta (Hirsjärvi & Hurme 2022). Kuten tutkimuksen alussa kerrottiin, tutkimuksen tarkoituksena oli kehittää ja parantaa toimintaa eikä tutkimuskysymyksillä pyritty löytämään vastauksia aihepiirin laajuuteen tai voimakkuuteen liittyen. Kirjallisuuskatsaus loi teoreettisen pohjan ja auttoi tutkijaa ymmärtämään tämänhetkistä tietoa aihepiirin ympäriltä. Empiirisellä osuudella syvennettiin tietoa ja käytiin vuoropuhelua yhdessä kirjallisuuden kanssa. Luotua mallia testattiin, jotta pystyttiin varmistumaan sen toimivuudesta.

Tutkimuksen taustaa tutkiessa huomattiin, että low-code -teknologia on suhteellisen uusi ja moderni tapa kehittää nopeasti digitaalisia ratkaisuja. Huomiona tehtiin myös ennustukset siitä, kuinka suuressa osassa liiketoiminnan digitalisaation kehitystä se tulee olemaan. Tämän takia aihe koetaan erityisen relevanttina tänä päivänä. Yhteistyötiimejä ja yhteistyökehittämistä oli low-coden kontekstissa myös perusteltua tutkia, sillä kansalaisehittäjien tuominen ohjelmistokehitykseen vaatii uusia innovatiivisia tapoja tuottaa ratkaisuja. Odotuksena oli, että tällä mallilla voitaisiin luoda ketteriä ja tehokkaita ratkaisuita liiketoiminnan tarpeisiin. Toisaalta nähdään, että palveluiden kehittäminen on ensiarvoisessa asemassa, jotta low-code -ohjelmistokehitys voi olla selkeää, tehokasta, tuottavaa ja kilpailukykyistä.

Tutkimuksessa onnistuttiin luomaan yhteistyökehittämisen malli low-code -teknologialle sekä tarkastelemaan mallin tuomia mahdollisuuksia ja kehystä low-code -palveluiden tuotteistamiseen. Päättökysymyksen tarkoituksena oli löytää vastaus siihen, miten

palvelujen tuotteistaminen, yhteistyökehittäminen sekä low-codella tehdyt teknologiset ratkaisut liittyvät toisiinsa ja miten ne yhdessä voisivat muodostaa parhaan mahdollisen tavan toimia toimeksiantajan kontekstissa. Alatutkimuskysymysten tarkoituksena oli tukea laajempaa päätutkimuskysymystä. Ensimmäisen alatutkimuskysymyksen avulla pyrittiin löytämään vastaus siihen, mitä palvelujen tuotteistaminen tarkoittaa ja miten sitä voisi hyödyntää low-code -teknologian ympärillä. Toisella alatutkimuskysymyksellä pyrittiin löytämään vastaus siihen, mitä yhteistyökehittäminen on ja miten ne liittyvät palveluiden tuotteistamiseen. Kun pohditaan tutkimuksen alussa asetettuja tutkimuskysymyksiä, voidaan todeta, että tutkimuksella pystyttiin vastaamaan näihin.

8.2 Teoreettinen ja käytännön kontribuutio

Lukan (2006) mukaan konstruktivisen tutkimusotteen prosessin viimeisenä vaiheena on tunnistaa teoreettinen kontribuutio ja analysoida sitä. Tarkastellaan siis seuraavaksi tutkimuksen teoreettista sekä käytännön kontribuutiota.

Teoreettinen kontribuutio

Low-code -teknologian ympäriltä löydettiin tutkimusta melko hyvin, jota oli tehty erityisesti viimevuosien aikana. Yhteistyökehittämistä sekä ketterien menetelmien hyödyntämistä low-code -teknologian kontekstissa oli myös tutkittu aiemmassa kirjallisuudessa jonkin verran. Palveluiden tuotteistamisesta yhdessä low-code -teknologian kanssa ei löytynyt paljoakaan aiempaa tutkimusta. Tästä syystä tutkimuksessa hyödynnettiin kirjallisuutta asiantuntijapalveluiden tuotteistamisesta. Di Ruscio et al. (2022) esittämä low-code -teknologian elinkaari selittää, miksi low-code -teknologian ympärillä on tehty tutkimusta nimenomaan viime vuosina ja miksi siitä ei kovinkaan laajaa tutkimusta esimerkiksi yhdessä palveluiden tuotteistamisen kanssa vielä löydy (kappale 3.1, kuva 4). Tutkijan näkökulmasta mielenkiintoista oli kuitenkin se, että yhteistyökehittämisen teemaa oli jo jonkin verran tutkittu. Tämän tutkimuksen avulla muokattiin jo olemassa olevaa teoriaa, joka liittyy low-code -teknologiaan ja yhteistyökehittämiseen. Aiempaan teoriaan lisättiin uusia muuttujia empiirisen tutkimuksen avulla. Palveluiden tuotteistamisen ja low-code -teknologian kohdalla empiirinen osuus loi uutta teoriaa, koska aiempaa tutkimusta ei kovinkaan paljoa löytynyt. Toisaalta tutkimus toi aihepiiriin tutkimuskenttään uutta näkökulmaa tutkijan havaintojen perusteella siitä, että aihealueet ovat sidoksissa toisiinsa (kappale 5.1, kuva 21). Tämänkaltaista kuvausta aiheiden yhteydestä ei kirjallisuudesta löydetty. Näistä kolmesta aiheesta yhdessä low-code -teknologia, palveluiden tuotteistaminen sekä yhteistyökehittäminen ei löydetty aiempaa tutkimusta eikä näitä ollut aiemmin liitetty tällä

tavalla yhteen. Kuvaus näiden yhteydestä sekä niiden merkityksestä yhdessä oli siis täysin uutta ja täten loi uutta teoriaa aihealueen tutkimuskentälle.

Yhteistyökehittämisen mallissa nähdään kaksi muuttujaa, jotka erityisesti muokkasivat aiempaa teoriaa: kuvassa 25 esitetty konsultin ja kansalaiskehittäjän roolit. Kuten jo aiemmin on todettu, konsultin roolia ei tunnistettu esimerkiksi Boyer et al. (2021) esittelemässä fuusiokehittämisen mallissa lainkaan. Tämän muuttujan vaikutukset yhteistyökehittämiseen arvioidaan olevan positiiviset, sillä konsultin osaamisen avulla voidaan parhaassa tapauksessa suoraviivaistaa sekä parantaa low-code -kehitystä. Kansalaiskehittäjän roolin näkökulmasta aiempaan teoriaan täydentävä ominaisuus on selkeämpi roolin jako asiakkaan ja palveluntarjoajan välille. Boyer et al. (2021) mukaan lähtökohtana voidaan pitää sitä, että kansalaiskehittäjät eli yrityksen liiketoiminnan keskiössä työskentelevät työntekijät tekevät sovellusratkaisut. Yhteistyökehittämisen mallilla selkeytettiin tätä toteamaa, jolloin osaamista voidaan hyödyntää joko asiakkaan tai palveluntarjoajan toimesta. Heikossa markkinatestissä ilmeni myös näkökulma, jossa ehdotettiin, että molemmilta puolilta oma henkilö toteuttaa tätä roolia yhdessä. Empiirisen osuuden myötä vaikutettiin siis myös mallin selkeyteen. Taulukolla 7 selkeytettiin myös mallin eri mahdollisia kokoonpanoja, joka taas vaikuttaa yhteistyötiimin rooleihin ja vastuisiin. Tämänkaltaista kokoonpanojen erittelyä ei esitelty aiemmassa kirjallisuudessa lainkaan vaan se luotiin empiirisen tutkimuksen perusteella.

Kuvassa 26 esitettyjen yhteistyökehittämisen vaiheiden ja vastuiden kautta vahvistettiin sekä lisättiin tietoa olemassa olevaan teoriaan. Vaiheet oli tunnistettu kirjallisuudesta Al Alamin et al. (2021) toimesta, johon tämän tutkimuksen empiirisellä osuudella saatiin luotua vastuut. Al Alamin et al. (2021) esittelemässä mallissa vastuita ei määritelty lainkaan. Heikko markkinatesti osoitti, että tämän perusteella pystytään vaikuttamaan low-code -kehitykseen positiivisella tavalla. Täten yhteistyökehittämisen malli pystyy vastaamaan kysymyksiin roolituksen ja niiden vastuiden osalta: miten roolitettut tiimit kootaan sekä eri vaiheiden vastuujako. Kirjallisuuden perusteella pystyttiin vastamaan ainoastaan kysymyksiin minkälaisia rooleja low-code -kehityksessä tulisi olla ja mitä vaiheita se sisältää. Vastuihin eikä mahdollisiin kokoonpanoihin ei otettu kantaa.

Asiakaskeskeisyys oli yhteistyökehittämisen mallin kehittämisen lähtökohta. Asiakaskeskeisyys ja sen merkitys tunnistettiin aiemman kirjallisuuden perusteella arvon yhteisluonnin kontekstissa. Yhteistyökehittämisen mallissa asiakkaat otetaan huomioon monesta eri näkökulmasta ja malli mahdollistaakin, että asiakas tekee valinnan, miten low-code -ratkaisua toteutetaan. Toisaalta se mahdollistaa, että asiakas tekee päätöksen roolituksen vastuujaosta heidän kykyjen mukaan. Tämänkaltaista asiakaskeskeisyyttä ei havaittu kirjallisuudessa esitetyissä malleissa, koska niissä

vastuunjakoon tai eri kokoonpanoihin ei otettu kantaa. Yhteistyökehittämisen mallin asiakaskeskeisyys mahdollistaa myös arvon yhteisluonnin. Vastuiden jakaminen sekä asiakkaalle, että palveluntarjoajalle mahdollistaa asiakkaan osallistumisen prosessiin yhteistuottajana, joka luo mahdollisuuden todelliseen arvon luontiin Grönroos-Voima -mallin mukaisesti.

Kuten aiemmin esitettiin, palvelun tuotteistamisen näkökulma low-code -teknologian kontekstissa on melko suppeasti tutkittu osa-alue. Empiirisellä osuudella onnistuttiin kuitenkin täydentämään tätä tutkimuskenttää etenkin sisäisen tuotteistamisen osalta. Voidaan ajatella, että yhteistyökehittämisen malli itsessään toimii tuotteistamiskeinona, jossa nimenomaan vaikutetaan sisäiseen tuotteistamiseen. Ulkoisen tuotteistamisen näkökulmasta organisaation ulkopuolelle voidaan esittää, miten low-code -ratkaisu toteutetaan, minkälaisia resursseja se vaatii ja mitä eri vaiheita se sisältää. Myös viestinnän, hinnoittelun ja työmääräarvioiden tutkimisen avulla pystyttiin ottamaan kantaa ulkoiseen tuotteistamiseen. Low-code -palveluiden kontekstissa tapahtuva viestintä näyttäytyy myös organisaation ulkopuolelle muodollisen viestinnän merkeissä. Työmääräarvioilla sekä hinnoittelulla voi olla merkitystä siihen ostaako asiakas low-code -palveluita. Tutkimuksella pystyttiin siis selkeyttämään viestinnän, hinnoittelun ja työmääräarvioiden toteutustapaa low-code -palveluiden kontekstissa. Arvioidaan kuitenkin, että vaikutukset ulkoisen tuotteistamisen näkökulmaan jäivät suppeammaksi kuin sisäisen tuotteistamisen näkökulma. Tätä ei kuitenkaan nähdä negatiivisena lähtökohtana, koska kirjallisuudesta tunnistettiin, että sisäisen toiminnan systematisointi on edellytys, jotta ulkoista tuotteistamista voidaan tehdä (Sipilä 1995). Nähdään siis, että ulkoisen tuotteistamisen näkökulmaa voidaan vahvistaa mahdollisella jatkotutkimuksella.

Kun tarkastellaan tutkimuksen hyödyllisyyttä arvioidaan, että tutkimuksessa on onnistuttu. Kattavamman testaamisen avulla voidaan varmistua, että luotua mallia voitaisiin hyödyntää laajemmin. Esteitä laajemmalle käytölle ei tässä vaiheessa ole vielä ilmennyt, mutta vahvempi testaaminen vahvistaisi tätä näkökulmaa. Tutkimuksen hyödyllisyyttä kuvaa myös se, että sen avulla pysytään yleisesti vahvistamaan low-code -teknologian, yhteistyökehittämisen sekä palveluiden tuotteistamisen tutkimuskenttää.

Käytännön kontribuutio

Käytännön tasolla nähdään, että tällä on merkittävä vaikutus case-organisaatiolle. Tätä vahvistaa mm. heikon markkinatestin tulokset. Tutkimuksen tulos antaa konkreettisia ja käytännönläheisiä ehdotuksia siihen, miten low-code -kehitystä voidaan roolituksen, vaiheiden ja vastuujonon näkökulmista tehdä sekä miten low-code -palveluita voidaan

tuotteistaa ja kehittää case-organisaatiossa. Tutkimuksen tärkein käytännön kontribuutio on se, että case-organisaatio saa käytettäväkseen yhteistyökehittämisen mallin low-code -teknologialle ja pääsee hyödyntämään sitä yrityksen kehittämisessä asiakaslähtöisesti. Yhteistyökehittämisen mallilla voidaan selkeyttää case-organisaation työntekijöiden sekä asiakkaiden rooleja low-code -projekteissa. Tuotteistamisen näkökulmasta nähdään, että case-organisaatio pystyy low-code -palveluissa tehostamaan toimintaansa ja täten vähentää päällekkäisiä työtehtäviä. Tuotteistettu palvelu mahdollistaa työn monistamisen yhä helpommin, joten oletetaan, että case-organisaatio voi tehdä yhä enemmän low-code-ratkaisuita. Tämä mahdollistaa case-organisaation low-code -palveluiden liiketoiminnan sekä tuottojen kasvun.

Kun tarkastellaan tutkimuksen käytännön kontribuutiota laajemmalla näkökulmalla nähdään, että case-organisaatio voisi hyödyntää yhteistyökehittämisen mallia laajemmalla alueella. Yhteistyökehittämisen mallin hyödyntämistä voisi arvioida myös case-organisaation keskiössä olevan ERP-liiketoiminnan kontekstissa. Erityisesti nähdään, että yhteistyökehittämisen mallin hyödyntämistä voitaisiin arvioida case-organisaation tiimien kohdalla. Malli tukee moniosaamista ja sen lisäämistä yrityksessä. Yhteistyökehittämisen mallissa on piirteitä, jotka tukevat tiimien muodostamista eri osaajista tarpeen mukaisesti. Nähdään, että mallin osa, joka ottaa kantaa roolituksiin sekä eri tiimien kokoonpanoihin voisi olla erinomainen työkalu myös tiimien luomisen kohdalla ketterässä organisaatiossa.

8.3 Tutkimuksen arviointi

Tutkimuksen alusta asti oli selvää, että tutkimuksessa noudatetaan hyviä tieteellisiä käytäntöjä. TENK (2023) määrittelee, että hyvän tieteellisen käytännön peruseriaatteita ovat luotettavuus, rehellisyys, arvostus ja vastuunkanto. Thomas & Magilvy (2011) mukaan laadullista tutkimusta voidaan arvioida uskottavuuden, siirrettävyyden, luotettavuuden sekä vahvistettavuuden kautta. Viime aikoina on tunnustettu huoli tutkimusten eettisyyden näkökulmasta (Saunders et al. 2019). Tämän takia tutkimusta arvioidaan myös eettisyyden kannalta.

Eettisyys

Eettisyydellä tarkoitetaan eettisten periaatteiden noudattamista koko tutkimuksen ajan. Tutkimuksesta ei saa aiheutua haittaa tutkittavalle tai siihen liittyville tahoille. Tutkimuksessa tulee huomioida, että sen pyrkimys on saada hyviä asioita aikaan. (Puusa et al. 2020)

Tutkimuksessa noudatettiin eettisyyttä jokaisessa vaiheessa parhaalla mahdollisella tavalla ja tutkija piti tämän näkökulman mielessä koko tutkimuksen ajan. Asiakashaastatteluihin osallistuneille haastateltaville lähetettiin tietosuojailmoitus ennen haastatteluita. Jokaiselta haastateltavalta kysyttiin myös halukkuutta osallistua tutkimukseen eli osallistuminen perustui vapaaehtoisuuteen. Ennen jokaista haastattelua kerrottiin myös, että haastattelu nauhoitetaan, mikäli se sopii haastateltavalle. Haastattelun alussa tämä vielä varmistettiin ja perusteltiin, miksi haastattelu nauhoitetaan. Tietosuojailmoituksessa eriteltiin vielä tarkemmin mihin käyttötarkoitukseen nauhoituksia käytetään ja missä haastattelunauhoituksia säilytetään. Ilmoituksessa esitettiin myös haastateltavan oikeus pyytää poistaa kerättyä aineistoa. Haastateltavat anonymisoitiin, jotta heitä ei voi empiirisen tutkimuksen tuloksista tunnistaa. Myös jokaisen haastateltavan vastuksen pyrittiin myös huomioimaan tasapuolisesti.

Luotettavuus & Uskottavuus

Luotettavuudella tarkoitetaan ammattitaidon esiintymistä koko tutkimuksen ajan ja sen näkyminen lukijalle perusteltujen lähestymistapojen ja menetelmien kautta. Jotta luotettavuudesta voidaan vakuuttua, tulee tutkimuksen eteneminen kuvata sellaisenaan tutkimukseen. (Puusa et al. 2020) Tutkijalla on kokemusta low-code -teknologiasta ja sen hyödyntämisestä työtehtävien kautta. Tutkimuksen aikana pyrittiin kuitenkin unohtamaan tutkijan omat mielipiteet ja näkökulmat, mutta kokemuksen kautta tehdyillä havainnoilla on saattanut olla vaikutusta näkökulmiin, joita tutkimuksessa on tutkittu. Tutkimusprosessia on pyritty esittelemään mahdollisimman avoimesti ja läpinäkyvästi, mikä lisää tutkimuksen luotettavuutta ja toistettavuutta. Esimerkkinä eri vaiheet, joiden mukaan tutkimusta tehtiin sekä suurin osa hakulausekkeista on myös esitetty.

Mikäli yleisö hyväksyy tutkimuksen tulokset todenmukaisiksi sekä luotetaan siihen, että aineisto on kerätty asianmukaisesti ja analysoitu huolellisesti, voidaan puhua uskottavasta tutkimuksesta (Puusa et al. 2020). Tutkimuksessa aineisto kerättiin puolistrukturoidun teemahaastatteluiden avulla ja analysoitiin teemoittelua hyödyntäen. Ennen haastatteluiden organisointia pohdittiin relevantteja haastateltavia sekä riittävää aineiston määrä, jotta empiirisen tutkimuksen perusteella tehtävät päätelmät olisivat oikeita. Haasteena oli löytää haastateltavia riittävä määrä, jotta empiirisen osuuden perustella voitiin tehdä päätelmiä. Low-code -teknologian uutuuden vuoksi koettiin, että haastateltavia oli haastava löytää. Haastatteluiden oletuksena oli myös se, että haastateltavalla oli jonkinlainen ymmärrys low-code -teknologiasta ja ratkaisusta, joita teknologialla voidaan tehdä. Lopulta voidaan arvioida, että asiakashaastatteluiden osalta onnistuttiin löytämään tarpeeksi haastateltavia. Case-organisaation työntekijöitä, joilla

teknologiasta on kokemusta tarpeeksi, oli kaksi kappaletta. Molempien vastaukset olivat hyvin samankaltaisia ja täten se lisäsi uskottavuutta. Tämä nähdään kuitenkin melko pienenä otantana. Mallin validointi suoritettiin kahdella henkilöllä, joten otanta testaamisessa oli pieni. Rajoitteina oli tutkimuksen aikataulu ja laajuus. Toisaalta molemmat heikkoon markkinatestiin osallistuneista oli osallistunut myös empiirisen osuuden haastatteluihin, jolla on voinut olla vaikutusta testin lopputulokseen.

Siirrettävyys & Vahvistettavuus

Tutkimuksen siirrettävyyttä voidaan arvioida sen yleistettävyyden kautta (Saunders et al. 2019). Siirrettävyys tarkoittaa, että tutkimustuloksia voidaan siirtää toiseen kontekstiin ja kuinka sovellettavia ne ovat toisessa ympäristössä tai ryhmässä. (Thomas & Magilvy 2011) Tutkimuksen siirrettävyyttä puoltaa se, että päätelmät tehtiin yhdessä kirjallisuuden ja empirian pohjalta. Empiirisen osuuden rajoitteina siirrettävyyden näkökulmasta nähdään otannan pieni koko sekä tarkastelun kohteena oli yhden organisaation asiakkaat ja työntekijät. Toisaalta asiakkaat olivat eri toimialoilta ja rooleista, mikä puolestaan vahvistaa siirrettävyyttä. Toisaalta myös esimerkiksi taulukko 7 ja sen mukaisesti skaalautuvuuden mahdollistaminen lisää siirrettävyyttä eri kokoisiin organisaatioihin. Kuten aiemmin tuotiin esiin tutkimuksen läpinäkyvyyden olevan hyvällä tasolla, mikä toisaalta vahvistaa myös siirrettävyyden mahdollisuutta. Puolistrukturoitu teemahaastattelu mahdollisti mm. kysymysten muotoilun eri tavalla eri haastateltaville, mikä toisaalta heikentää myös tutkimuksen siirrettävyyttä ja saman lopputuloksen saantia toisen tutkijan toimesta.

Thomas & Magilvy (2011) mukaan vahvistettavuus tapahtuu, kun uskottavuus siirrettävyys ja luotettavuus on vahvistettu. Aiempien arvioiden perusteella voidaan todeta, että tutkimuksen luotettavuus ja siirrettävyys ovat hyvällä tasolla, minkä mukaan tutkimuksen vahvistettavuus on myös osoitettu. Siirrettävyydessä nähtiin kuitenkin, joitakin puutteita sekä tutkimuksen luottamuksellisuus lisää myös siirrettävyyden haasteita. Toisaalta vahvistettavuutta lisättiin heikolla markkinatestillä, jossa varmistettiin kirjallisuuden ja empirian päätelmät oikeiksi.

8.4 Jatkotutkimusmahdollisuudet

Tutkimuksen aikana tunnistettiin, että low-code -teknologian ympärillä tutkimuksen määrä on kasvanut merkittävästi vuosina 2021–2023. Tämän tutkimuksen aihe on laaja, joka mahdollistaa jatkotutkimusta eri näkökulmista. Low-codesta ja palveluiden tuotteistamisesta ei löydetty merkittävästi aiempaa tutkimusta, vain yksittäisiä kappaleita. Tämän perusteella voidaankin ehdottaa, että aiheen tutkiminen eri

näkökulmista on relevanttia myös tulevaisuudessa. Palvelun tuotteistamisesta löydettiin myös hyvin paljon tutkimuksia 1980–1990-luvuilta, jotka jossain tapauksessa nähtiin hieman vanhentuneeksi. Tämä on myös yksi näkökulma, joka perustelee tuotteistamisen tutkimista, hyvinkin uuden ja nykypäiväisen teknologian kontekstissa. Myös selkeästi tunnistettu aihe on ulkoinen tuotteistaminen, joka jäi tässä tutkimuksessa kapeaksi. Ulkoisen tuotteistamisen tutkiminen low-coden kontekstissa olisi myös hyvä jatkumo tälle tutkimukselle, sillä sisäisen tuotteistaminen antaa tähän hyvät lähtökohdat.

Tässä tutkimuksessa kehitetty malli testattiin heikolla markkinatestillä, joten mallin testaaminen ja mahdollinen jatkokehittäminen nähdään tarpeellisena. Tässä tutkimuksessa ei otettu kantaa yritysten kokoon tai eri toimialoihin, joten myös näiden näkökulmien huomioiminen mallin testaamisessa tulisi tutkia. Mallia voisi kehittää esimerkiksi toimiala kohtaiseksi ottaen huomioon eri kokoiset yritykset. Näihin tekijöihin ei tässä tutkimuksessa otettu kantaa diplomityön aikarajoitteen sekä laajuuden vuoksi. Kokonaisuudessaan nähdään, että low-code -teknologian tutkiminen eri näkökulmista on relevanttia, koska sen suosio on kasvanut ja sen rooli tulevaisuuden liiketoiminnassa nähdään merkittävänä.

LÄHTEET

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.
- Ajmal, M. M., Jan, A., Khan, M., Hussain, M., & Salameh, A. A. (2023). Exploring the barriers and motivators of value co-creation through a theoretical lens of service-dominant logic. *Journal of Business & Industrial Marketing*, (ahead-of-print).
- Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B., & Iqbal, A. (2021). An empirical study of developer discussions on low-code software development challenges. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (pp. 46-57). IEEE.
- Anttonen, Kyösti. (2003). *Tehosta projektityötä: johda hanketta 80/20-periaatteella*. Talentum.
- Ayyar, S., Gelders, R. and Heyes, J. (2023). 5 Principles to Help Fusion Teams Power Your Digital Business. Saatavilla: <https://www.gartner.com/document/4049599?ref=solrAll&refval=366950701> (Viitattu 19.5.2023)
- Bharosa, N., Meijer, K., & Van Der Voort, H. (2020). Innovation in public service design: Developing a co-creation tool for public service innovation journeys. In *The 21st Annual International Conference on Digital Government Research* (pp. 275-284).
- Bhattacharyya, S. S., & Kumar, S. (2021). Study of deployment of “low code no code” applications toward improving digitization of supply chain management. *Journal of Science and Technology Policy Management*, (ahead-of-print).
- Bock, A. C., & Frank, U. (2021). Low-code platform. *Business & Information Systems Engineering*, 63, 733-740.
- Bonamigo, A., da Silva, A. A., da Silva, B. P., & Werner, S. M. (2022). Criteria for selecting actors for the value co-creation in startups. *Journal of Business & Industrial Marketing*, 37(11), 2332-2343.
- Boyer, S., Sharp, J., Matthews, A. & Stollery P. (2021). Ebook: Fusion development approach to building apps using Power Apps. EDITION v1.0, Microsoft Developer Relations and Power Apps product teams. Saatavilla: <https://learn.microsoft.com/en-us/power-apps/guidance/fusion-dev-ebook/> (Viitattu 15.6.2023)
- Brady, M. K., & Cronin Jr, J. J. (2001). Some new thoughts on conceptualizing perceived service quality: a hierarchical approach. *Journal of marketing*, 65(3), 34-49.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- Cabot, J. (2020, October). Positioning of the low-code movement within the field of model-driven engineering. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (pp. 1-3).

- Di Ruscio, D., Kolovos, D., de Lara, J., Pierantonio, A., Tisi, M., & Wimmer, M. (2022). Low-code development and model-driven engineering: Two sides of the same coin?. *Software and Systems Modeling*, 21(2), 437-446.
- Duncan, A. D. (2021). Over 100 Data and Analytics Predictions Through 2025.
- Eiglier, P., Langeard, E., & Mathieu, V. (1997). Le marketing des services. *Encyclopédie de gestion*, 2, 1937-1948.
- Eskola, J., & Suoranta, J. (1998). Johdatus laadulliseen tutkimukseen. Vastapaino.
- Grönroos, C. (1982). An applied service marketing theory. *European journal of marketing*, 16(7), 30-41.
- Grönroos, C. (1998). Nyt kilpaillaan palveluilla. Porvoo: WSOY.
- Grönroos, C. (2001). The perceived service quality concept—a mistake?. *Managing Service Quality: An International Journal*, 11(3), 150-152.
- Grönroos, C. (2017). On Value and Value Creation in Service: A Management Perspective. *Journal of Creating Value*, 3(2), 125–141. <https://doi.org/10.1177/2394964317727196>
- Haapala, W. (2020). Konsultoinnin hyödyt yrityksille: Konsultoinnin riskienhallinta.
- Harkonen, J. (2021). Exploring the benefits of service productisation: support for business processes. *Business Process Management Journal*, 27(8), 85-105.
- Hemple, G., Finch, J., & Pasternak, O. (2015). Productization as a service: innovation through standards in B2B relationships. In *ISPIM Conference Proceedings* (p. 1). The International Society for Professional Innovation Management (ISPIM).
- Hirsjärvi, S. & Hurme, H. (2022). Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. E-kirja, Gaudeamus. Vaatii käyttöoikeuden. Saatavilla: https://catalog-fi.nextory.com/kirja/tutkimushaastattelu-teemahaastattelun-teoria-ja-kaetanto-11800696/?gclid=EAlalQobChMIwdurvrSu_wIVQfgYCh0jpA2QEAAAYASAAEgLI4vD_BwE (Viitattu 6.6.2023)
- Hyötyläinen, M., & Möller, K. (2007). Service packaging: key to successful provisioning of ICT business solutions. *Journal of Services Marketing*, 21(5), 304-312.
- Iho, S., Krejci, D., & Missonier, S. (2021). Supporting Knowledge Integration with Low-Code Development Platforms. In *ECIS*.)
- Jaakkola, E. (2011). Unraveling the practices of “productization” in professional service firms. *Scandinavian Journal of Management*, 27(2), 221–230. <https://doi.org/10.1016/j.scaman.2011.03.001>
- Jain, A., Iijima, K., Leow, A., Wong, J. & Vincent, P. (2021). Critical Capabilities for Enterprise Low-Code Application Platform. Saatavilla: <https://www.gartner.com/document/code/738188?ref=authbody&refval=4015020> (Viitattu 19.7.2023)

Johnson, S.P., Menor, L.J., Roth, A.V. & Chase, R.B. (2000). A Critical the Evaluation N e w Service Development of Process. Teoksessa: Fitzsimmons, J. A., & Fitzsimmons, M. J. (2000). A Critical Evaluation of the New Service Development Process: Integrating Service Innovation and Service Design. In New Service Development (pp. 1-). SAGE Publications, Incorporated. <https://doi.org/10.4135/9781452205564.n1>

Kasanen, E., Lukka, K., & Siitonen, A. (1993). The constructive approach in management accounting research. *Journal of management accounting research*, 5.

Kunnela, A. (2022). Kirjallisuuskatsaukset. Saatavilla: <https://oppimateriaalit.jamk.fi/yamk-kasikirja/kirjallisuuskatsaukset/> (Viitattu 4.6.2023)

Laitinen, E. K. (2007). *Kilpailukykyä hinnoittelulla*. Talentum.

Lindman, M., Pennanen, K., Rothenstein, J., Scozzi, B., & Vincze, Z. (2016). The value space: how firms facilitate value creation. *Business Process Management Journal*, 22(4), 736-762.

Lukka, K. (2001). Konstruktiivinen tutkimusote. Saatavilla: <https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/> (Viitattu 7.12.2023)

Lukka, K. (2006). Konstruktiivinen tutkimusote: luonne, prosessi ja arviointi. Teoksessa Rolin, K., Kakkuri-Knuuttila, M.-L., Henttonen, E., Lagerspetz, E., Kuorikoski, J., Lukka, K., Katila, S., Meriläinen, S., Vaara, E., Laine, P.-M., Moisander, J., Eräranta, K., Sappinen, J.-V., Lehtinen, A., Knuuttila, T., Paavola, S., & Hakkarainen, K. (2006). *Soveltava yhteiskuntatiede ja filosofia*. Gaudeamus. Helsinki.

Martinez, E., & Pfister, L. (2023). Benefits and limitations of using low-code development to support digitalization in the construction industry. *Automation in Construction*, 152, 104909.

Matvitskyy, O., Iijima, K., Leow, A., Vincent, P. & West, M. (2023). Critical Capabilities for Enterprise Low-Code Application Platforms. Saatavilla: <https://www.gartner.com/document/4023306?ref=solrAll&refval=368507347> (Viitattu 6.6.2023)

Microsoft. (2023). Add Copilot control to a canvas app (preview). Saatavilla: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/add-ai-copilot> (Viitattu 28.5.2023)

Misra, S., Kumar, V., Kumar, U., Fantasy, K., & Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, 29(9), 972-980.

Navakka. (2023, a). Navakka. Saatavilla: <https://navakka.com/yritys/navakka/> (Viitattu 9.8.2023)

Navakka. (2023, b). Navakka lisäarvoratkaisut. Saatavilla: <https://navakka.com/dynamics-365-business-central/navakan-lisaarvoratkaisut/> (Viitattu 9.8.2023)

Oliva, R., & Kallenberg, R. (2003). Managing the transition from products to services. *International Journal of Service Industry Management*, 14(2), 160–172.

- Parantainen, J. (2007). *Tuotteistaminen, rakenna palveluista tuote 10 päivässä*. Hämeenlinna: Kariston Kirjapaino Oy.
- Pasian, B. (2015). *Designs, Methods and Practices for Research of Project Management*. Routledge. <https://doi.org/10.4324/9781315270197>
- Phillips, J. (2016). Microsoft PowerApps and Flow are generally available starting tomorrow. Saatavilla: <https://blogs.microsoft.com/blog/2016/10/31/microsoft-powerapps-flow-generally-available-starting-tomorrow/> (Viitattu 19.9.2023)
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13, 303-337.
- Pinho, D., Aguiar, A., & Amaral, V. (2022). What about the usability in low-code platforms? A systematic literature review. *Journal of Computer Languages*, 101185.
- Rathor, S., Xia, W., & Batra, D. (2023). Achieving software development agility: different roles of team, methodological and process factors. *Information Technology & People*.
- Prahalad, C. K., & Ramaswamy, V. (2004). Co-creation experiences: The next practice in value creation. *Journal of interactive marketing*, 18(3), 5-14.
- Polyakova, O., & Mirza, M. (2015). Perceived service quality models: are they still relevant?. *The Marketing Review*, 15(1), 59-82.
- Puusa, A., Juuti, P., & Aaltio, I. (2020). Laadullisen tutkimuksen näkökulmat ja menetelmät. *Gaudeamus*.
- Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, 17(4), 26-32.
- Rosa-Bilbao, J., Boubeta-Puig, J., & Rutle, A. (2023). EDALoCo: Enhancing the accessibility of blockchains through a low-code approach to the development of event-driven applications for smart contract management. *Computer Standards & Interfaces*, 84, 103676.
- Russell-Bennett, R., Wood, M., & Previte, J. (2013). Fresh ideas: services thinking for social marketing. *Journal of Social Marketing*, 3(3), 223-238.
- Saarijärvi, H., Kannan, P. K., & Kuusela, H. (2013). Value co-creation: theoretical approaches and practical implications. *European business review*, 25(1), 6-19.
- Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 171-178). IEEE.
- Sanchis, R., García-Perales, Ó., Fraile, F. & Poler, R. (2019). Low-Code as Enabler of Digital Transformation in Manufacturing Industry. *Applied Sciences*. [Online]. 10 (1). p.p. 12.
- Salminen, A. (2011). Mikä kirjallisuuskatsaus?: Johdatus kirjallisuuskatsauksen tyypeihin ja hallintotieteellisiin sovelluksiin.

- Saunders, M., Lewis, P., & Thornhill, A. (2019). *Research Methods for Business Students*. Pearson Education, Limited.
- Schwaber, K. & Sutherland, J. (2020). *Scrum-opas, Scrumin määritelmä ja pelissäännöt*. Saatavilla: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Finnish.pdf> (Viitattu 28.6.2023)
- Shostack, G. L. (1984). Designing services that deliver. *Harvard business review*, 62(1), 133-139.
- Sipilä, J. (1995). *Asiantuntijapalvelujen tuotteistaminen*. Porvoo: WSOY.
- Sipilä, J. (2003). *Palvelujen hinnoittelu*. WSOY.
- Sundberg, L. and Holmström, J., (2023). Democratizing artificial intelligence: How no-code AI can leverage machine learning operations. *Business Horizons*.
- TENK. (2023). Hyvä tieteellinen käytäntö (HTK). Saatavilla: <https://tenk.fi/fi/hyva-tieteellinen-kaytanta-htk> (Viitattu 29.11.2023)
- Thomas, E., & Magilvy, J. K. (2011). Qualitative Rigor or Research Validity in Qualitative Research. *Journal for Specialists in Pediatric Nursing*, 16(2), 151–155. <https://doi.org/10.1111/j.1744-6155.2011.00283.x>
- Trott, P. (2008). *Innovation management and new product development* (pp. xxvii–xxvii).
- Vargo, S. L., & Lusch, R. F. (2004). Evolving to a new dominant logic for marketing. *Journal of marketing*, 68(1), 1-17.
- Vincent, P., Wong, J., Ray, S., Jain, A., Guttridge, K., Iijima, K., Leow, A. & Natis, Y., (2021). *Identify and Evaluate Your Next Low-Code Development Technologies*. Gartner.-2021.
- Vincent, P., Iijima, K., Leow, A., West, M., & Matvitskyy, O. (2022). *Magic quadrant for enterprise low-code application platforms*. Gartner report.
- Wirtz, J., Fritze, M. P., Jaakkola, E., Gelbrich, K., & Hartley, N. (2021). Service products and productization. *Journal of Business Research*, 137, 411–421. <https://doi.org/10.1016/j.jbusres.2021.08.033>
- Wu, L. W., Rouyer, E., & Wang, C. Y. (2022). Value co-creation or value co-destruction: co-production and its double-sided effect. *International Journal of Bank Marketing*, 40(4), 842-864.
- Zhou, J., Shafique, M. N., Adeel, A., Nawaz, S., & Kumar, P. (2017). What is theoretical contribution? A narrative review. *Sarhad Journal of Management Sciences*, 3(02), 261-271.

LIITE A: HAASTATTELURUNKO ASIAKASHAASTATTELUT

Low-code -teknologia

- Kertoisitko roolistasi yrityksessä?
- Miten low-code -teknologia liittyy työhösi ja miten olet ollut tekemisessä low-code -teknologian kanssa?
- Kertoisitko lisää teillä jo olemassa olevista low-code -ratkaisuista? Mikä on ollut roolisi tämän ratkaisun ympärillä?
- Mitä arvoa low-code -ratkaisut tuovat teidän liiketoimintaanne?

Yhteistyökehittäminen

- Mitä yhteistyökehittäminen tarkoittaa sinulle?
- Miten koet ketterän kehityksen piirteet low-code -kehityksessä?
- Oletko mielelläsi mukana kehittämässä low-code -ratkaisuja tekemällä niitä itse vai haluatko mieluummin olla määrittelemässä ja testaamassa ratkaisuita? (kartoitetaan minkälaisessa roolissa asiakas mielellään olisi)
- Mitä arvoa yhteistyökehittäminen tuo low-code -ratkaisuun?

Palvelun tuotteistaminen ja tarpeet

- Minkälaisia low-code -ratkaisuihin liittyviä palveluita tarvitset? Mitkä ovat odotuksesi low-code -palveluille?
- Minkälaista koulutusta organisaatiosi haluaa/tarvitsee low-code -ratkaisuihin liittyen?
- Minkälaisia tarpeita koet low-code -ratkaisuiden jatkuvissa palveluissa ja tuessa?
- Minkälainen sisäinen tuki teillä on ratkaisulle?
- Minkälaista viestintää ja viestintäkanavaa käyttäisit mieluiten eri vaiheissa? (määrittely, kehittäminen, testaaminen, tuki)
- Minkälaisen erottelun haluaisit nähdä ratkaisun työmääräarvoissa/tarjouksessa?
- Minkälainen palauteprosessi olisi mielestäsi selkein ratkaisun tuotantoon oton jälkeen?

Ratkaisu

- Millaiseksi koit jo tehdyn low-code -ratkaisun prosessin kokonaisuudessaan? (määrittely, kehitys, testaaminen, tuki)

- Mitkä osa-alueet ovat onnistuneet palveluntarjoajalta erityisen hyvin?
- Mitkä osa-alueet toivoisit erityisesti kehittyvän?

Muuta

- Haluaisitko vielä lisätä jotain, mitä en ole aiheen ympäriltä aiemmin kysynyt?
- Saako sinuun myöhemmin olla yhteydessä, jos tulee vielä lisäkysymyksiä?

LIITE B: HAASTATTELURUNKO CASE-ORGANISAATIO

Low-code -teknologia

- Kertoisitko roolistasi yrityksessä?
- Miten low-code -teknologia liittyy työhösi ja miten olet ollut tekemisessä low-code -teknologian kanssa?
- Minkälaisia low-code -ratkaisuita olet tehnyt? Kertoisitko niistä lisää. Mikä on ollut roolisi tämän ratkaisun ympärillä?
- Mitä arvoa low-code -ratkaisut ovat tuoneet asiakkaille?
- Miksi päätitte toteuttaa ratkaisun juuri low-codea hyödyntämällä?
- Minkälaisena näet low-code -teknologian tulevaisuuden organisaatiossasi?
- Minkälaisia haasteita näet low-code -teknologiaa hyödynnettäessä?
- Mitä hyötyjä näet low-code -teknologiaa hyödynnettäessä?

Yhteistyökehittäminen

- Mitä yhteistyökehittäminen tarkoittaa sinulle?
- Miten koet ketterän kehityksen piirteet low-code -kehityksessä?
- Miten koet yhteistyökehittämisen low-code -ratkaisuja tehdessä?
- Minkälaisessa roolissa näet itsesi ammattikehittäjänä low-code -ratkaisuja tehdessä?
- Ketä tiimissä tulisi olla low-code -kehityksessä? Minkälaiset roolit?
- Minkälaista yhteistyötä toivot asiakkaalta low-code -kehitykseen liittyen?
- Mitä arvoa yhteistyökehittäminen tuo low-code -ratkaisuun?

Palvelun tuotteistaminen ja tarpeet

- Minkälaisia low-code -ratkaisuihin liittyviä palveluita tarvitaan?

- Minkälaista koulutusta low-code -ratkaisuihin liittyen tulisi tarjota?
- Minkälaisia low-code -ratkaisujen jatkuvat palvelut ja tuki tulisi olla?
- Minkälaista viestintää ja viestintäkanavaa käyttäisit mieluiten eri vaiheissa? (määrittely, kehittäminen, testaaminen, tuki)
- Minkälaisen erottelun tekisit ratkaisun työmääräarvoista/tarjouksesta? Mitkä tekijät vaikuttavat tähän?
- Minkälainen palauteprosessi olisi mielestäsi selkein ratkaisun tuotantoon oton jälkeen?

Ratkaisu

- Millaiseksi koit jo tehdyn low-code -ratkaisun prosessin kokonaisuudessaan? (määrittely, kehitys, testaaminen, tuki)
- Mitkä osa-alueet ovat onnistuneet ratkaisussa erityisen hyvin?
- Missä osa-alueissa näet erityisesti kehitystarpeita?

Muuta

- Haluaisitko vielä lisätä jotain, mitä en ole aiheen ympäriltä aiemmin kysynyt?
- Saako sinuun myöhemmin olla yhteydessä, jos tulee vielä lisäkysymyksiä?

LIITE C: HEIKKO MARKKINATESTI

Yhteistyötiimin roolitus low-code -kehityksessä

- Voisitko hyödyntää tämänkaltaista mallia roolitukselle low-code -kehityksessä omassa työssäsi?
- Näkisitkö, että voisiko mallissa olla vielä jokin muu rooli? Jos kyllä, niin mikä?
- Kertoisitko näkemyksesi kuinka ymmärrettävä ja selkeä malli on?
- Muita huomioita/kehitettävää mallissa?

Yhteistyötiimi low-code -kehityksessä

- Koetko nämä vaihtoehdot relevanttina? (yhteistyötiimin kokoonpano riippuu, minkälainen tarve asiakkaalla on ja minkälaista ratkaisua lähdetään kehittämään).
- Kertoisitko näkemyksesi kuinka ymmärrettävä ja selkeä malli on?

Low-code -kehityksen vaiheet ja vastuut

- Voisitko hyödyntää tämänkaltaista mallia low-code -kehityksessä omassa työssäsi? (vaiheet ja vastuut)
- Kertoisitko näkemyksesi kuinka ymmärrettävä ja selkeä malli on?
- Muita huomioita/kehitettävää mallissa?