

Petra Kataja-aho

**TYÖNTEKIJÖIDEN KOKEMUKSIA LOW-CODE TYÖ-
KALUJEN KÄYTTÖÖNOTOSTA JA KÄYTÖSTÄ**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2024

TIIVISTELMÄ

Kataja-aho, Petra

Työntekijöiden kokemuksia low-code työkalujen käyttöönotosta ja käytöstä

Jyväskylä: Jyväskylän yliopisto, 2024, 46 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja(t): Marttiin, Pentti

Low-code teknologioilla tarkoitetaan ohjelmistokehitystyökaluja, joissa käyttäjän ei tarvitse itse juurikaan kirjoittaa koodia, vaan hän luo sovelluksen esimerkiksi visuaalisia komponentteja käyttämällä. Low-code ohjelmistokehitys on yleistynyt viime vuosina voimakkaasti, sillä sen avulla voidaan toteuttaa ratkaisuja nopeasti ja kustannustehokkaasti ja näin vastata paremmin työelämän muuttuviin tarpeisiin ja digitalisaation tuomiin haasteisiin. Sanotaan, että low-code ohjelmistokehitys sopii myös maallikoille, eikä koodauskokemusta tarvitse aloittaakseen. Tämän tutkimuksen tavoitteena on tutkia työntekijöiden kokemuksia low-code työkalujen käyttöönotosta ja käytöstä. Tutkimus on toteutettu kvalitatiivisena tutkimuksena toteuttamalla teemahaastattelut kohdeyrityksen työntekijöille, jotka käyttävät työssään low-code työkaluja. Tutkimuksen teoriaksi valittiin UTAUT malli, joka on tietojärjestelmätieteen alalla yleisesti käytetty tutkimusmalli, kun halutaan tutkia teknologian omaksumista. Aiempi tutkimus low-code aihepiirin ympärillä on vahvasti keskittynyt low-coden tekniseen määrittelyyn ja sen keskeisimpiin hyötyihin yritykselle. Tuloksista selviää, että tunnistamalla low-code työkalujen käytön omaksumiseen vaikuttavat tekijät, voidaan helpottaa työkalujen omaksumista ja käyttöä. Suorituskykyodotukset ovat yksi merkittävä tekijä, mikäli käyttäjä kokee, että työkalu tulee helpottamaan työtehtäviä, sillä on positiivinen vaikutus käyttöön. Lisäksi uuden työkalun onnistuneen omaksumisen edellytyksenä on tarjota käyttäjälle riittävää ja oikeanlaista tukea käytön alkuvaiheessa.

Asiasanat: low-code, teknologian omaksuminen, UTAUT, TAM

ABSTRACT

Kataja-aho, Petra

Employees' experiences of adopting and using low-code tools

Jyväskylä: University of Jyväskylä, 2024, 46 pp.

Information Systems, Master's Thesis

Supervisor(s): Marttiin, Pentti

Low-code technologies refer to software development tools where users do not need to write much code themselves; instead, they create applications using visual components, for example. Low-code software development has become increasingly popular in recent years as it allows for quick and cost-effective solution implementations, better addressing the changing needs of the workforce and the challenges brought by digitalization. It is said that low-code software development is also suitable for non-experts, as coding experience is not necessary to start. The aim of this study is to investigate employees' experiences of adopting and using low-code tools. The research was conducted as a qualitative study by conducting theme interviews with employees of a target company who use low-code tools in their work. The Unified Theory of Acceptance and Use of Technology (UTAUT) model was chosen as the theory for the study, which is commonly used in the field of information systems to study technology adoption. Previous research around the low-code domain has heavily focused on the technical specifications of low-code and its key benefits to the enterprise. The results reveal that by identifying factors influencing the adoption of low-code tools, the adoption and usage of the tools can be facilitated. Performance expectations are a significant factor; if users perceive that the tool will ease their tasks, it has a positive impact on usage. Additionally, successful adoption of a new tool requires providing users with sufficient and appropriate support during the initial stages of usage.

Keywords: low-code, technology adoption, UTAUT, TAM

KUVIOT

KUVIO 1. Gartner Magic Quadrant 2022 enterprise-tason low-code-alustoista, lähde: Vincent ym. (2022)	11
KUVIO 2. TAM-malli (Davis, 1989)	15
KUVIO 3. UTAUT-malli (Venkatesh ym. 2003)	17

TAULUKOT

TAULUKKO 1. Microsoft Power Platform Työkalut	12
TAULUKKO 2. Merkittäviä teknologiaan liittyviä malleja (mukaillen Rahman et. al., 2021).....	14
TAULUKKO 3 Haastateltavat	22
TAULUKKO 4 Haastateltavat ja työkalut.....	25
TAULUKKO 5 Low-code teknologian omaksumiseen vaikuttavat tekijät.....	36

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT JA TAULUKOT

1	JOHDANTO	6
2	LOW-CODE OHJELMOINTI.....	8
2.1	Low-code ohjelmistokehityksen historiaa.....	8
2.2	Low-code ohjelmistokehitys	9
2.3	Microsoft Power Platform	11
3	TEKNOLOGIAN OMAKSUMINEN	13
3.1	Määritelmä.....	13
3.2	TAM-malli.....	15
3.3	Teknologian omaksumismalli UTAUT	16
3.4	Aikaisempi tutkimus aiheesta.....	18
4	TUTKIMUSMENETELMÄ.....	20
4.1	Tapaustutkimus ja kohdeorganisaatio	20
4.2	Haastattelujoukko ja haastatteluiden toteutus.....	21
4.3	Aineiston analysointi.....	22
5	TUTKIMUKSEN TULOKSET	24
5.1	Taustaa tutkittavista	24
5.2	Teknologian omaksumiseen vaikuttavat tekijät	26
5.2.1	Käytön vaikutukset työstä suoriutumiseen	26
5.2.2	Koettu käytön vaivattomuus	27
5.2.3	Koettu sosiaalinen vaikutus.....	29
5.2.4	Käyttöä tukevien olosuhteiden vaikutus.....	30
5.2.5	Aiemman teknologian käyttöön liittyvän kokemuksen vaikutus	32
6	POHDINTA	34
6.1	Low-code työkalujen käytön omaksuminen	34
6.2	Ajatuksia kouluttajille	37
6.3	Tutkimuksen luotettavuus	38
6.4	Jatkotutkimusaiheita	38
7	YHTEENVETO.....	40
	LÄHTEET	42
	Haastattelukysymykset:	46

1 JOHDANTO

Low-code-teknologiat ja -alustat ovat viime vuosina tulleet markkinoille vauhdikkaasti ja yleistyneet voimakkaasti. Low-code kehitystyökalujen tavoitteena on vastata nykypäivän nopeatempoisen työelämän tarpeisiin, sillä niiden avulla voidaan kehittää digitaalisia ratkaisuja ketterästi ja nopeasti. Asiakkaiden tarpeisiin vastaamalla ja olemalla jopa niiden edellä, yritys voi parantaa kilpailukykyään markkinoilla. (Why low-code development matters right now, 2022.)

Low-code teknologiat helpottavat ohjelmistokehitystä monin tavoin muun muassa säästämällä aikaa, vähentämällä kustannuksia, tehostamalla tuottavuutta ja lisäämällä joustavuutta kehitysprojekteissa. Työntekijöiden ei tarvitse olla pitkän linjan koodareita ja ohjelmistokehittäjiä, voidakseen tehdä low-code kehitystä ja sen onkin sanottu olevan mahdollista jopa 'maallikoille', joilla ei ole ollenkaan kokemusta koodaamisesta.

Tämän tutkimuksen tavoitteena on selvittää, miten työntekijät ovat omaksuneet low-code työkalujen käyttöönoton ja samalla tutkia mitkä tekijät omaksumiseen ovat vaikuttaneet. Onko low-code koodauksen aloitus ollut helpompaa heille, joilla on jo useampi vuosi työuraa takana ohjelmistokehityksen parissa ja miten junioritason työntekijät ovat selvinneet uuden työkalun eli low-code ohjelmistojen käyttöönotosta.

Koska low-code on suhteellisen uusi teknologia markkinoilla, aiemmasta kirjallisuudesta ei löytynyt juurikaan vastaavia tutkimuksia. Aiempi kirjallisuus on keskittynyt low-code työkalujen tekniseen määrittelyyn sekä niiden hyötyjen arviointiin, aivan tuoreimmissa tutkimuksissa oli otettu myös kantaa low-code työkalujen negatiivisiin ominaisuuksiin. Low-code työkalujen käyttöönottoon liittyvää tutkimusta on tehty jonkin verran organisaation näkökulmasta ja näissä tutkimuksissa kartoitettu syitä miksi organisaatio on ottanut/ei ole ottanut low-code työkaluja käyttöön. Aiempaa tutkimusta, jossa työkalujen käyttöönottoa olisi tutkittu työntekijöiden omaksumisen näkökulmasta ei löytynyt.

Tämän vuoksi onkin mielenkiintoista nähdä, millaiset tekijät omaksumiseen vaikuttavat ja millaiset työntekijät teknologian parissa menestyvät ja viihtyvät. Tutkimuksen tuloksia voidaan hyödyntää muun muassa rekrytoinnissa,

jotta osataan hakea oikeanlaisia henkilöitä sekä työntekijöiden koulutuksessa. Omaksumista arvioidaan UTAUT teoriaviitekehityksen avulla, joka on kehitetty teknologian käyttöaikomusten ennustamiseen erityisesti organisaatiokontekstissa (Venkatesh ym., 2003). UTAUT-mallia on käytetty tietojärjestelmätieteen tutkimuksissa laajalti, kun on haluttu tutkia teknologian käyttöönottoa ja omaksumista.

Tässä tutkimuksessa low-code työkalujen omaksumista tutkittiin suomessa toimivassa teknologia- ja konsultointiyrityksessä ja kohdeyrityksen myötä tutkittavaksi työkaluksi valikoitui Microsoft Power Platform low-code kehitysalusta. Low-code työkaluja on käytetty yrityksessä vuodesta 2019 saakka ja tällä hetkellä Power Platformia jollain tavalla työssään hyödyntää jopa 50 henkilöä.

Tutkimuksessa pyritään vastaamaan seuraavaan tutkimuskysymykseen:

- *Mitkä tekijät vaikuttavat low-code teknologian omaksumiseen ja onnistuneeseen käyttöönottoon?*

Tutkimuksen teoriaosuudessa (luvut 2 ja 3) on käyty läpi low-code ohjelmoinnin historiaa ja käsitettä sekä tutustuttu tietojärjestelmätieteen alalla yleisesti käytettyihin teorioihin, kun halutaan tutkia teknologian omaksumista ja käyttöönottoa. Tarkemmin on esitelty TAM ja UTAUT teorialat. Varsinainen tutkimus on tehty laadullisena tutkimuksena ja tutkimusmenetelmänä on käytetty teemahaastattelua, joissa haastateltiin kohdeyrityksen työntekijöitä, jotka työskentelevät low-code työkalujen parissa. Tutkimusmenetelmä on kuvattu luvussa 4.

Luvussa 5 on esitelty tutkimuksen tulokset sekä lainauksia haastatteluista, arvioitu tutkimuksen luotettavuutta sekä esitetty jatkotutkimusideoita low-code työkaluihin liittyen. Viimeisessä luvussa (luku 6) on katsaus tutkimukseen kokonaisuutena ja tulokset sekä johtopäätökset tiivistetysti.

2 LOW-CODE OHJELMOINTI

Tässä luvussa tehdään katsaus low-code ohjelmistokehityksen historiaan ja tarkastellaan, miten se on kehittynyt nykyisen kaltaiseksi työkaluksi. Lisäksi määritellään tarkemmin low-code terminä ja tarkastellaan sen nykytilaa ja esitellään tutkimuksessa käsitelty low-code työkalu Microsoft Power Platform.

2.1 Low-code ohjelmistokehityksen historiaa

Low-code terminä esiintyi ensimmäisen kerran kirjallisuudessa vasta vuonna 2014 Forresterin julkaisemassa tutkimusraportissa, jossa Clay Richardson sitä käyttää (Sanchis ym., 2019). Raportissa low-code määritellään alustoiksi, jotka mahdollistavat liiketoimintasovellusten nopean toimituksen vähäisellä koodauksella ja vähäisillä alkuinvestoinneilla asennuksessa, koulutuksessa ja käyttöönotossa. Tämä artikkeli ei kuitenkaan ollut vertaisarvioitu. Vertaisarvioidussa kirjallisuudessa Bucaionin, Cichettin ja Ciccozzin (2022) mukaan low-code alkoi esiintyä terminä vuonna 2018. Myös Sanchis ym. (2019) mainitsee, että termi on vielä tuore kirjallisuudessa.

Sahay, Indamutsa, DiRuscio ja Pierantonio (2021) ovat määritelleet low-code ohjelmistoalustat seuraavasti. Ne ovat pilvipohjaisia ohjelmistoalustoja, jotka mahdollistavat eri alojen asiantuntemusta omaavien kehittäjien ja erilaisella teknisellä osaamisella varustettujen kehittäjien tuottaa tuotantokäyttöön valmiita sovelluksia. Sovellukset kehitetään käyttämällä MDE-periaatteita ja hyödyntäen pilvi-infrastruktuureja, automaattista koodigenerointia ja korkean tason graafisia abstraktioita. Alustat tarjotaan Platform-as-a-Service (PaaS) -palveluna ja ne varmistavat haluttujen sovellusten tehokkaan ja tehokkaan kehittämisen, käyttöönoton ja ylläpidon.

Vaikka termi on tuore, se ei tarkoita, että ilmiö olisi kuitenkaan täysin uusi. Bock ja Frank (2021) toteavat, että low-code alustat ja mallipohjainen

kehittäminen kulkevat hieman limittäin ja niiden raja on vaikea määrittää. Seuraavaksi esitellään hieman low-coden historiaa ja verrataan low-code kehitystä mallipohjaiseen kehittämiseen.

Low-code kehityksen historia juontaa juurensa 1980-luvulle, jolloin ensimmäiset graafiset käyttöliittymät ja kehitysympäristöt tulivat markkinoille. Silloin otettiin ensimmäistä kertaa käyttöön graafisia työkaluja, kuten valintaruutuja ja raahaa-liitä toimintoja ohjelmointilogiikan luomiseksi, tällaisia työkaluja olivat muun muassa 4GL ja CASE (Computer-Aided Software Engineering). Esimerkiksi CASE-työkalujen avulla voitiin luoda tietokantakaavioita, piirtää käyttötapauskaavioita, luoda tietomalleja sekä dokumentoida ohjelmistoprojekteja. (Ruscio ym., 2022)

1990-luvulla suosiotaan kasvatti Microsoftin kehittämä visuaalinen ohjelmointikieli Visual Basic. Sen toiminta perustui komponenttien raahaamiseen ja pudottamiseen. Toinen tämän ajan trendi oli RAD-työkalut (Rapid Application development), jotka nopeuttivat ohjelmistojen luomista. Tyypillisiä piirteitä kehitykselle oli prototyyppien nopea luominen ja iterointi asiakkaan palautteen perusteella. Tunnettuja RAD-työkaluja olivat esimerkiksi PowerBuilder, Delphi ja Oracle APEX.

2000-luvulla low-code alkoi yleistymään ja menttiin kohti nykyisen kaltaisia kehitysalustoja. Syntyi useita low-code alustoihin keskittyviä yrityksiä kuten OutSystems, Appian ja Mendix, joidenka tavoitteena oli mahdollistaa sovelluskehitys ilman erityistä ohjelmointikokemusta. Pinnalla oli erityisesti MDE eli malliohjattu kehittäminen. (Ruscio ym., 2022).

2010-luvulta eteenpäin low-coden nousukausi vasta toden teolla alkoi ja se sai laajempaa hyväksyntää yritysten keskuudessa. Suuret teknologiayritykset kuten Microsoft, IBM ja Salesforce investoivat low-code kehitykseen ja kehittivät omia alustoja. Organisaatiot alkoivat hyödyntää low-code alustoja sovelluskehityksessä aiempaa enemmän vastatakseen liiketoiminnan tarpeisiin kustannustehokkaasti.

2.2 Low-code ohjelmistokehitys

Viime vuosien nopea digitalisaatio on saanut aikaan pulan kokeneista ohjelmistokehittäjistä, IT-ammattilaisten kysyntä on tarjontaan nähden valtavaa ja uusien osaajien kouluttaminen hidasta sekä kallista yrityksille.

Low-code ohjelmistokehitys eli vapaasti suomennettuna matalakoodi ohjelmistokehitys pyrkii vastaamaan tähän ongelmaan tarjoten helppokäyttöisen työkalun, jossa koodia tarvitsee kirjoittaa mahdollisimman vähän tai ei ollenkaan. Tällainen työskentelytapa mahdollistaa nopeamman kehitysprosessin. (Tisi ym., 2019.) Low-code alustat yrittävät kuroa umpeen myös järjestelmävaatimusten ja käyttäjien välistä kuilua, joka syntyy, kun järjestelmävaatimukset ja käyttäjien tarpeet eivät kohta. (Alamin ym., 2022). Muita low-code työkalujen etuja on

nopeamman kehitysprosessin myötä työn kustannusten aleneminen sekä mahdollisuus integroida muita yksiköjä mukaan kehitysprosessiin (Beranic ym., 2020).

Wazkovskin (2019) mukaan low-coden tavoitteena on nopeuttaa ohjelmistokehittäjien työtä tarjoamalla graafisia ratkaisuja käyttöliittymän perustoimintojen luomiseen. Low-code kehitykseen tarkoitettuja alustoja ja palveluntarjoajia on nykypäivänä useita. Kehitysalustat ovat tavallisimmin PaaS-alustoja eli pilvipohjaisia Platform as a Service -palveluita, joita ohjelmistokehittäjä käyttää lasutettavan lisenssimaksun hinnalla. Kehitys tehdään tällöin internet-selaimen käyttöliittymässä, ilman että käyttäjän tarvitsee ladata tietokoneelle erillistä ohjelmistoa. (Wazkowski, 2019). Käyttöliittymässä on näkyvillä muun muassa graafisia työkaluja ja komponentteja sekä mahdollisuuksia yhdistää sovellus eri datalähteisiin esimerkiksi SQL-tietokantaan. Tyypillisiä pääpiirteitä low-code alustoille ovat visuaalinen mallintaminen, automatisoitu ohjelmointi, uudelleen käytettävät komponentit, low-code alustan itsenäisyys ja julkaisemisen ketteryys (Sahay ym., 2020).

Bockin & Frankin (2021) mukaan valmiit komponentit ja pohjaratkaisut aiheuttavat myös haasteita, sillä kustomointimahdollisuudet ovat niiden myötä rajalliset ja skaalautuvuus joissakin tapauksissa kallista, mikäli käyttäjien määrä on suuri, koska usein kustannukset perustuvat käyttäjälisenssien määrään.

MDA (Model-Driven Architecture) ja low-code kehitys ovat molemmat lähestymistapoja, jotka pyrkivät tehostamaan ohjelmistokehitystä ja vähentämään koodin kirjoittamisen tarvetta perinteisiin menetelmiin verrattuna. Vaikka niillä on joitain yhtäläisyyksiä, niillä on myös merkittäviä eroja. Ruscion yms. (2022) MDA keskittyy korkean abstraktion malleihin, jotka kuvaavat järjestelmän toiminnallisuutta ja rakennetta. Nämä mallit ovat riippumattomia teknologiasta ja käänösvaiheessa muunnetaan alustakohtaisiksi toteutuksiksi. Kehitysympäristöt ovat työpöytäpohjaisia, eli käyttäjä asentaa tietokoneelleen tarvittavan ohjelmiston. MDA:n parissa työskentelee kokeneita sovelluskehittäjiä ja MDA on perinteisesti keskittynyt suurempiin ja teknisempiin järjestelmiin muun muassa auto- ja energiateollisuudessa sekä kyberfyysisiin järjestelmiin. (Ruscio ym., 2022.)

Low-code kehityksessä abstraktiotaso voi vaihdella, mutta yleensä se tarjoaa pilvipohjaisen alustan, joka mahdollistaa ohjelmiston kehittämisen visuaalisesti tai konfiguraatioiden avulla ilman syvällistä koodin kirjoittamista, jolloin käyttäjät voivat olla "kansalaiskehittäjiä" tai alan asiantuntijoita. Koodia voidaan kuitenkin lisätä tarvittaessa tiettyjen toimintojen laajentamiseksi tai muokkamiseksi. Low-code kehitys on niin sanotusti kohdealueriippumatonta kehitystä, mutta erityisesti sitä on käytetty liiketoimintasovelluksissa ja IoT:ssä ja koneoppimisessa. (Ruscio ym., 2022.)



KUVIO 1. Gartner Magic Quadrant 2022 enterprise-tason low-code-alustoista, lähde: Vincent ym. (2022)

Kuviossa 1 x-akselilla esitetään yrityksen vision valmius low-code alustan tulevaisuuden ja kyvykkyyden näkökulmasta ja y-akselilla yrityksen low-code alustan kykyä luoda toimivia sovelluksia. Kuvioista voidaan nähdä, että tämän hetken johtavia palveluntarjoajia ovat OutSystem, Mendix, Microsoft, Salesforce sekä ServiceNow (Vincent, 2022). Seuraavassa alaluvussa esitellään tarkemmin Microsoftin kehittämä low-code alusta Microsoft Power Platform.

2.3 Microsoft Power Platform

Tutkimuksen kohdeyritys on Microsoft-kumppani, joten tutkimuksessa tarkastellaan low-code työkaluna Microsoft Power Platformia, jota kohdeyrityksessä käytetään. Microsoft Power Platform on yksi seitsemästä globaalista low-code alustasta (Wazkowski, 2019). Sen vahvuus markkinoilla on low-code alustan vahva skaalautumiskapasiteetti ja mahdollisuus laajaan tukipalveluun käyttäjille. Lisäksi on ennustettu, että se tulee saavuttamaan johtoaseman markkinoilla

Microsoft 365 ja Dynamics ympäristöjen nykyisten käyttäjämäärien suuruuden myötä (Vincent ym., 2022).

Power Platform tuoteperhe koostuu viidestä tuotteesta: *Power BI, Power Apps, Power Automate, Power Pages ja Power Virtual Agents*. Työkaluja voidaan käyttää joko itsenäisesti tai ne voidaan yhdistää toisiinsa tai muihin sovelluksiin kuten Microsoft 365:een, Dynamics 365:een, Azureen halutun lopputuloksen saavuttamiseksi. Power Apps tarjoaa yli 240 valmista liitintä, joiden avulla sovellusten yhdistäminen eri verkkopalveluihin on helppoa.

Power Platform -työkalut on suunniteltu kokonaisvaltaisten liiketoimintaratkaisujen luomiseen ja niillä jokaisella on omat käyttötarkoituksensa, jotka ovat esitelty lyhyesti Taulukossa 1. (Taulukko 1). (Microsoft, 2022.)

TAULUKKO 1. Microsoft Power Platform Työkalut

Työkalu	Kuvaus
Power BI	Suunniteltu erilaisten datalähteiden yhdistämiseen ja johdonmukaisten & vuorovaikutteisten raporttien muodostamiseen liiketoiminnan tueksi.
Power Apps	Käyttöliittymä, jonka avulla käyttäjä voi luoda nopeasti ja helposti sovelluksia liiketoiminnan tarpeisiin.
Power Automate	Työkalu erilaisten automaatioiden luomiseen, jolla automatisoida yrityksen prosesseja.
Power Pages	Tuoteperheen uusin tulokas, jonka avulla voi luoda nopeasti nettisivuston omaisia yrityssivustoja.
Power Virtual Agent	Työkalu asiakkaiden kanssa kommunikoivan chatbotin luomiseen esimerkiksi yrityksen nettisivuille.

Power Platformin ”ohjelmointikieli” on PowerFx, joka on kehitetty yksinomaan Power Platform työkaluja varten. PowerFx:ä luodessa on hyödynnetty eityisesti Microsoft Excelissä käytettyjä kaavoja ja ominaispiirteitä, jonka ansiosta alustaa käyttävät henkilöt voivat hyödyntää aiempaa osaamistaan Power Platformin parissa helpommin (Voronkov & Saradgishvili, 2021).

3 TEKNOLOGIAN OMAKSUMINEN

Tässä luvussa tarkastellaan teknologian omaksumista ja käyttöönottoa yleisellä tasolla, esitellään informaatioteknologia-alan yleisiä teknologian omaksumisteorioita sekä tarkastellaan aiempaa tutkimusta teknologian omaksumisesta. Luku esittelee tarkemmin teknologian omaksumismalli UTAUT:n, jonka avulla on mahdollista ymmärtää mitkä tekijät vaikuttavat uuden teknologian, tässä tutkimuksessa low-coden teknologian omaksumiseen.

3.1 Määritelmä

Venkateshin (2003) mukaan teknologian omaksuminen voidaan määritellä seuraavasti: käyttäjä hyväksyy ja ottaa käyttöönsä uuden toimintamallin tai teknologian. Uuden teknologian omaksuminen on prosessi, joka etenee teknologian hyväksymisestä käytön ja sopeutumisen kautta muutokseen yksilön tai organisaation toimintamalleissa.

Uusi teknologia on käytännössä arvoton niin kauan, kuin kohderyhmä ei ole omaksunut sitä. Tästä syystä tietojärjestelmätieteen yksi tutkituin aihealue on juuri teknologian omaksumisen ymmärtäminen ja sitä varten onkin kehitetty useita teorioita (Venkatesh ym., 2003). Teknologian omaksuminen ei kiinnosta ainoastaan tutkijoita, vaan siihen vaikuttavien tekijöiden ymmärtäminen on erittäin tärkeää myös innovaatioiden ja teknologian kehittäjille sekä organisaatioiden teknologioista ja järjestelmistä vastaaville henkilöille (Dillon & Morris, 1996). Ymmärrys on tärkeää sen vuoksi, että organisaatiot investoivat kasvavassa määrin informaatioteknologiaan (Ali ym., 2015).

Taulukossa 2 on listattu merkittäviä teknologian omaksumiseen liittyviä malleja. Seuraavissa alaluvuissa (3.2 ja 3.3) esitellään niistä tarkemmin TAM-malli ja UTAUT-malli.

TAULUKKO 2. Merkittäviä teknologiaan liittyviä malleja (mukaillen Rahman et. al., 2021)

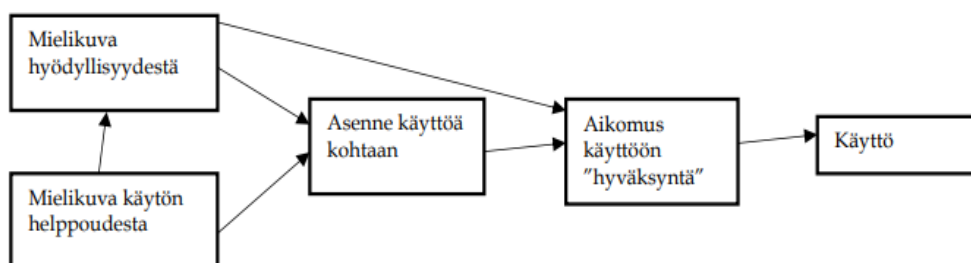
Teoria/ Malli	Tekijä(t), vuosi	Määritelmä
Innovaation diffuusioteoria (IDT)	Rogers, 1962	Malli kuvaa innovaatioon liittyvää päätöksentekoprosessia.
Perustellun toiminnan teoria (TRA)	Fishbein ja Ajzen, 1975	Muunneltava käyttäytymisteoria, joka tarjoaa mallin asenteen ja käyttäytymisen välisille suhteille.
Sosiaalinen kognitiivinen teoria (SCT)	Bandura, 1989	Oppimisteoria, joka perustuu ajatukseen, että ihmiset oppivat havainnoimalla muita.
Tknologian hyväksymismalli (TAM)	Davis, 1989	Psykologisiin tekijöihin perustuva malli, joka vaikuttaa teknologian hyväksymiseen
Laajennettu teknologian hyväksymismalli (TAM2)	Venkatesh ja Davis, 2000	TAM2-malli sisältää kaksi teoreettista prosessia, jotka kattavat sosiaalisen vaikutuksen prosessit ja kognitiiviset instrumentaaliset prosessit. Näillä selitetään eri määrävien tekijöiden vaikutukset koettuun hyödyllisyyteen ja käyttäytymisaikomukseen.
Suunnitellun käyttäytymisen teoria (TPB)	Ajzen, 1991	Teoria pyrkii ennustamaan yksilön aikomusta ryhtyä käyttäytymään tietyllä hetkellä ja paikassa.
Tietokoneiden hyödyntämismalli (MPCU)	Thompson, Higgins ja Howell, 1991	Malli ennustaa käyttäytymistä liittyen tietokoneen käyttöön
Motivaatiomalli (MM)	Davis, Bagozzi ja Warshaw, 1992	Mallia käytetään laajalti psykologian tutkimuksissa
TAM-mallin ja TPB-teorian yhdistelmä (C-TAM-TBP)	Taylor ja Todd, 1995	Malli määrittelee sosiaalisten ja hallintatekijöiden vaikutuksen, jotka eivät ole läsnä TAM:ssa, mutta ovat läsnä TPB:ssä.
Teknologian hyväksymisen ja käytön yhtenäinen teoria (UTAUT)	Venkatesh, Morris, Davis ja Davis, 2003	Mallin tavoitteena on selittää käyttäjien aikomukset käyttää tietojärjestelmää ja aikomuksen/käytön jälkeistä käyttäytymistä.
Teknologian hyväksymisen ja käytön yhtenäinen teoria 2 (UTAUT2)	Venkatesh, Thong ja Xu, 2012	UTAUT2 laajentaa UTAUT-mallia rakenteilla, jotka selittävät paremmin uusien teknologioiden omaksumista ja kuluttajakäyttäytymistä.

3.2 TAM-malli

Teknologian omaksumismalli (*Technology Acceptance Model*) eli TAM on 1980-luvulla kehitetty tietojärjestelmätieteen alalla laajasti käytetty malli, kun tarkastellaan teknologian käytön ennustamista ja selittämistä koskevia tutkimuksia. Sen avulla on tutkittu mitkä tekijät/mikä saa käyttäjän hyväksymään tai hylkäämään tutkittavan tietotekniikan tai sovelluksen. (Davis, 1989.) Legris ym. (2003) määrittelee mallin seuraavasti: TAM tarkastelee havaittua helppokäyttöisyyttä ja havaittua hyödyllisyyttä suhteessa järjestelmän ominaisuuksiin (ulkoiset muuttujat) ja todennäköisyyteen käyttää järjestelmää. TAM malli on yksi avainteorioita informaatioteknologian omaksumisen ja käytön ymmärtämisessä sekä ennustamisessa (Maranguinic & Granic, 2015). King & Hen (2006) mukaan mallin suosio ja laaja käyttö selittyy sen yksinkertaisuudella.

TAM mallin lähtökohtana on kaksi muuttujaa, jotka vaikuttavat tietojärjestelmän käyttöön. Niistä ensimmäinen on mielikuva hyödyllisyydestä (*Perceived Usefulness*), jonka mukaan ihmiset käyttävät tietojärjestelmää, mikäli he uskovat sen auttavan suoriutumaan työtehtävistään paremmin tai helpottavan niitä. Davis (1989) on määritellyt toiseksi muuttujaksi helppokäyttöisyyden (*Perceived Ease of Use*), joka on myös tärkeässä roolissa, kun tarkastellaan ihmisten asenteita tietojärjestelmää kohtaan. Vaikka käyttäjät kokisivat tietojärjestelmän hyödylliseksi työn sujumisen kannalta, mutta uskovat sen käytön olevan liian vaikeaa, järjestelmän käyttökelpoisuus kärsii eikä se ehkä pääse niin laajasti käyttöön. (Davis ym., 1989.)

Toisaalta Davisin (1989) mukaan käyttäjät voivat sietää vaikealta tuntuvaa järjestelmää, jos sen työtä helpottava hyöty on riittävän suuri. On myös otettava huomioon, että käyttäjien arvioima hyöty voi poiketa siitä mitä järjestelmästä saatu todellinen hyöty on, sillä he voivat arvioida sen suuremmaksi tai pienemmäksi kuin mitä se todellisuudessa on. TAM-malli on esitetty kuviossa 2.



KUVIO 2. TAM-malli (Davis, 1989)

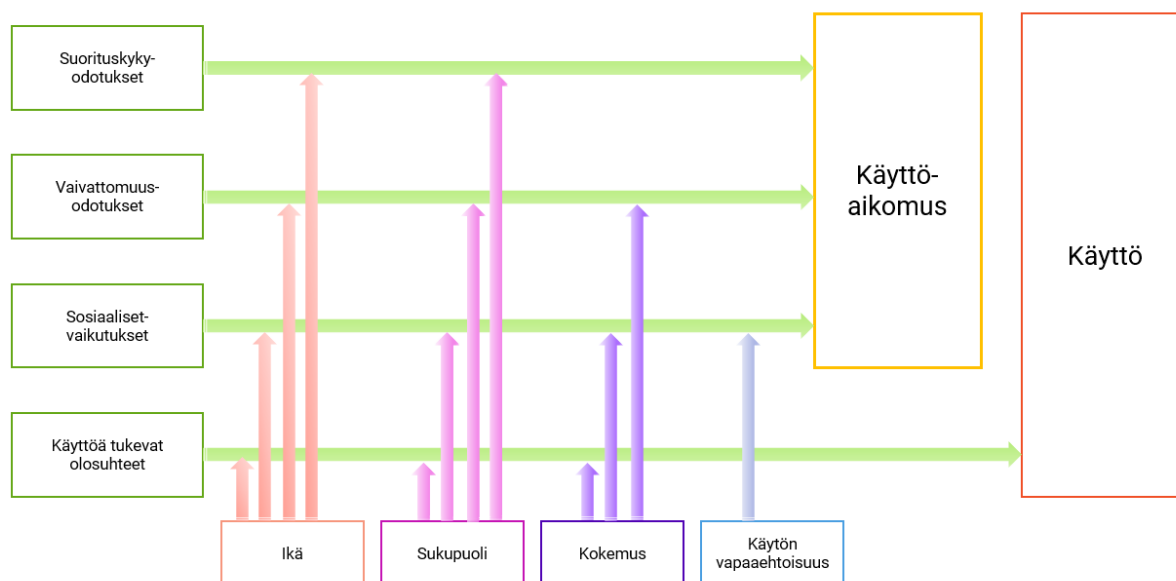
Kuten kuviosta 2 nähdään, teorian mukaan mielikuva käytön helppoudesta vaikuttaa sekä mielikuvaan hyödyllisyydestä sekä asenteeseen, kun taas mielikuva hyödyllisyydestä vaikuttaa sekä asenteeseen että käyttöaikomukseen ja lopulta käyttöön (Davis ym., 1989).

3.3 Teknologian omaksumismalli UTAUT

UTAUT-malli (*United Theory of Acceptance and Use of Technology*) eli yhdistetty teoria teknologian omaksumisesta ja käytöstä on muodostettu yhdistelemällä luvun 3.1 taulukossa 2 esitetyt teknologianomaksumismallit yhtenäiseksi teoriaksi. Mallin tavoitteena on auttaa aiempaa paremmin ymmärtämään teknologian hyväksymiseen ja käyttöönoton onnistumiseen vaikuttavia tekijöitä. (Venkatesh, 2003.) UTAUT mallin avulla voidaan arvioida, miten seuraavat tekijät vaikuttavat yksilöiden tai ryhmien päätökseen hyväksyä tai käyttää uutta teknologiaa.

UTAUT-mallin perustana ovat neljä päätekijää, joita ovat: suorituskykyodotukset (*Performance Expectancy*), vaivattomuusodotukset (*Effort Expectancy*), sosiaaliset vaikutukset (*Social Influence*) ja käyttöä tukevat olosuhteet (*Facilitating Conditions*). Kolme ensimmäistä päätekijää vaikuttaa teknologian käyttöaikomukseen ja viimeinen tekijä eli käyttöä tukevat olosuhteet vaikuttavat suoraan käyttöön. UTAUT-mallissa neljän päätekijän lisäksi keskeisiä ovat neljä moderaattoria, jotka kuvaavat yksilön ominaisuuksia: käyttäjän kokemus informaatio-teknologian käytöstä (*Experience*), sukupuoli (*Gender*), ikä (*Age*) ja järjestelmän käytön vapaaehtoisuus (*Voluntariness of Use*). (Venkatesh, ym. 2003.)

Kuviossa 3 esitellään visuaalisesti tekijöiden ja moderaattorien vaikutavuus suhteessa toisiinsa.



KUVIO 3. UTAUT-malli (Venkatesh ym. 2003)

Suorituskykyodotukset viittaavat yksilön arvioon siitä, kuinka paljon uuden teknologian, esimerkiksi tietojärjestelmän käyttö helpottaa työn tekoa tai vaikuttaa kykyyn saavuttaa tiettyjä tavoitteita. Suorituskykyodotukset arvioivat sitä, miten uusi teknologia parantaisi työn tehokkuutta tai auttaisi suoriutumaan työstä. Sitä voidaan mitata useilla mittareilla, joita ovat muun muassa nopeus, tarkkuus ja työn laatu. Aiempi kokemus samankaltaisista teknologioista, koulutus ja yleinen asenne teknologiaa kohtaan voivat vaikuttaa henkilön suorituskykyodotuksiin. Jos asenne on positiivinen, on todennäköisempää, että hän kokee teknologian hyödyllisenä. (Venkatesh, ym. 2003.)

Vaivattomuusodotuksilla viitataan yksilön kokemukseen järjestelmän käytöstä; onko se helppoa vai vaikeaa. Yksilön vaivattomuusodotukset kuvaavat sitä, miten paljon hän joutuu näkemään vaivaa uuden teknologian käyttöönottoon ja oppimiseen. Siihen vaikuttaa muun muassa käyttäjän taitotaso, resurssit ja saatavilla oleva tuki. Helpommin käytettävät teknologiat ovat todennäköisemmin hyväksytyjä. Koettuun helppokäyttöisyyteen vaikuttaa esimerkiksi käyttöliittymän selkeys, käytettävyyden taso ja koulutuksen sekä tuen saatavuus. (Venkatesh, ym. 2003.)

Sosiaalisella vaikutuksella tarkoitetaan kollegoiden, esihenkilön, perheen tai ystävien vaikutusta uuden teknologian käyttöönottoon ja hyväksymiseen. Erilaiset sidosryhmät voivat luoda painetta teknologian käyttöön eri tavoin. Tässä tutkimuksessa tarkastellaan lähinnä esihenkilön tai kollegoiden vaikutusta käyttöön, kun asiaa tutkitaan organisaatiokontekstissa. Organisaatiossa

käyttöönottoa voidaan edistää, mikäli organisaatio tukee käyttöä tai kollegat suosittelevat sitä. (Venkatesh, ym. 2003.)

Käyttöä tukevat olosuhteet käsitteellä viitataan yksilön arvioon siitä, miten hyvin organisaatio tukee teknologian käyttöönottoa ja käyttöä tarjoamalla tarvittavat resurssit, koulutuksen ja infrastruktuurin. Käyttöä tukevia osatekijöitä ovat muun muassa saatavilla oleva tekninen tuki, mahdollisuus saada opastusta, tarvittavien laitteiden ja ohjelmistojen saatavuus, organisaation ilmapiiri sekä teknologian käyttäjäystävällisyys. Jos yksilö kokee nämä resurssit ja tuen puutteellisena, se voi vaikuttaa kielteisesti teknologian omaksumiseen ja käyttöön. (Venkatesh, ym. 2003.)

UTAUT-mallista on myöhemmin, vuonna 2012, kehitetty laajennettu UTAUT2-malli, joka käsittelee alkuperäisen mallin lisäksi myös hedonistista motivaatiota, hinta-arvo -suhdetta sekä tottumuksia, jotka vaikuttavat uuden teknologian omaksumiseen. Malli ottaa huomioon myös kulttuuriset tekijät sekä uskomukset ja normit. (Venkatesh, Thong ja Xu, 2012). Kyseinen malli sopii erityisen hyvin kuluttajakäyttäjryhmän tutkimiseen. Sen vuoksi, että tässä tutkimuksessa tarkastellaan työntekijöiden näkökulmaa organisaatiossa, on teoriaviitekehukseksi valittu kuitenkin alkuperäinen UTAUT-malli.

3.4 Aikaisempi tutkimus aiheesta

Tässä luvussa esitellään muutamia low-code teknologiaan liittyviä tutkimuksia viime vuosilta. Teknologian omaksumista on tutkittu tietojärjestelmätieteen alalla paljon eri konteksteissa, low-code teknologiana on kuitenkin vielä varsin tuore, joten sen vuoksi juuri sitä koskevia tutkimuksia on toistaiseksi verrattain vähän saatavilla. Low-code teknologian käyttöönottoa organisaation näkökulmasta tarkastelevia tutkimuksia löytyi, mutta työntekijöiden näkökulmasta tarkasteltuna ei juurikaan tullut vastaan.

Pinhon, Aguiarin ja Amaralin (2023) artikkeli tarkastelee low-coden käsitettä ja low-code työkalujen käytettävyyttä, erityisesti käyttäjien näkökulmasta. Tutkimus on suoritettu kirjallisuuskatsauksena 38 vertaisarvioidun artikkelin pohjalta. Pinhon ym. (2023) mukaan low-codelle ei löydy yksiselitteistä määritelmää, vaikka yhteisiä piirteitä on tunnistettu. Tunnettuja piirteitä low-code työkaluille ovat raahaa ja pudota -toiminnot, tarve kirjoittaa vähän koodia sekä visuaaliset työkalut. Johtopäätöksissä ehdotetaan yhtenäisen määritelmän muodostamista, joka helpottaisi jatkotutkimusten tekoa. Tutkimus korostaa, että käyttäjillä on hyvä käsitys ja odotukset low-code työkalujen käytettävyydestä. He keskittyvätkin erityisesti työkalujen käytettävyyden arvioimiseen teknisten ominaisuuksien sijasta, sillä tavallisesti low-code työkalujen käyttäjä ei ole teknisesti niin osaava henkilö. Tutkimus toteaa, että jatkotutkimus low-code työkalujen käytettävyydestä voisi edistää low-code työkalujen kehitystä. (Pinho ym., 2023.)

Beranic, Rek ja Heričko (2020) ovat tutkineet asenteita low-code työkalujen käyttöön slovenialaisissa liiketoimintaorganisaatioissa sekä maisteriopiskelijoiden näkemyksiä työkalun käytettävyydestä käytännön kokemusten perusteella. Tarkoituksena on ollut selvittää low-code työkalujen hyväksyntää, käyttöönottoa ja käytettävyyttä sekä ymmärtää millaisia odotuksia ja kokemuksia siihen liittyy.

Tehdyn kyselytutkimuksen mukaan käyttöönottoprosentti Slovenian organisaatioissa on alhainen (10 %). Syitä tähän on muun muassa epäily siitä, että low-code teknologia ei sovellu heidän organisaationsa tarpeisiin tai että sillä ei voida rakentaa riittävän monimutkaisia ja liiketoiminnan kannalta kriittisiä järjestelmiä. Beranic ym. (2020) mainitsee myös, että tutkimukseen osallistuneiden organisaatioiden vastaajat kertovat, että heillä ei ole riittävästi tietoa low-code työkaluista ja niiden mahdollisuuksista. Lisäksi halutaan säilyttää joustavuus ja valinnanvapaus eri työkalujen suhteen eikä siksi sitoutua yhteen low-code työkalun palveluntarjoajaan. Tutkimuksen tulosta on vertailtu globaaleihin tutkimuksiin ja todettu, että käyttöönottoprosentti on alhaisempi Sloveniassa mitä muualla. Positiiviset odotukset työkalun suhteen viittaavat kuitenkin mahdolliseen kasvuun.

Lisäksi samassa tutkimuksessa tutkittiin maisteriopiskelijoiden näkemyksiä Microsoft Power Apps työkalun käytettävyydestä lyhyen demokäytön jälkeen. Opiskelijoiden vastauksista voidaan todeta, että se on helppokäyttöinen ja intuitiivinen käyttöliittymän kehittämistyökalu ja sopii sovelluksien nopeaan toteuttamiseen.

4 TUTKIMUSMENETELMÄ

Tässä luvussa käsitellään pro gradu -tutkielman aineistonkeruu- ja analyysimenetelmiä. Ensimmäisessä alaluvussa käydään läpi tutkimusmenetelmä ja perustelut sen valinnalle sekä esitellään kohdeorganisaatio. Toinen alaluku käsittelee haastateltavia ja kolmas alaluku käsittelee itse haastatteluita. Viimeinen, eli neljäs alaluku, käsittelee tulosten analysointia.

4.1 Tapaustutkimus ja kohdeorganisaatio

Tutkimusmenetelmänä tässä pro gradu -tutkielmassa käytetään tapaustutkimusta, jossa aineiston keruu toteutetaan laadullista tutkimusotetta käyttäen teemahaastatteluin. Tapaustutkimus koettiin sopivaksi tutkimusmenetelmäksi tähän tutkimukseen, koska tavoitteena ei ole mitata tai yleistää asioita, vaan tutkia syvällisesti valittua ilmiötä, ja laajentaa ymmärrystä niistä tekijöistä, jotka vaikuttavat uuden teknologian käyttöönottoon ja omaksumiseen. (Tapaustutkimus, 2015.)

Haastattelutavaksi valittiin teemahaastattelu, joka on tyyliltään strukturoidun ja ei-strukturoidun haastattelun välimaastosta. Haastatteluun on mietitty ennalta teemat, joista keskustellaan, mutta se antaa tilaa myös teemojen ulkopuolisille ajatuksille ja huomioille. Kysymyslistaa ei pidä määritellä liian tiukasti etukäteen, jotta luonteva keskusteluyhteys haastattelijan ja haastateltavan välillä säilyy ja voidaan löytää jotain uutta ja poikkeavaakin tietoa aiheesta. Tämänkaltaisen haastattelu sopii tutkimuksiin, joissa halutaan saada tietoa tuntemattomasta aiheesta ja syventää saatuja vastauksia haastattelun edetessä. (Hirsjärvi & Hurme, 2000.)

Tapaustutkimus on laajasti käytetty menetelmä liiketalouden ja teknisten tieteiden aloilla. Se sopii erityisen hyvin tutkimusmenetelmäksi silloin, kun tutkimuskohteena on itsenäinen kokonaisuus kuten yritys tai muu hallinnollinen

organisaatio ja valitusta tutkimusaiheesta halutaan saada syvällistä tietoa. (Aaltio, 2022.)

Kohdeyritys Fellowmind on kansainvälinen teknologia- ja konsultointitalo, jonka toiminnan keskiössä on tukea asiakkaita heidän liiketoimintansa kehittämisen eri vaiheissa. Fellowmind on Microsoftin kumppaniyritys, mikä tarkoittaa, että käytössä on Microsoftin laaja tuotetarjooma, jonka avulla kehitetään asiakkaille parempia toiminnanohjausjärjestelmiä (ERP), data- ja analytiikkaratkaisuja, asiakkuudenhallintajärjestelmiä (CRM), pilvipalveluiden IT-infraa sekä sovelluskehitystä. Holistisuus on keskeisessä roolissa, kun tarkastellaan yrityksen arvoja ja tavoitteena onkin olla asiakkaille kokonaisvaltainen yhteistyökumppani, joka tukee eri liiketoiminnan osa-alueilla sen sijaan, että asiakas joutuu ostamaan IT-palveluita usealta eri toimittajalta.

Fellowmind toimii Suomen lisäksi viidessä maassa: Ruotsissa, Tanskassa, Hollannissa, Saksassa ja Puolassa, joissa noin 1800 asiantuntijaa muodostaa kattavan osaajaverkoston asiakkaiden käyttöön. Suomessa yrityksellä on toimisto yhdeksässä eri kaupungissa. Asiakkaita on useilta eri toimialoilta, joita on muun muassa valmistava teollisuus, energia-ala, vähittäiskauppa, rahoitusala, autoteollisuus ja julkinen sektori sekä terveydenhuolto.

Microsoft Power Platformin parissa on yrityksessä työskennellyt vuodesta 2019 saakka, jolloin se julkaistiin. Tätä ennen käytettiin jo PowerBI- ja Azuren Logic Appseja, vuodesta 2015 saakka. Osaamista on kasvatettu siten, että tänä päivänä low-code tekijät muodostavat yrityksessä oman tiimin, jossa tällä hetkellä työskentelee 15 henkilöä. Muissa yksiköissä on CE- ja F&O-konsultteja 15 henkilöä ja integraatioihin keskittyviä Logic Appsien kanssa toimivia henkilöitä 20. Eli low-code työkalujen parissa jollain lailla toimivia työntekijöitä on noin 50 tällä hetkellä.

Low-code ratkaisuja myydään asiakkaille helpottamaan heidän liiketoimintansa ja jokapäiväisessä työssä esiintyviä haasteita. Automatisoimalla prosesseja tai rakentamalla Power Appsilla toteutettu canvas app helpottamaan työprosesseja, voidaan tehostaa työntekoa ja parantaa työtyytyväisyyttä ns. pienillä teoilla.

4.2 Haastattelujoukko ja haastatteluiden toteutus

Haastateltavat ovat kohdeyrityksen työntekijöitä, jotka työskentelevät erilaisten low-code työkalujen parissa. Haastateltavat henkilöt omaavat erilaiset taustat ja kokemuspohjat, eli joukossa on jo pidemmän työuran IT-alalla tehneitä kuin myös vasta hiljattain aloittaneita työntekijöitä. Haastateltavia henkilöitä oli yhteensä seitsemän, heistä 2 oli naisia ja 5 miehiä. Keski-ikä oli 31,57 vuotta, nuorimman haastateltavan ollessa 28 ja vanhimman 38 vuotta. Taulukkoon 1 on koottu haastateltavien taustatiedot eli ikä, sukupuoli ja työkokemus low-code työkalujen parissa.

TAULUKKO 3 Haastateltavat

Tunnus	Ikä	Sukupuoli	Low-code kokemus
H1	34	Nainen	2 vuotta
H2	38	Mies	5 vuotta
H3	29	Mies	7 vuotta
H4	30	Mies	8 kuukautta
H5	29	Mies	2 vuotta
H6	33	Nainen	1 vuosi
H7	28	Mies	2 vuotta

Haastatteluissa on tarkoitus keskustella ennalta suunniteltujen teemojen avulla tutkimuksen aihepiireistä. Tavoitteena on pitää keskustelu luontevana, eikä teemoja ole pakko käsitellä ennalta määritetyssä järjestyksessä, koska usein saadaan esille syvällisempää tietoa ja poikkeavampia näkemyksiä, mikäli keskustelu ei ole liian johdattelevaa.

Haastattelut suoritettiin 2023 marraskuussa ja ne toteutettiin etänä käyttäen Microsoft Teams työkalua, jolla ne myös nauhoitettiin litterointia varten. Jokaiselle haastattelulle oli varattu aikaa 30 minuuttia, haastatteluiden pituus vaihteli 16–30 minuutin välillä. Kaikki haastattelut tehtiin suomeksi.

Haastatteluissa käytiin läpi liitteistä löytyvä haastattelurunko, jossa on kysymyksiä jokaisesta teemasta. Haastattelukysymyksiä ei välttämättä kysytty suunnitellussa järjestyksessä riippuen keskustelun etenemisestä, haastattelun lopussa varmistettiin, että kaikki teemat tulivat käsitellyksi.

4.3 Aineiston analysointi

Haastatteluiden jälkeen haastatteluaineisto litteroitiin sanatarkasti luonnolliselle kielelle ja poistettiin ylimääräiset täytesanat ja äännähdykset sekä ylimääräinen toisto sekä jätettiin litteroimatta keskustelu, joka ajautui sivuraiteille teemaan nähden. Näin saatiin aikaiseksi helppolukuinen aineisto muuttamatta kuitenkaan vastausten sisältöä. Tämän jälkeen litteroitu teksti värikoodattiin siten, että jokaiseen teemaan liittyvät vastaukset ja keskustelut korostettiin omalla värillään. Näin vastauksia eri haastateltavien välillä on helpompi verrata toisiinsa, kun voidaan yhdellä silmäyksellä nähdä mitä teemasta on keskusteltu.

Teemahaastattelun avulla tehdyissä tutkimuksissa haastattelurungon teemat toimivat usein myös analysointivaiheessa teemoitteluprosessin teemoina. (Tuomi & Sarajarvi, 2009.)

5 TUTKIMUKSEN TULOKSET

Tässä luvussa käsitellään tutkimuksen tulokset ja tarkemmin tutkittavien taustatietoja. Tarkastelun kohteena tutkimuksessa on Microsoft Power Platform low-code työkalut, joidenka omaksumista tutkittiin kohdeyrityksessä. Teknologian omaksumista tutkitaan UTAUT-mallin näkökulmasta hyödyntäen mallissa käytettyjä teknologian omaksumiseen vaikuttavia tekijöitä.

5.1 Taustaa tutkittavista

Haastatteluita tehtiin yhteensä seitsemän. Tutkimusjoukosta miehiä oli viisi ja naisia kaksi. Keski-ikä oli 31,57 vuotta, nuorimman haastateltavan ollessa 28 ja vanhimman 38 vuotta. Haastateltavilla oli erilaisia koulutustaustoja, osa heistä oli opiskellut esimerkiksi tietojenkäsittelyä tai muuta IT-alalle soveltuvaa koulutusta, mutta eivät kaikki. Kokemusta low-code työkalujen käytöstä oli vaihtelevasti 8 kuukautta - 7 vuotta. Keskiarvo kokemuksen määrästä oli 2,81 vuotta. Haastateltavat käyttävät power appsin työkaluja vaihtelevasti riippuen heidän työroolistaan ja työtehtävistään.

TAULUKKO 4 Haastateltavat ja työkalut

Tunnus	Ikä	Low-code kokemus	Työkalut
H1	34	2 vuotta	Canvas App, Power Automate
H2	38	5 vuotta	Canvas App, Power Automate, Model-Driven App
H3	29	7 vuotta	Kaikki ja AI:n hyödyntäminen näissä
H4	30	8 kuukautta	Canvas App, Power Automate, Model-Driven App
H5	29	2 vuotta	Canvas App, Power Automate
H6	33	1 vuosi	Canvas App, Power Automate, Power Pages
H7	28	2 vuotta	Canvas App, Power Automate, Power Automate Desktop (RPA)

Taulukon 4 perusteella voidaan todeta, että Power Platformin tarjoamia low-code työkaluja käytetään monipuolisesti ja niistä olivat haastattelujoukossa edustettuna kaikki muut paitsi Power Virtual Agents. Teoriaosuudessa mainittu Power BI, jonka Microsoft listaa kuuluvaksi Power Platformin alle on kohdeyrityksessä eriytetty Data & Analytiikka yksikköön, joten sen vuoksi Power BI käyttäjiä ei tutkimusjoukosta löydy. Haastateltavat eivät olleet kukaan käyttäneet kunnolla toisen palveluntarjoajan low-code työkaluja, kaksi heistä oli hieman kokeillut, mutta eivät mieltäneet niitä kuitenkaan kovin tutuiksi. Osan työrooliin sisältyy myös muuta kuin pelkästään low-code tekemistä:

H3: No mun työtehtävät on tällä hetkellä meillä aika pitkälle erilaisia tarjoaman kehityspalveluja liittyen low-codeen sekä nyt tähän generatiiviseen AI:hin (tekoälyyn), että kuinka se saadaan myöskin tähän mukaan. Mutta se liittyy myös tähän low-code kehitykseen.

H5: Kyllä mä oon välillä tehnyt pientä koodausta esimerkiksi jotain plugareita, käyttänyt C# ja JavaScriptiä sekä FO:ssa selviteltyt asioita. Vaikka aika lailla kaikki liittyy Power Apps tekemiseen kuitenkin.

5.2 Teknologian omaksumiseen vaikuttavat tekijät

Tässä luvussa käydään läpi haastatteluvastaukset ja tarkastellaan haastateltavien käsityksiä ja kokemuksia teknologiasta ja teknologian omaksumisesta sekä näihin liittyvistä haasteista. Haastatteluvastauksia peilataan UTAUT-malliin, joka tutkimuksen teoriaosuudessa on esitelty. Tulokset on jaettu viiteen alalukuun UTAUT-mallista löytyvien teknologiaan vaikuttavien tekijöiden perusteella. Yksi malliin kuuluva tekijä *käytön vapaaehtoisuus* on pudotettu pois, sillä haastateltavien työrooleihin kuuluu tämän teknologian parissa työskentely, joten vapaaehtoisuuden arviointi ei tuo lisäarvoa tutkimukselle.

5.2.1 Käytön vaikutukset työstä suoriutumiseen

Kaikki haastateltavat olivat sitä mieltä, että low-code työkalujen käyttö helpottaa työn tekoa. Erityisesti työkalun mahdollistama nopeus sovelluskehityksessä mainittiin useaan otteeseen.

H6: Kyllä se on myös nopeampaa. Ei tarvitse kirjoittaa koodia sillain sana kerrallaan.

H7: Kyllähän ne nopeuttavat työntekoa mikä monesti madaltaa kynnyistä tehdä joku ratkaisu erityisesti jatkokehityksessä, jossa tarpeet ovat usein pieniä. Esimerkiksi power automate on tosi nopea saada pystyyn, kun ei tarvitse sitä koodia kaiken.

Sen lisäksi, että manuaalinen työ vähenee, kun käytetään valmiita komponentteja koodin kirjoittamisen sijaan myös datan haku ja käsittely on helpompaa sekä nopeaa vastaajien mielestä.

H1: Esimerkiksi kun käytetään canvas appeja Microsoftin sisällä, niin on todella helppoa päästä kaikkeen dataan käsiksi. Lisää vaan oikean tietokannan tai muun datalähteen ja sen jälkeen voit käyttää ja näyttää dataa visuaalisesti canvas appissa. On ihan eri asia tehdä se datan hakeminen manuaalisesti versus näitä valmiita komponentteja käyttäen.

H5: No siis parhaimmillaan se menee niin, että jos tekee canvas appin, niin siinä on valmiita palikoita, joiden tekeminen voisi viedä monta päivää tai viikkoja. Nyt

ne käytännössä valmiina siinä, ihan mihin vaan appiin niitä haluaa käyttää. Esimerkiksi datan haku ja päivittäminen manuaalisesti on ihan hirveän iso työ, mutta canvas appissa saat parissa sekunnissa semmoisen integraation ja näkymän, jolla voit näyttää ja muokata dataa. Eli kyllä se sillä tavalla todella nopeasti parhaimmillaan toimii.

Lisäksi low-code työkalujen hyötynä mainittiin kokonaisuuden hallinta, joka on hyvin pitkälle automatisoitu Microsoft Power Platformissa.

H4: Kyllä se kokonaisuudenhallinta on myös plussa. Kun tehdään laajoja toteutuksia, on helppo tehdä muutoksia sinne väliin ja kaikki toimii sen hallinnan kanalta.

5.2.2 Koettu käytön vaivattomuus

Käyttöönoton ja käytön vaivattomuus jakoi mielipiteitä haastateltavien kesken. Osalle käyttöönotto oli ollut helppoa ja toisille hieman haastavampaa. Suurin osa mainitsi, että itse työkalun käyttö on helppoa, mutta haasteita tuo sen rajoitukset ja/tai rajalliset mahdollisuudet tehdä asioita ja erilaiset asiakaskohtaiset ympäristöt, joissa niitä käytetään.

H1: Se on ollut itselle uusi maailma, niin aluksi oli helppoa tehdä sellaista visuaalista, lisätä tuollainen palikka ja vaihtaa fonttia tai väriä. Ihan heti ei ymmärtänyt niitä logiikoita siellä taustalla.

H2: Kyllä se käyttöönotto oli mulle helppoa, toki siellä nyt on ne omat haasteet ja rajoitteet, mutta sitten täytyy elää niiden kanssa ja tavallaan siinä tulee sitten luovuudelle tilaa, jos pitää miettiä miten joku voidaan tehdä toisella tavalla.

H3: Sillon kun lähdin tekemään ensimmäisiä canvas appeja, niin ensimmäiset haasteet tuli siinä, että miten mä kirjoitan näitä nimenomaan low-code lauseita sinne sisälle ja yhdistän ne liiketoimintajärjestelmän datatyyppeihin.

H4: Vähän ristiriitainen fiilis, tavallaan se oli tosi helppoa, mutta piti oppia uusi tapa tehdä asioita low-code kielellä, kaikki ei mennytkään samalla tavalla kuin koodatessa.

H5: Oli se aika helppoa, mutta oon tehnyt ohjelmointia aikaisemmin ja tässä huomaa, että jotkut asiat helpottuu tosi paljon ja sitten taas jotkin asiat ei ole aina saatavilla low-codella.

H7: Riippuen työkalusta, canvas appit ja power automate on ollut helppoja, joskin editori voisi olla parempi. Mutta portaalien kanssa on ollut enempi haasteita, kun ohjeita ei löydy ja on vaan pitänyt itse kokeilla.

Moni mainitsi, että low-code työkalun toimintalogiikan ja ohjelmointikielen oppiminen vei hetken aikaa. Ohjelmointikokemus tai -taidot nähtiin kuitenkin hyödyksi low-code työkalujen käyttöönotossa, mutta ei kuitenkaan pakollisena, vaan niitä voi alkaa käyttämään myös ilman koodauskokemusta. Muita hyödyllisiä taitoja mainittiin esimerkiksi tiedonhakutaidot, englannin kielen taito, web kehitys kokemus sekä looginen päättelykyky.

H2: No siis looginen päättelykyky, jos syttyy semmoisista ”siirrä tulitikkua -tyyppisistä” älytehtävistä, niin sitten näiden tekeminen on aika mukavaa ja helppoakin.

H3: Onhan siitä hyötyä, jos sulla on tietokoneosaamista tai kehitystaustaa, sä pääset tavallaan nopeammin siihen pihviin ja osaat yhdistää niitä asioihin, mitä oot aiemmin oppinut. Sitten taas kokematon henkilö kyllä pääsee varmasti samaan lopputulokseen, mutta pitää investoida enemmän aikaa.

H5: No mä sanoisin, että sen voi aloittaa tosi helposti ilman ohjelmointiosaamista. Noi on kuitenkin suht simppeleitä iso osa noista pääcaseista.

H6: No ehkä myös semmoinen uteliaisuus siihen, kokeilee rohkeasti, että miten se toimii. Ja että miten saa mitäkin asioita tapahtumaan siellä, ettei pelkää sitä.

H7: Kyllä mä sanoisin, että se auttaa, jos ymmärtää koodia jotenkin. Kun se on kumminkin aika pitkälti vaan loogista päättelyä, että kaikki if:t ja tämmöiset, niin ne on niin perusjuttuja ja niitä osaa soveltaa paljon paremmin jos on koodannut. Lisäksi tiedonhaku ja sen soveltaminen omaan kontekstiin on tärkeää.

Kaikki haastateltavat kokivat saavansa low-code työkalujen käytöstä hyötyä suhteessa käytön vaativuuteen ja että käyttö on tällä hetkellä riittävän helppoa, kun kokemusta on jo kertynyt. Välillä kuitenkin saattaa mietityttää, olisiko jokin iso kokonaisuus ollut parempi toteuttaa jotain muuta teknologiaa hyödyntäen.

H1: On se aika helppoa, kun on tässä näitä 2 vuotta väsänyyt.

H2: On se lähes poikkeuksetta sen vaivan arvoista.

H4: Tää on semmoinen kyllä ja ei. Kyllähän ne on moniin tilanteisiin hyviä, mutta joskus on tullut sellainen tapaus, että on miettinyt olisiko ollut parempi tehdä koodilla.

H5: Riippuu aika paljon projektista, mutta yleensä on helppoa. Jos on tosi erikoiscase, niin silloin on vaikeampaa.

Mahdolliset haasteet vastaajat kertovat selvittävänsä googlaamalla ja etsimällä tietoa Microsoftin dokumentaatioista, blogipostauksista tai keskustelupalstoilta, myös kollegoilta sekä chat gpt:ltä kysytään apua.

5.2.3 Koettu sosiaalinen vaikutus

Haastateltavilla oli hieman eri näkemyksiä siitä, miten työyhteisössä suhtaudutaan low-code työkalujen käyttöön. Oman yksikön ja tiimin kesken ymmärrys teknologiasta on hyvä, mutta sen ulkopuolella koettiin olevan vaihtelua siinä, ymmärretäänkö ja tiedetäänkö mitä low-code työkaluilla voidaan tehdä.

H1: Joo mun mielestä koko power platform on hyvin paljon semmoinen, että tosi monet ei ymmärrä yhtään mitä me tehdään.

H3: No se on sellainen hyvin rajoittunut millä tasolla ihmiset ymmärtää sen. Monet ihmiset on tehnyt luultavasti jonkun oletuksen siitä, että mitä se on ja sitten ne pitää sitä oletusta päästä, mutta eivät täysin kuitenkaan ymmärrä mitä se on.

H4: No aika hyvin porukka mun mielestä suhtautuu low-code työkaluihin, mutta tietoa siitä ei ole kaikilla kovin paljon. Esimerkiksi jos low-code ei ole niin tuttu,

niin helposti alkaa tekemään asiaa jollain muulla tavalla, vaikka sen olisi voinut hyvin toteuttaa low-codella. Helposti ihmisille tulee ehkä sellainen mielikuva, että sillä voi tehdä vaan aika pieniä juttuja, vaikka todellisuudessa se mahdollistaa aika paljon enemmänkin kuin ihan yksinkertaisia juttuja.

H5: Oon välillä huomannut, että työkaverit, jotka ei oo tehnyt low-code juttuja miettivät, että miten ne toimii ja mitä niillä saa aikaiseksi. Voi olla myös vähän sellaisia ennakkoluuloja, että ei siellä nyt saa monimutkaisia juttuja tehtyä.

H6: Aika heikosti tietää. Ne, joita kiinnostaa, tietää mitä se on ja mitä niillä voi tehdä. Mutta kyllä se on vähän ongelma, että monilla ei ole sitä kiinnostusta ja eikä tietoa ole jaettu sellaisella riittävän eksaktilla tasolla, että ihmiset olisi sen omak-sunut organisaatiotasolla.

Vastauksissa korostuu, että kun tarkkaa tietoa ja kokemusta low-code työkaluista ei ole, helposti ajatellaan, että sillä voi toteuttaa vain hyvin pieniä ja yksinkertaisia asioita, mikä ei pidä täysin paikkaansa.

Toisaalta low-code työkaluja on hyödynnetty jonkin verran yrityksen sisäisissä töissä ja tehty esimerkiksi automaatioita työtä helpottamaan. Yleinen näkemys kuitenkin oli, että low-codea voitaisiin hyödyntää enemmän myös sisäisissä asioissa.

H2: Kyllä tässä meidänkin ympäristössä on jonkin verran käytössä noita ratkaisuja, mutta enemmänkin varmasti voisi olla. Tein itsekin joku vuosi takaperin tänne muutaman simppelein automaation.

H3: Kyllä meillä pienissä asioissa hyödynnetään low-codea itsekin, esimerkiksi markkinointitapahtumissa käytetään power appeja siihen, että skannataan QR-koodit ja kerätään ihmisiltä niiden kommentit, kun ne tulee puhumaan meidän ständille.

5.2.4 Käyttöä tukevien olosuhteiden vaikutus

Haastateltavat ovat saaneet kukin eri tavalla koulutusta low-code työkalujen käyttöönottoon, osa vähemmän ja osa enemmän. Osa haastateltavista ei ole saanut mitään koulutusta vaan opiskellut itse muun muassa Microsoftin dokumentaation ja netistä löytyvien muiden ohjeiden avulla.

H1: Alkuun meillä oli koulutuksia ja yksi parhaista oli sellainen, jossa oikeasti päästiin harjoittelemaan canvas appin tekemistä eikä vaan katsottu vierestä miten joku tapahtuu. Mutta siitä meni aikaa, ennenkö työssä pääsi tekemään mitään varsinaisia canvas appeja ja lisäksi olisi kaivannut jotain varsinaisia esimerkkejä, että missä niitä käytetään.

H2: Varsinaisesti mitään tukea ei ollut tarjolla, eli Microsoftin dokumentaation avulla menttiin silloin.

H3: Tää oli aika uutta silloin kun mä aloitin, eli paljoa tukea ei ollut saatavilla. Se oli pelkästään sitä, että itse tein ja kokeilin ja googletin. Jos vastausta ei löytynyt saatoinkin kysyä joltain koodaajalta, että miten ne tekisi tän. Sillä tavalla opittiin enemmän ja enemmän.

H4: Sain ensin koulutuksesta perustietoa ja sitten oli mentori, niin siltä ne sitten käytännössä oppi suurimman osan.

Ne, jotka olivat saaneet koulutusta, kokivat sen pääosin hyväksi. Lähes kaikki kuitenkin mainitsivat, että tekemällä oppii lopulta parhaiten. Haastateltavat myös kokivat, että saavat tällä hetkellä riittävästi tukea low-code työkalujen käytössä muun muassa kollegoilta.

H4: Parhaiten oppii joo, kun alkaa itse tekemään. Pääsee lukemaan error-rivejä.

H5: Silloin alussa tein omia canvas appeja jonkin verran ja se kyllä auttoi ihan siikana ymmärtämään sitä toimintalogiikkaa ja oli enemmän varmuutta mennä tekemään sitten asiakasprojekteissa niitä. Pitää olla motivaatiota harjoitteluun.

H6: Aika pian olin mukana asiakasprojektissa, siinä oppi eniten vaikka paljon tarvittiin kollegan tukea vielä siinä kohtaa.

Avun kysymiseen kollegoilta koettiin kuitenkin ristiriitaisia fiiliksiä. Osalle se oli helppoa, mutta moni myönsi miettivänsä avun kysymistä turhan pitkään. Vaikeus avun pyytämiseen nähtiin henkilökohtaisena ominaisuutena eikä esimerkiksi siinä, ettei olisi ketään keltä kysyä tai kollegat suhtautuisivat negatiivisesti siihen.

H1: En mä koe, että olisi kynnystä kysyä. Tiedän, että on ne muutama tyyppi, jotka osaa vastata, mutta jos niiltä kysyy niin en välttämättä odota vastausta heti, jos he ovat kiireisiä.

H2: Liian pitkään tulee itekseen mietittyä eikä homma etene. Ehkä se on joku pätemisen tarve, että myöntäisi sen oman tietämättömyyden. Se on semmoinen luonteenvika. Mutta on kiva, että meillä on niin paljon jengä kuitenkin keltä kysyä tarvittaessa.

H5: Riippuu vähän ongelmasta, onko kynnystä kysyä apua. Kyllä mä yritän aina ensin itse selvittää ja jos tuun siihen tulokseen, että en pysty, niin sitten kysyn.

H6: On vähän kynnystä kysyä, mutta se on ihan oma vika, koska kyllähän sitten kun kysyy niin saa aina nopeasti apua. Liikaa mieltii sitä, että mikä sen toisen työtilanne on, se on tän etätyön haittapuolia.

H7: Ei oikeastaan ole kynnystä kysyä. Että jos vaan tietää keltä kysyä, tai ei oikeastaan tarvitse sitäkään tietää. Kyllä joku osaa aina kertoa.

5.2.5 Aiemman teknologian käyttöön liittyvän kokemuksen vaikutus

Kaikki haastateltavat henkilöt kokivat olevansa teknologiaorientoituneita ja ottavansa uutta teknologiaa käyttöön arjessaan mielellään. Kukaan ei maininnut olevansa ns. edelläkävijä uuden teknologian käytössä, mutta kaikki sanoivat ottavansa uutta teknologiaa käyttöön sitä mukaan, kun sitä markkinoille tulee. Osa oli kiinnostuneita myös syvällisemmin ymmärtämään, miten jokin uusi teknologia toimii.

H1: Ehkä se myös arjessa näkyy, että ei jännitä, jos tulee joku uusi teknologia, vaan jos se on joku sellainen mikä kuulostaa mielenkiintoiselta, niin sitten sitä selvittää enemmänkin. Mutta en ole sellainen tyyppi, joka jatkuvasti etsii uusia juttuja, vaan enempi sillein, että mitä nyt tulee vastaan.

H2: Kyllä mä oon aina pitänyt itseäni vähän nörttinä, oon kiinnostunut kaikennäköisistä laitteista ja haluan nähdä mitä ne on syönyt. Arjessa käytän esimerkiksi lämmityksen ja valaistuksen säätöön omassa talossa, niihin on nykyään hyviä valmiita ratkaisuja.

H3: Siis mähän oon niinku semmoinen teknologianörtti, että haluan testata kaikkea uutta teknologiaa ja tavallaan niinku yrittää viedä sitä rajoilleen, että miten mä pystyn saada sen toimimaan, sillä tavalla kun mä haluan. Mut toisaalta mä voisin elää seuraavat kymmenen vuotta jossain farmilla traktorin kanssa ilman teknologiaa, sillä on myös haittapuolensa mitä se tekee ihmiselle.

H6: Koen kaikessa. Jos sen voi tehdä teknologian avulla, niin teen sen teknologian avulla.

6 POHDINTA

Tässä luvussa esitellään tärkeimmät tutkimustulokset ja tarkastellaan tutkimuksen luotettavuutta. Tämän lisäksi pohditaan tekijöitä, joilla voidaan helpottaa järjestelmän omaksumista ja joita voidaan hyödyntää esimerkiksi työntekijöiden koulutuksessa. Luvun lopussa esitetään jatkotutkimusaiheita tästä aiheesta.

6.1 Low-code työkalujen käytön omaksuminen

Tässä tutkimuksessa tarkasteltiin low-code työkalujen käytön omaksumiseen vaikuttavia tekijöitä. Uuden teknologian omaksuminen on pitkään tutkittu aihe tietojärjestelmätieteen tieteenalalla. Tässä tutkimuksessa analysoitiin omaksumista yksilön näkökulmasta organisaatiokontekstissa, jossa low-code työkalujen käyttö kuuluu haastateltavien henkilöiden työnkuvaan. Empiirisenä tutkimusmenetelmänä käytettiin teemahaastattelua, jonka kysymykset pohjautuivat tietojärjestelmätieteessä yleisesti käytettyyn UTAUT-malliin, jonka tavoitteena on ymmärtää teknologian käyttöönottoon ja omaksumiseen vaikuttavia tekijöitä. Tutkimuksen tavoitteena oli arvioida low-code työkalujen käytön omaksumiseen vaikuttavia tekijöitä. Tavoitteeseen pyrittiin vastaamalla seuraavaan tutkimuskysymykseen:

- *Mitkä tekijät vaikuttavat low-code työkalujen omaksumiseen ja onnistuneeseen käyttöönottoon?*

Vastauksena tähän kysymykseen voidaan todeta, että käyttäjän suorituskyky-odotuksilla, vaivattomuusodotuksilla ja mahdollistavilla olosuhteilla on vaikutusta teknologian omaksumiseen.

Suorituskykyodotuksilla viitataan siihen, kuinka paljon yksilö arvioi teknologian auttavan häntä päivittäisistä työtehtävistä suoriutumiseen. Suorituskykyodotuksiin liittyen haastateltavat kertoivat, että he hyötyvät low-code työkalujen käytöstä. Low-code työkalut nopeuttavat työntekoa ja sen avulla säästetään aikaa. Se myös madaltaa kynnystä tehdä pienempiä kehitystöitä asiakkaille, joihin menisi perinteiseen tapaan koodaamalla huomattavasti enemmän aikaa. Low-code työkaluja käyttämällä manuaalinen työ vähenee, kun koodia ei tarvitse kirjoittaa vaan voidaan käyttää valmiita komponentteja. Muita mainittuja hyötyjä olivat datan haun ja käsittelyn helppous sekä nopeus, sillä palvelun tarjoamien liittimien avulla päästään käsiksi moniin eri datalähteisiin. Lisäksi etuna mainittiin kokonaisuuden- ja versionhallinta, josta ei tarvitse huolehtia itse vaan työkalu tekee sen käyttäjän puolesta. Myös Beranicin ym. (2020) artikkelissa mainitaan low-code työkalujen eduksi helppokäyttöisyys ja niiden sopivuus nopea-tempoiseen sovelluskehitykseen. Venkateshin ym. (2003) mukaan suorituskykyodotuksilla on suurin vaikutus käyttöaikomukseen ja sen myötä käyttöön. Kaikki haastateltavat olivat sitä mieltä, että low-code työkalujen käytöstä on selkeää hyötyä työtehtävissä, jonka perusteella voidaan tehdä johtopäätös, että suorituskykyodotuksilla on positiivinen vaikutus teknologian omaksumiseen ja käyttöön. Kassin, Strahringerin ja Westnerin (2023) artikkelissa on päädytty samaan lopputulokseen ja todetaan käyttöä puoltavaksi tekijäksi ohjelmistokehityksen tehokkuuden paraneminen, joka voidaan listata suorituskykyodotuksiin.

Vaivattomuusodotuksilla tarkoitetaan sitä, kuinka helpoksi yksilö kokee uuden teknologian käytön. Tässä esiintyi hieman vaihtelua haastateltavien kesken, etenkin kun tarkastellaan käyttöönottoa eli sitä vaihetta, kun kokemusta työkalusta ei vielä ollut. Moni mainitsi, että alussa oli hieman haasteita ymmärtää työkalun toimintalogiikkaa ja oppia low-code ohjelmointikieli. Myöskään tietoa siitä mitä kaikkea low-code työkaluilla voi tehdä ei vielä ollut riittävästi, joten oli vaikea hahmottaa sen kaikkia hyötyjä. Muuten työkalun käyttöliittymä koettiin helpoksi ja sitä on pääosin miellyttävä käyttää. Beranicin ym. (2020) tutkimuksessa ollaan samoilla linjoilla ja mainitaan, että työkalun käyttöliittymä on helppokäyttöinen ja intuitiivinen.

Kaikilla vastaajilla oli jonkinlaista ohjelmointiosaamista tai taustaa työelämästä tai opiskeluista. Vain muutama oli kuitenkin työkseen ohjelmoinut. Low-code työkalujen käyttöönotossa nähtiin kuitenkin hyödyksi, että käyttäjällä on jonkinlaista ohjelmointiosaamista, esimerkiksi jokin ohjelmointikieli on tuttu, jotta osaa lukea ja tunnistaa koodin rakenteita. Power Platformin ohjelmointikieli PowerFx:ssä käytetään samantyyppisiä rakenteita, kuin muissakin ohjelmointikielissä, joten se, että osaa luoda ja tunnistaa niitä, auttaa alkuun pääsemisessä. Pinhon ym. (2023) tutkimuksessa korostetaankin, että low-code alustojen etu on, että ohjelmointitaidottomatkin henkilöt voivat kehittää niiden avulla sovelluksia. Tämän mahdollistamiseksi, työkalujen hyvä käytettävyys on kuitenkin välttämätöntä (Pinho ym., 2023).

Muina hyödyllisinä henkilökohtaisina taitoina nähtiin muun muassa tiedonhakutaidot sekä looginen päättelykyky, joka korostui useissa vastauksissa. Venkateshin (2003) mukaan vaivattomuusodotuksilla on suurin vaikutus käytön

alkuvaiheessa ja se vähenee, kun käyttö jatkuu pidempään. Tutkimuksen tulokset puoltavat tätä väitettä, sillä vaikka käyttöönotto olisi tuntunut haastateltavista hieman hankalalta, he kaikki kuvasivat nykytilannetta käytön suhteen helpoksi ja sujuvaksi, kun kokemusta on ehtinyt kertyä.

Sosiaalisella vaikutuksella tarkoitetaan käyttäjän sidosryhmien suhtautumista kyseisen teknologian käyttöön. Sidosryhmiä voivat olla muun muassa kollegat, ystävät ja esihenkilöt. Haastateltavat olivat yksimielisiä siitä, että työyhteisö ja esihenkilö suhtautuivat positiivisesti ja kannustavasti low-code työkalujen käyttöön. Haastateltavat eivät kuitenkaan kokeneet näiltä tahoilta painostusta käyttöön. Venkateshin (2003) mukaan sosiaalisella vaikutuksella ei ole niin suurta vaikutusta käyttöaikomukseen niissä tilanteissa, kun käyttö on vapaaehtoista, mutta sen vaikutus kasvaa, mikäli käyttö on pakollista. Tässä tutkimuksessa käyttö on niin sanotusti pakollista, sillä low-code työkalut ovat kyseisen tiimin keskeisin työkalu. Yhteenvetona voidaan todeta, että käyttäjät käyttävät low-code työkaluja mielellään ja saavat siihen kannustusta keskeisiltä sidosryhmiltä työssä.

Käyttöä tukevilla olosuhteilla viitataan organisaation tarjoamiin resursseihin, jotka tukevat uuden teknologian käyttöä. Tällaisia resursseja ovat esimerkiksi infrastruktuuri, ohjelmistojen saatavuus, koulutus ja tekninen tuki. Tämän tutkimuksen tapauksessa organisaatio tarjoaa laitteet, ohjelmistot ja tarvittavat lisenssit työntekijöiden käyttöön, joten siltä osin käyttöä tukevat olosuhteet ovat kunnossa, eikä kenelläkään haastateltavista ollut niihin liittyen negatiivista sanottavaa.

Osa haastateltavista oli saanut organisaatiolta koulutusta low-code työkalujen käyttöön, sillä he olivat tulleet taloon niin sanottujen koulutusohjelmien kautta, ja toiset olivat enempi itse opiskelleet niiden käyttöä. Koulutuksen saaminen nähtiin positiivisena tekijänä, jonka myötä oli helpompaa lähteä käyttämään työkaluja itsenäisesti. Tärkeänä tekijänä käytön alkuvaiheessa oli myös kollegan tai mentorin tuki, jolta saattoi kysyä asioita sitä myötä, kun niitä tulee eteen. Koulutus nähtiin tärkeänä, mutta vielä tärkeämmäksi koettiin alkuvaiheen omatoiminen harjoittelu ja nopea pääsy asiakastöihin, sillä tekemällä oppii parhaiten, haastateltavat totesivat. Haastateltavat kokivat, että organisaatiosta löytyy henkilöitä, jolta kysyä apua tiukan paikan tullen, mutta toiset kokivat avun pyytämisen hankalampana kuin toiset ja osa sanoi pohtivansa hankalaa asiaa liian pitkään itsekseen ennen avun kysymistä. Kokemuksella ei ollut vaikutusta avun kysymiseen, vaan se koettiin enempi henkilökohtaiseksi luonteenpiirteeksi. Vastausten perusteella organisaation tuki ja sen myötä käyttöä tukevat olosuhteet nähtiin varsin hyvänä.

Vaikuttava tekijä	Havaittu vaikutus
Suorituskykyodotukset	Positiivinen vaikutus käyttöön
Vaivattomuusodotukset	Lisääntynyt kokemus vähentää vaivattomuusodotusten vaikutusta
Sosiaaliset vaikutukset	Neutraali vaikutus käyttöön
Käyttöä tukevat olosuhteet	Positiivinen vaikutus käyttöön
Kokemus	Lisääntynyt kokemus kasvattaa suorituskykyodotuksia
Käytön vapaaehtoisuus	<i>Ei tutkittu/arvioitu</i>

6.2 Ajatuksia kouluttajille

Kouluttaessa uutta työntekijää low-code työkalujen käyttöön, on hyvä kiinnittää huomiota siihen, että työntekijälle esitellään työkalulla jo toteutettuja sovelluksia tai automaatioita. Tällöin alkuvaiheessa on helpompi hahmottaa, mihin kaikkeen työkalua voidaan käyttää ja mitä sillä voidaan toteuttaa eikä työntekijälle pääse syntymään fiilistä, että kiva työkalu, mutta en tiedä mitä tällä voidaan tehdä. Konkreettiset esimerkit lisäävät suorituskykyodotuksia ja siten motivaatiota opetella ja käyttää työkalua.

Koulutuksessa on hyvä käydä läpi teoriatasolla perustyökalut ja -prosessit, mutta antaa työntekijöille nopeasti mahdollisuus kokeilla ja käyttää itse työkalua. Hyvän oppimisprosessin tukena voidaan käyttää pienehköjä harjoitustöitä, joihin on mahdollista kysyä myös apua kouluttajalta. Näin voidaan varmistaa, että koulutettavat henkilöt voivat turvallisesti kokeilla työkalun toimintoja, ilman, että tarvitsee pelätä tekevänsä virheitä asiakasympäristössä. Kun työkalua on hieman käyttänyt, on helpompi muodostaa kuva mitä kaikkea sillä voi tehdä ja lähteä tekemään asiakastöitä.

Alkuvaiheessa on lisäksi hyvä varmistaa, että tukea on riittävästi saatavilla, sillä kaikille avun pyytäminen ei ole yhtä helppoa ja työntekijä voi etenkin näin etätöiden kulta-aikana kokea häiritsevänsä kollegaa avunpyynnöillään. Hyväksi havaittu toimintamalli on mentorointi, jossa työntekijälle on nimetty henkilö, joka auttaa häntä aina tarpeen vaatiessa. Tämä voi madaltaa kynnystä kysyä apua.

Varmistamalla työntekijöiden riittävä osaaminen saadaan low-code työkalujen käytöstä eniten hyötyä ja työntekijät kokevat käytön vaivattomaksi. Varmistamalla riittävä tuki alkuvaiheessa, voidaan ehkäistä järjestelmään kohdistuvaa kielteisyyttä tai epävarmuutta. Käyttäjät hyväksyvät kohtuulliset haasteet niin käytön kuin infrastruktuurin suhteen, mikäli heille on riittävästi tukea tarjolla ja he kokevat saavansa käytöstä hyötyä työssään.

6.3 Tutkimuksen luotettavuus

Tutkimuksen teoriaosuus käsittelee tietojärjestelmätieteen alalla yleisesti käytettyjä teorioita. Teoriaosuus toimii pohjana myös haastattelurungolle, mikä takaa sen, että kysymykset pohjautuvat luotettavina pidettyihin ja tunnettuihin teknologian omaksumistekijöihin.

Empiirisen tutkimuksen rajoituksena voidaan pitää pienehköä tutkimusjoukkoa. Tutkittavat edustivat myös suhteellisen yksipuolista ikäjakaumaa ja suuri osa heistä oli miehiä, tasapuolisempi sukupuolijakauma olisi voinut vaikuttaa tulokseen. Tutkimusjoukko oli homogeeninen myös kansallisuuden suhteen, sillä kaikki olivat suomalaisia. Edellä mainittujen rajoitusten lisäksi tutkimusjoukosta kaikki kokivat olevansa teknologiaorientoituneita henkilöitä ja käyttävänsä teknologiaa arjessaan. Tämän perusteella voidaan tulkita, että kaikilla oli suhteellisen samanlaiset valmiudet omaksua uutta teknologiaa.

Tutkittavaan aiheeseen perehdyttiin kuitenkin kattavasti kohdeyrityksen sisällä ja pystyttiin kuvailemaan ilmiötä ja tekemään havaintoja yrityksen konteksti huomioiden. Tulokset ovat vain yhden organisaation tuloksia ja yleistettäviä tuloksia varten haastattelujoukon olisi hyvä olla heterogeenisempi sekä suurempi.

6.4 Jatkotutkimusaiheita

Koska low-code teknologiat ovat suhteellisen tuore ja viime vuosina kasvanut trendi IT-alalla, niitä on tutkittu vasta suhteellisen vähän. Tämän tutkimuksen teoriaosuudessa esitelty tutkimus on tutkinut low-code työkalujen käyttöönottoa slovenialaisissa organisaatioissa. Vastaava tutkimus olisi mielenkiintoinen Suomen sisällä tehtynä. Sen avulla voitaisiin selvittää kuinka paljon organisaatiot käyttävät low-code työkaluja ja mitä syitä käytölle ja käyttämättömyydelle löytyy. Näin saataisiin selville todellisia tekijöitä, jotka vaikuttavat uuden teknologian käyttöönottoon organisaatiossa.

Toisena jatkotutkimusaiheena voisi toteuttaa tätä tutkimusta vastaavan tutkimuksen isommalle kohderyhmälle, jossa olisi eri ikäisiä henkilöitä eri organisaatioista, erilaisilla kokemuspohjilla ja teknologian taidoilla varustettuna. Näin saataisiin luotettavampaa tietoa ja paremmin selville esimerkiksi iän tai sukupuolen vaikutusta teknologian omaksumiseen.

Low-code työkaluja voitaisiin tutkia myös käyttöliittymän näkökulmasta ja haastatella eri low-code alustoja käyttäviä henkilöitä. Näin saataisiin selville

mitkä tekijät vaikuttavat käyttökokemuksen miellyttävyyteen ja sitä kautta käyttöön. Näitä tuloksia voidaan hyödyntää low-code työkalujen suunnittelussa ja jatkokehityksessä.

7 YHTEENVETO

Low-code-tekniikat ovat verrattain tuore aihe kirjallisuudessa, koska teknologian käyttö on yleistynyt voimakkaasti vasta viime vuosina. Aiemmasta kirjallisuudesta löytyi kattavasti aineistoa liittyen uuden teknologian käyttöönottoon ja omaksumiseen ja toisaalta taas low-code työkalujen tekniseen määrittelyyn sekä niiden tuomiin etuihin keskittyen. Näiden kahden yhdistelmästä eli low-code työkalujen omaksumisesta yksilötasolla ei kuitenkaan löytynyt aiempaa kirjallisuutta. Tämän tutkimuksen tavoitteena olikin lisätä tutkittua tietoa low-code työkaluista ja erityisesti niiden omaksumisesta.

Tässä tutkimuksessa tarkasteltiin low-code-työkalujen käyttöönottoa yksilöiden näkökulmasta organisaatiokontekstissa, jossa low-code-työkalujen käyttö on keskeistä heidän tehtäviensä kannalta. Kvalitatiivista tutkimusmenetelmää käyttäen suoritettiin teemahaastatteluita, jotka perustuivat laajasti käytettyyn Unified Theory of Acceptance and Use of Technology (UTAUT) -malliin tietojärjestelmätutkimuksessa teknologian omaksumisen ymmärtämiseksi. Tavoitteena oli arvioida tekijöitä, jotka vaikuttavat low-code-työkalujen käyttöönottoon ja omaksumiseen. Haastatteluita tehtiin yhteensä seitsemän kappaletta kohdeyrityksen työntekijöille.

Tulokset osoittavat, että käyttäjän suorituskykyodotuksilla, vaivattomuusodotuksilla ja mahdollistavilla olosuhteilla on vaikutusta teknologian omaksumiseen. Suorituskykyodotukset viittaavat siihen, kuinka paljon yksilöt uskovat teknologian auttavan heitä suoriutumaan päivittäisistä tehtävistään. Vastaajat raportoivat hyötyvänsä low-code-työkalujen käytöstä, sillä se nopeuttaa työtä ja säästää aikaa, madaltaen kynnystä toteuttaa pienempiä kehitystehtäviä asiakkaille. Lisäksi helppoon tiedonhakuun ja käsittelyyn, versiohallintaan ja yleiseen hallintaan liittyviä etuja korostettiin. Vaivattomuusodotukset osoittivat kuitenkin lievää vaihtelua vastaajien keskuudessa, erityisesti käytön alkuvaiheessa. Vaikka käyttöliittymää yleisesti pidettiin käyttäjäystävällisenä, alussa oli haasteita työkalun logiikan ja ohjelmointikielen ymmärtämisessä. Jossain määrin ohjelmointitaitoja nähtiin hyödyllisinä, helpottaen oppimisprosessia ja alkuvaiheen käyttöönottoa.

Lisäksi sosiaalisella vaikutuksella, joka koskee käyttäjien sidosryhmien suhtautumista teknologian käyttöönottoon, oli yksimielisesti positiivinen vaikutus vastaajien keskuudessa, työtovereiden ja esimiesten osoittaessa kannustavaa asennetta. Lisäksi mahdollistavat olosuhteet, kuten organisaation tarjoamat resurssit ja tuki koettiin hyväksi. Yhteenvetona käyttäjät näkevät low-code-työkalut hyödyllisinä ja saavat kannustusta keskeisiltä sidosryhmiltä. Varmistamalla riittävä tuki käyttöönoton alkuvaiheissa ja tarjoamalla jatkuvaa koulutusta ja mentorointia edistetään sujuvampaa käyttöönottoa.

Kässin, Strahringerin ja Westnerin (2023) artikkelissa on tutkittu low-code teknologian käyttöönottoa edistäviä ja estäviä tekijöitä organisaatiokontekstissa. Vaikka tutkimusasetelma ei olekaan aivan sama kuin tässä tutkimuksessa, voidaan yhtymäkohtia löytää. Tutkimuksessa on mainittu käyttöönottoa puoltaviksi tekijöiksi muun muassa ohjelmistokehityksen tehokkuuden parantaminen ja sovelluskehitykseen tarvittavan tiedon vähentäminen. Nämä samat tekijät voidaan havaita myös tämän tutkimuksen tuloksista, kun tarkastellaan suorituskykyodotuksia ja vaivattomuusodotuksia. Käyttäjät kokevat teknologian nopeuttavan heidän päivittäistä työtään ja mainitsevat, että sen käytön voi aloittaa hyvin vähäisillä lähtötaidoilla.

Tutkimusta voidaan pitää onnistuneena, sillä sen avulla saatiin uutta tietoa low-code työkaluista ja niiden käyttöönotosta sekä omaksumisesta. Vaikka tutkimuksella oli rajoituksia, kuten pieni otoskoko ja osallistujien homogeenisuus, se tarjoaa näkemyksiä low-code-työkalujen käyttöönotosta tietyn organisaation asetelmassa ja luo pohjaa tuleville tutkimuksille suuremmilla ja monipuolisemmilla otoksilla.

LÄHTEET

- Aaltio, I. (2022). Case-tutkimus metodisena lähestymistapana. Luettu 5.12.2022, <https://metodix.fi/2014/05/19/aaltio-marjosola-casetutkimus/>
- Alamin, M. A. A., Uddin, G., Malakar, S., et al. (2023). Developer discussion topics on the adoption and barriers of low code software development platforms. *Empirical Software Engineering*, 28(4). <https://doi.org/10.1007/s10664-022-10244-0>
- Ali, S., Green, P., & Robb, A. (2015). Information technology investment governance: What is it and does it matter?. *International Journal of Accounting Information Systems*, 18, 1-25.
- Beranic, T., Rek, P., & Heričko, M. (2020). Adoption and usability of low-code/no-code development tools. Varazdin: Faculty of Organization and Informatics Varazdin. Retrieved from <https://www.proquest.com/conference-papers-proceedings/adoption-usability-low-code-no-development-tools/docview/2531366275/se-2>
- Bock, A. C., & Frank, U. (2021). Low-code platform. *Business & Information Systems Engineering*, 63, 733-740.
- Braun, V., & Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4), 589-597.
- Bucaioni, A., Cicchetti, A., & Ciccozzi, F. (2022). Modelling in low-code development: A multi-vocal systematic review. *Software and Systems Modeling*. <https://doi.org/10.1007/s10270-021-00964-0>
- Daniel, G., Cabot, J., Deruelle, L., & Derras, M. (2020). Xatkit: A multimodal low-code chatbot development framework. *IEEE Access*, 8, 15332-15346.
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319-340. <https://doi.org/10.2307/249008>

- Dillon, A., & Morris, M. G. (1996). *User acceptance of new information technology: theories and models*. Medford, NJ: Information Today.
- Eriksson, P., & Koistinen, K. (2014). *Monenlainen tapaustutkimus*. Kuluttajatutkimuskeskus.
- Hirsjärvi, S., & Hurme, H. (2000). *Tutkimushaastattelu: Teemahaastattelun teoria ja käytäntö*. Helsinki: Yliopistopaino.
- King, W., & He, J. (2006). A Meta-analysis of the Technology Acceptance Model. *Information & Management*, 43(6), 740-755.
- Käss, S., Strahringer, S., & Westner, M. (2023). Practitioners' Perceptions on the Adoption of Low Code Development Platforms. *IEEE Access*, 11, 29009-29034. <https://doi.org/10.1109/ACCESS.2023.3258539>
- Legris, P., Ingham, J., & Collerette, P. (2003). Why do people use information technology? A critical review of the technology acceptance model. *Information & Management*, 40(3), 191-204.
- Marangunić, N., & Granić, A. (2015). Technology Acceptance Model: A Literature Review from 1986 to 2013. *Universal Access in the Information Society*, 14(1), 81-95.
- Microsoft. (2022). Power Platform. Effective December 03, 2022. <https://powerplatform.microsoft.com/en-us/>
- Myers, M. D. (2019). *Qualitative research in business and management*. Qualitative research in business and management, 1-364.
- Pinho, D., Aguiar, A., & Amaral, V. (2023). What about the usability in low-code platforms? A systematic literature review. *Journal of Computer Languages*, 74. <https://doi.org/10.1016/j.cola.2022.101185>
- Prinz, N., Rentrop, C., & Huber, M. (2021). Low-Code Development Platforms- A Literature Review. In *AMCIS*.

- Rahman, F. B. A., Hanafiah, M. H. M., Zahari, M. S. M., & Jipiu, L. B. (2021). Systematic Literature Review on The Evolution of Technology Acceptance and Usage Model used in Consumer Behavioural Study. *International Journal of Academic Research in Business and Social Sciences*, 11(13), 272-298.
- 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.
<https://doi.org/10.1007/s10270-021-00970-2>
- 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).
- Sanchis, R., García-Perales, Ó., Fraile, F., & Poler, R. (2019). Low-Code as Enabler of Digital Transformation in Manufacturing Industry. *Applied Sciences*, 10(1). <https://doi.org/10.3390/app10010012>
- Tisi, M., Mottu, J. M., Kolovos, D. S., De Lara, J., Guerra, E. M., Di Ruscio, D., ... & Wimmer, M. (2019, July). Lowcomote: Training the next generation of experts in scalable low-code engineering platforms. In STAF 2019 Co-Located Events Joint Proceedings: 1st Junior Researcher Community Event, 2nd International Workshop on Model-Driven Engineering for Design-Runtime Interaction in Complex Systems, and 1st Research Project Showcase Workshop co-located with Software Technologies: Applications and Foundations (STAF 2019).
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3), 425-478. <https://doi.org/10.2307/30036540>
- Venkatesh, V., Thong, J. Y. L., & Xu, X. (2012). Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology. *MIS Quarterly*, 36(1), 157-178.
<https://doi.org/10.2307/41410412>

Voronkov, I. A., & Saradgishvili, S. E. (2021). Power Fx: Low-code language for collaboration tools. Труды Института системного программирования, 101-108.

Why low-code development matters right now. (2022). Retrieved from <https://powerapps.microsoft.com/en-us/what-is-low-code/>

Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. IFAC-PapersOnLine, 52(10), 376-381.

Yin, R. K. (1994). Case study research: Design and methods (2nd ed.). Newbury park, CA: SAGE publications.

Haastattelukysymykset:

Osa 1. Taustatiedot

- Ikä ja sukupuoli
- Kuvaile nykyistä työtehtävääsi, millaisia low-code työkaluja käytät työssäsi?
- Miten pitkään olet työskennellyt näiden parissa?
- Minkälainen teknologinen osaaminen/työtausta sinulla oli ennen näiden työkalujen käyttöönottoa, esimerkiksi ohjelmointitaidot/kokemus? Opiskelu?
- Onko kokemusta esimerkiksi jonkin toisen palvelutarjoajan low-code työkaluista?
- Koetko olevasi teknologiaorientoitunut henkilö? Millaisissa tilanteissa käytät/hyödynnät teknologiaa esim. arjessa?

Osa 2. Suorituskykyodotukset

- Miten koet, että low-code työkalut helpottavat työn tekoa (vs. jos sama tehtäisiin jotain muuta teknologiaa käyttäen)?

Osa 3. Vaivannäköodotukset

- Millaisia haasteita olet kokenut low-code työkalujen käyttöönotossa vai oliko se helppoa?
- Mainitse tekijöitä/taitoja, jotka vaikuttivat positiivisesti käyttöönoton onnistumiseen? Koetko, että ohjelmointiosaaminen tai muu tekijä helpottaa käyttöönottoa?
- Onko low-code työkalujen käyttäminen tällä hetkellä riittävän helppoa, mikä tekee siitä helppoa/vaikeaa? Miten selvität eteen tulevat haasteet?
- Koetko saavasi low-code työkalujen käytöstä hyötyä suhteessa käytön vaativuuteen?

Osa 4. Sosiaalinen vaikutus

- Miten työyhteisössä suhtaudutaan low-code työkalujen käyttöön?
- Tukeeko esihenkilösi työkalujen käyttöä?

Osa 5. Mahdollistavat olosuhteet

- Käyttöönotto: Millaista tukea sait ja koit tarvitsevasi käyttöönotossa?
- Koetko saavasi tarpeeksi apua low-code työkalujen käytössä?
- Onko sinulla kynnystä kysyä apua? Miksi?