LIIKETOIMINTAPROSESSIEN AUTOMATISOINTI LOW-CODE ALUSTOJEN AVULLA: MAHDOLLISUUDET JA HAASTEET

Lappeenrannan–Lahden teknillinen yliopisto LUT

Tietotekniikan diplomityö

2024

Anni-Sofia Niekkamaa

Tarkastajat: Apulaisprofessori Jussi Kasurinen

Professori Sami Hyrynsalmi

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tietotekniikka

Anni-Sofia Niekkamaa

Liiketoimintaprosessien automatisointi low-code alustojen avulla: Mahdollisuudet ja haasteet

Tietotekniikan diplomityö

2024

75 sivua, 5 kuvaa, 5 taulukkoa ja 1 liite

Tarkastajat: Apulaisprofessori J.K. ja Professori S.H.

Avainsanat: Low-code alustat, Liiketoimintaprosessien automatisointi, Prosessimallintaminen, Kansalaiskehitys

Low-code alustojen käyttö mahdollistaa prosessien automatisoinnin ilman syvällistä ohjelmointiosaamista, mikä tekee niistä hyvän vaihtoehdon useille organisaatioille. Tämän työn tavoitteena on tutkia miten low-code alustat voivat tehostaa yritysten prosessien automatisointia ja mitä haasteita mahdollisesti niiden käyttöön liittyy. Tarkoituksena on tutkia low-code alustojen roolia liiketoimintaprosessien automatisoinnissa hyötyjen ja haasteiden näkökulmasta, keskittyen miten alustat voivat nopeuttaa ja parantaa kehitystä sekä mahdollistaa myös kansalaiskehittäjien osallistumisen automatisointiin.

Työ jakautuu kahteen eri osaan: kirjallisuuskatsaukseen sekä empiiriseen tutkimukseen. Kirjallisuuskatsauksessa tarkastellaan low-code alustoja, liiketoimintaprosessien automatisointia sekä näiden kahden linkittymistä yhteen. Empiirinen tutkimus toteutetaan puolistrukturoituina asiantuntijahaastatteluina kohdeyritykselle, jossa low-code alustoja käytetään prosessien automatisoinnissa. Tutkimuksessa analysoidaan keskeisiä teemoja, havaintoja ja trendejä low-code alustojen käytöstä, hyödyistä ja haasteista kohdeyrityksen asiantuntijoiden näkökulmasta.

Työn tulokset osoittavat, että low-code alustat tarjoavat etuja prosessien automatisoinnissa, kuten nopeamman sovelluskehityksen, joustavuutta ja mahdollisuuden osallistaa eritasoisia henkilöitä. Haasteita puolestaan ovat monimutkaisen prosessien mallintaminen ja alustojen tekninen rajallisuus. Low-code alustat voivat olla tehokas työkalu liiketoimintaprosessien automatisoinnissa, mutta niiden käyttöönotto vaatii huolellista suunnittelua ja resursseja.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

School of Engineering Science

Software Engineering

Anni-Sofia Niekkamaa

Automating Business Processes with Low-Code Platforms: Opportunities and Challenges

Master's thesis

2024

75 pages, 5 figures, 5 tables and 1 appendix

Examiners: Associate Professor J.K. and Professor S.H.

Keywords: Low-code platforms, Business process automation, Process modeling, Citizen development

The use of low-code platforms enables automation of processes without deep programming knowledge, which makes them a good option for many organizations. The goal of this thesis is to study how low-code platform can enhance the automation of companies processes and what challenges may occur with their use. The purpose is to study the role of low-code platforms in the automation of business processes from perspective of benefits and challenges, focusing on how platforms can speed up and improve development and enable citizen developers to participate in automation.

The work is divided into two different parts: literature review and empirical research. Literature review explores low-code platforms, automation of business processes and the link between the two. Empirical research is conducted through semi-structure expert interviews for the target company that uses low-code platforms in process automation. The research analyzes key themes, findings and trends regarding the use, benefits and challenges of low-code platforms from the perspective of target company's experts.

The results of the work show that low-code platforms offer advantages in process automation, such as faster application development, flexibility and possibility to involve people with varying skill levels. The challenges, on the other hand, are modeling of complex processes and the technical limitations of the platform. Low-code platforms can be an effective tool for automating business processes, but their implementation requires careful planning and resources.

KIITOKSET

Haluaisin kiittää Jussi Kasurista diplomityöni ohjaamisesta sekä hyvistä kommenteista ja neuvoista tämän kirjoitusprosessin aikana. Lisäksi kiitokset kaikille asiantuntijahaastatteluihin osallistujille, jotka annoitte arvokasta aikaanne tälle työlle ja mahdollistitte sen syntymisen.

Kiitokset myös perheelleni ja ystävilleni tuesta ja tsempeistä tätä projektia kohtaan. Te uskoitte tämän työn ja tutkinnon valmistumiseen välillä enemmän kuin minä itse. Nyt voin vihdoin yli seitsemän vuoden rupeaman jälkeen mennä sunnuntaipäivälliselle ilman kyselyjä tutkinnon valmistumisarviosta. Iso kiitos myös Laurille vertaistuesta ja tärkeistä tsempeistä, tämä työ ei olisi koskaan valmistunut ilman sua.

LYHENNELUETTELO

Lyhenteet

AutoML	automatisoitu koneoppiminen (Automated Machine Learning)
API	palvelurajapinta (Application Programming Interface)
BPA	liiketoimintaprosessien automatisointi (Business Process Automation)
BPM	liiketoimintaprosessien hallinta (Business Process Management)
BPNM	liiketoimintaprosessin mallinnusjärjestelmä (Business process modelling and notation)
EUD	loppukäyttäjäkehitys (End-User Development)
IDE	ohjelmointiympäristö (Integrated Development Environment)
LCD	low-code kehitys (Low Code Development)
LCDP	low-code sovelluskehitysalusta (Low-Code Development Platform)
LCP	low-code alusta (Low-Code Platform)
MDSE	mallipohjainen ohjelmistosuunnittelu (Model-driven Software Engineering)
NLP	luonnollisen kielen ohjelmointi (Natural-Language Programming)
RPA	robottiprosessiautomaatio

Sisällysluettelo

Tiivistelmä

Abstract

Kiitokset

Lyhenneluettelo

1 Johdanto 11
1.1 Tavoitteet ja rajaus 12
1.2 Tutkimuksen toteutus 12
1.3 Diplomityön rakenne 13
2 Teoreettinen viitekehys 15
2.1 Low-code alustojen määritelmä 15
2.1.1 Low-code alustojen pääpiirteet 16
2.1.2 Low-code alustojen kehitys, arkkitehtuuri ja ominaisuudet 17
2.2 Low-code alustojen taustat ja historia 19
2.2.1 Low-code alustojen käyttöönotto 20
2.2.2 Alan sovellukset ja trendit 23
2.3 Liiketoimintaprosessien hallinta ja sen merkitys liiketoiminnalle 24
2.3.1 Liiketoimintaprosessien automaation tärkeys ja hyödyt 25
2.3.2 Liiketoimintaprosessiautomaation mahdollisuudet ja haasteet 26
2.4 Low-code alustat liiketoimintaprosessien automaatiossa 27
2.4.1 Low-code alustojen rooli nykyaikaisessa liiketoiminta-automaatiossa 29
2.4.2 Prosessien mallintaminen low-code alustoilla 30
2.5 Aiempi kirjallisuus low-code alustojen käytöstä liiketoimintaprosessiautomaatiossa 31
2.5.1 Low-code alustojen hyödyt liiketoimintaprosessiautomaatiossa 32
2.5.2 Low-code alustojen haasteet liiketoimintaprosessiautomaatiossa 34
3 Tutkimusmenetelmät 38
3.1 Tutkimuslähestymistapa 38
3.2 Aineiston keruu 40
3.3 Aineiston analyysi ja luotettavuus 41

4 Tulokset ja analyysi 43
4.1 Kohdeyrityksen kuvaus 43
4.2 Analyysi ja havainnot low-code alustojen käytöstä ja haasteista 43
4.2.1 Monimutkaiset prosessiautomaatiot 44
4.2.2 Tuotteen soveltuvuus ja puutteet 46
4.2.3 Suorituskyky ja skaalautuvuus 48
4.2.4 Kansalaiskehityksen haasteet ja hyödyt 49
4.2.5 Turvallisuus ja tiedonsiirto 52
4.3 Haastatteluista saadut tulokset 53
4.3.1 Hyödyt 53
4.3.2 Haasteet 56
4.3.3 Parhaat käytännöt 58
5 Tulosten soveltaminen 62
5.1 Tulosten tarkempi tarkastelu teoreettisen viitekehyksen valossa 62
5.2 Tutkimuksen aineiston rajallisuus 65
6 Johtopäätökset 67
Lähteet 72
Liitteet

Liite 1. Asiantuntijahaastattelujen kysymykset

Kuvaluettelo

Kuva 1 Input -output kaavio

Kuva 2 Low-code kehitysalustojen yleinen arkkitehtuuri

Kuva 3 Low-code kehitysalustojen keskeiset komponentit ja tukiprosessit

Kuva 4 Automaation ehdokkaat

Kuva 5 Low-code sovellusalustojen osa-alueita

Taulukkoluettelo

Taulukko 1 Low-code alustojen käyttöönoton haasteita

Taulukko 2 Liiketoimintaprosessien automaation hyötyjä

Taulukko 3 Low-code alustojen hyötyjä liiketoimintaprosessiautomaatiossa

Taulukko 4 Low-code alustojen haasteita liiketoimintaprosessiautomaatiossa

Taulukko 5 Kerätyn aineiston teemat ja alateemat

1 Johdanto

Liiketoimintaprosessien automatisointi (BPA) on kasvava trendi, jonka merkitys korostuu yritysten pyrkiessä parantaa operatiivista tehokkuuttaan sekä vähentämään manuaalisten prosessien aiheuttamia virheitä. BPA markkinat ovat tällä hetkellä lähes 15 miljardin dollarin kokoiset ja niiden odotetaan yli kaksinkertaistuvan vuoteen 2032 mennessä. Tätä kasvua vauhdittavat digitaalinen transformaatio ja IT-automaatioratkaisujen laajempi käyttöönotto yrityksissä. Myös pilvipohjaisten ja tekoälyä hyödyntävien BPM-ratkaisujen suosio lisääntyy, minkä odotetaan parantavat operatiivista tehokkuutta sekä vähentävän inhimillisiä virheitä. (Fortune Business Insights, 2024)

Yritykset pyrkivät parantamaan liiketoimintaprosessien hallintaa (BPM) määrittelemällä, mallintamalla ja optimoimalla toimintansa. Organisaatiot, jotka haluavat olla ketterämpiä ja reagoida nopeammin markkinoiden muutoksiin ovat alkaneet hyödyntää low-code teknologioita mahdollistaen prosessien mallintamisen myös ei-teknisille käyttäjille eli kansalaiskehittäjille. (Tetratech, 2022) Low-code alustojen konsepti on ollut olemassa jo vuosikymmeniä. Low-code alustat kehitettiin mahdollistamaan sovellusten rakentaminen ja kehittäminen myös niille henkilöille, joilla ei ole yhtä syvällistä teknistä osaamista kuin ammattiohjelmoijilla. Alustojen avulla voidaan silti hyödyntää kehittyneitä teknologioita, kuten tietokantoja, verkkosivujen kehitystä ja koneoppimista. (Lethbridge, 2021) Low-code kehitysalustat tarjoavat visuaalisia sovelluskehitystyökaluja sekä valmiita komponentteja prosessien automaatioon, mikä minimoi manuaalisen ohjelmoinnin tarpeen. Tämä mahdollistaa eritasoisten käyttäjien osallistamisen ohjelmistokehitykseen, parantaen liiketoiminnan ja kehitystiimien välistä yhteistyötä sekä nopeuttaen kehitysprosessia. (Rokis & Kirikova, 2023)

Kansalaiskehittäjien merkitys on kasvanut automaatioratkaisuiden määrän kasvun myötä. Kun yritykset jatkavat automaation omaksumista, kasvaa tarve kansalaiskehittäjille, jotka voivat rakentaa automaatioratkaisuja päivittäisiin liiketoimintatehtäviin. Kansalaiskehittäjät ovat oman alansa asiantuntijoita, mutta heiltä puuttuu usein tarvittavia teknisiä taitoja automaatioiden luomiseen ja ylläpitoon. (Desmond et al., 2022)

1.1 Tavoitteet ja rajaus

Tutkimuksen tavoitteena on tutkia millä tavoilla low-code alustat voivat tehostaa liiketoimintaprosessien automatisointia ja millaisia mahdollisia haasteita niiden käyttöönotossa voi ilmetä. Tutkimuksessa keskitytään erityisesti siihen, miten low-code alustat voivat parantaa prosessien tehokkuutta ja joustavuutta yrityksissä. Näiden tavoitteiden pohjalta johdettiin tutkimuksen päätutkimuskysymys, joka on:

Mitkä ovat hyödyt ja haasteet low-code alustojen käytössä liiketoimintaprosessien automatisoinnissa?

Lisäksi työn osatutkimuskysymyksiä ovat:

- *Mitkä ovat suurimmat haasteet ja tekniset rajoitukset, joita organisaatiot kohtaavat pyrkiessään hyödyntämään low-code alustoja monimutkaisten liiketoimintaprosessien automatisoinnissa?*
- *Mitä parhaita käytäntöjä ja oppeja voidaan saada organisaatioilta, jotka ovat onnistuneesti hyödyntäneet low-code alustoja prosessien automatisoinnissa?*

Tutkimus keskittyy low-code alustojen käyttöön liiketoimintaprosessien automatisoinnissa eikä käsittele laajemmin muita ohjelmointialustoja -tai kieliä. Tarkoituksena on luoda ymmärrys, miten aihealuetta on aikaisemmin kirjallisuudessa tutkittu ja miten kohdeyrityksessä samoja haasteita ja hyötyjä on ratkaistu sekä koettu. Tutkimuksen empiirinen osuus perustuu kohdeyrityksen asiantuntijahaastatteluihin eikä laajempia kvantitatiivisia kyselytutkimuksia sisällytetä tutkimukseen ajan ja resurssien rajallisuuden vuoksi. Lisäksi tämä tutkimus rajoittuu kohdeyrityksen sisäisten sidosryhmien näkemyksiin sekä ekosysteemiin. Tutkimus keskittyy liiketoiminnallisiin näkökulmiin ja prosessien automatisointiin eikä sisällä teknisiä syväanalyysejä yksittäisten low-code alustojen ohjelmointikielistä tai yksityiskohdista.

1.2 Tutkimuksen toteutus

Tämä diplomityö on kvalitatiivinen tapaustutkimus, jonka tavoitteena on tutkia, selittää sekä kuvata ilmiöitä tutkimuskysymysten avulla. Tutkimus toteutettiin kirjallisuuskatsauksena sekä empiirisenä tutkimuksena. Tutkimus aloitettiin perehtymällä aihealueeseen etsimällä aiempia tutkimuksia sekä tieteellisiä artikkeleita aihealueesta. Aiheeseen perehtymisen

jälkeen tutkimukselle laadittiin tutkimussuunnitelma, jossa työn aikatauluksi päätettiin kuusi kuukautta.

Kvalitatiivinen aineisto kerättiin asiantuntijahaastatteluilla. Asiantuntijahaastattelut toteutettiin puolistrukturoiduilla teemahaastatteluilla, joiden teemat sekä kysymykset perustuivat teoreettisen viitekehyksen tuloksiin. Haastattelut tehtiin kohdeyrityksen työntekijöiden kanssa syksyllä 2024 ja niiden tavoitteena oli saada syvällinen ymmärrys siitä, miten low-code alustoja on käytetty, millaisia hyötyjä ja haasteita on kohdattu sekä miten organisaatio on ratkaissut näihin liittyviä kysymyksiä ja ongelmia.

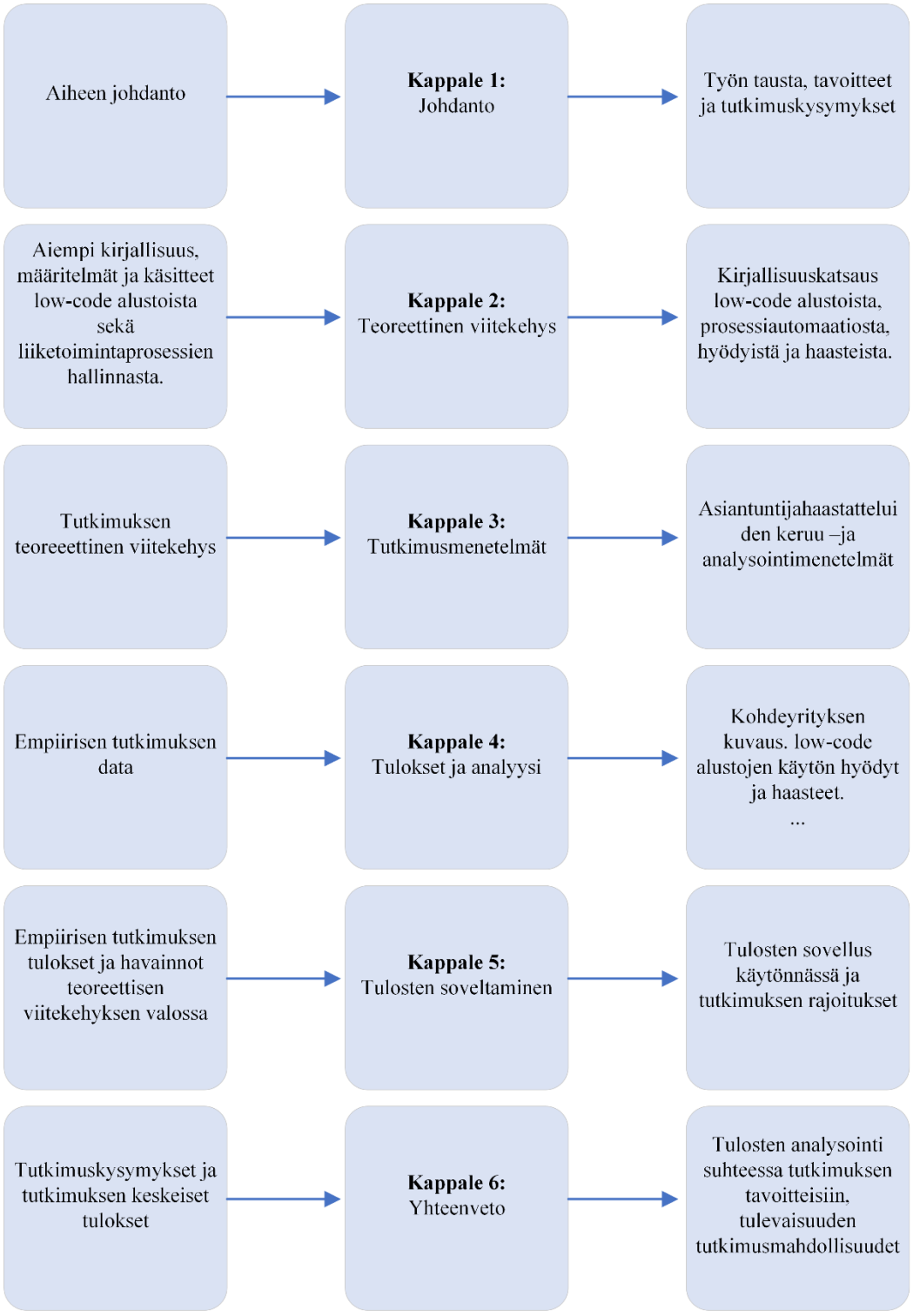
1.3 Diplomityön rakenne

Diplomityö koostuu viidestä kokonaisuudesta, joita ovat johdanto, kirjallisuuskatsaus, empiirinen tutkimus, tulokset ja havainnot sekä johtopäätökset. Raportin johdannossa esitellään tutkimuksen rakenne sekä työn tausta. Kokonaisuus myös käsittää tutkimuksen tavoitteet, rajoitukset, tutkimuskysymykset sekä tutkimuslähestymistavan.

Työn kirjallisuuskatsaus on yksi kokonaisuus, joka on jaettu viiteen eri osakokonaisuusteen. Ensimmäinen kokonaisuus käsittelee low-code alustojen määritelmää sekä tärkeimpiä pääpiirteitä. Toinen osakokonaisuus keskittyy low-code alustojen taustoihin sekä historiaan. Kolmannessa osassa perehdytään liiketoimintaprosesseihin ja niiden automaatioon. Neljäs osa käsittelee low-code alustoja liiketoimintaprosessien automaatiossa ja viides kokonaisuus aiempaa kirjallisuutta low-code alustojen käytöstä liiketoimintaprosessien automaatiossa.

Empiirisen tutkimuksen osiossa tarkastellaan laadullisia eli kvalitatiivisia tutkimusmenetelmiä, datan keräämistä ja aineiston analyysia sekä luotettavuutta. Osiossa esitellään myös kohdeyritys, jolle asiantuntijahaastattelut tehtiin. Lisäksi osiossa analysoidaan tarkemmin empiirisen tutkimuksen tuloksia teemoittelun menetelmin.

Tulokset ja havainnot osio yhdistää kirjallisuuskatsauksen ja empiirisen tutkimuksen tulokset low-code alustojen haasteista ja hyödyistä liiketoimintaprosessien automatisoinnissa. Tuloksia tarkastellaan tarkemmin sekä esitetään tutkimuksen aineiston rajallisuudet. Diplomityön viimeisenä osiona on johtopäätökset, jossa vastataan asetettuihin tutkimuskysymyksiin sekä annetaan suosituksia jatkotutkimusmahdollisuuksille. Työn kokonaisrakenne on kuvattu Kuva 1:ssä.



Kuva 1 Input -output kaavio

2 Teoreettinen viitekehys

Tässä pääluvussa kuvataan kirjallisuudessa esitettyjä low-code alustojen ominaisuuksia sekä liiketoimintaprosessien automaatiota. Luvussa myös kuvataan low-code prosessien taustaa sekä miten liiketoimintaprosessien automaatio ja low-code alustat linkittyvät toisiinsa. Tämä lisäksi käydään läpi aikaisempaa tutkimusta low-code alustojen käytöstä prosessiautomaatiossa.

2.1 Low-code alustojen määritelmä

Low-code sovellusalusta (LCAP) on sovellusalusta, joka tukee nopeaa sovelluskehitystä, yhden vaiheen käyttöönottoa ja käyttää kuvailevia korkean tason ohjelmointikieliä, jotka ovat esimerkiksi mallipohjaisia tai metadataan perustuvia. Low-code sovellusalustat tukevat käyttöliittymien, liiketoimintalogiikan ja datapalveluiden kehitystä ja niiden on tutkittu parantavan tuottavuutta verrattuna perinteisiin sovellusalustoihin. (Vincent et al., 2020) Alustat hyödyntävät mallipohjaisen suunnittelun periaatteita, joissa malleja käytetään ensisijaisina elementteinä monimutkaisten järjestelmien rakentamisessa. Low-code alustat ovat usein pilvipohjaisia ja tämän vuoksi käyttäjät odottavat niiden olevan reaktiivisia ja suorittavan monimutkaisia toimintoja lyhyessä ajassa. (Horváth et al., 2020)

Olemassa olevat low-code alustat voidaan jakaa kolmeen eri tyyppiin niiden soveltamisalan mukaan, jotka ovat:

- Liiketoimintaprosessien automatisointi (BPM-järjestelmien kehittäminen)
- Web-sovellusten kehittäminen
- Tietokannan hallinta

Liiketoimintaprosessien automatisoinnin alustat mahdollistavat liiketoimintaprosessien kuvaamisen sekä niihin liittyvien tietokantojen ja käyttöliittymien luomisen. Web-sovellusten kehittämisen alustoilla voidaan rakentaa verkkosivuja -tai sovelluksia käyttämällä valmiita komponentteja, jota edustavat erilaisia osia verkkosivusta. Tietokannan hallinnan alustoilla luodaan tietokantoja määrittelemällä perustiedot, rakentamalla käyttöliittymä ja liittämällä siihen datankäsittelyn logiikka. (Kovalev et al., 2022)

2.1.1 Low-code alustojen pääpiirteet

Low-code alustat (LCP) tarjoavat usein useita keskeisiä ominaisuuksia, jotka tekevät niistä tehokkaita työkaluja sovellusten kehittämiseen. Pääpiirteitä muun muassa ovat:

- Vedä ja pudota -käyttöliittymä: LCP:t tarjoavat yksinkertaisen vedä ja pudota -toiminnon, joka nopeuttaa sovellusten visuaalista luomista (Thaker, 2022).
- Liikkuvuus: LCP:t on suunniteltu toimimaan laajasti käytetyillä mobiililaitteilla, kuten älypuhelimilla ja tableteilla (Thaker, 2022).
- Skaalattavuus: LCP:t optimoivat kehittäjän tuottavuuden ja lisäävät ketteryyttä, mahdollistaen yrityksille uusien sovellusten nopeamman julkaisun sekä nykyisten sovellusten nopeamman muokkaamisen käyttäjän palautteen tai markkinamuutosten perusteella (Thaker, 2022).
- Ristiin toimiva saatavuus: LCP:t mahdollistavat sovellusten kehittämisen, jotka toimivat useilla eri alustoilla. Tämä tarkoittaa, että uusien sovellusten luominen ja käyttöönotto on mahdollista nopeasti, jotta voidaan vastata asiakkaiden tarpeisiin ja vaatimuksiin (Thaker, 2022).
- Ulkoisten tietolähteiden hyödyntäminen: LCP:t voivat hyödyntää ulkoisia tietolähteitä erilaisten palvelurajapintojen (API) avulla. Tämä mahdollistaa datan varastoimisen joko sisäiseen tietokantajärjestelmään tai olemassa oleviin ulkoisiin järjestelmiin (Bock & Frank, 2021).
- Yksinkertaiset ilmaisukielet ja vuoropuhelupohjaiset suunnittelutavat: LCP:t tarjoavat yksinkertaisia ilmaisukieliä päätössääntöjen määrittämiseen sekä vuoropuhelupohjaisia työkaluja eri prosessien ja työnkulkujen suunnitteluun (Bock & Frank, 2021).
- Roolien ja käyttöoikeuksien määrittely: LCP-ratkaisuissa on yleensä komponentti roolien ja käyttöoikeuksien määrittelyyn (Bock & Frank, 2021).
- Käsitteellinen mallinnuskieli: LCP:t käyttävät usein käsitteellistä mallinnuskieltä, kuten BPMN:ää, joka mahdollistaa monimutkaisten prosessien ja järjestelmien visuaalisen esittämisen ja hallinnan (Bock & Frank, 2021)

Nämä ominaisuudet tekevät LCP:stä joustavan ja tehokkaan työkalun sovellusten kehittämiseen. Ne tarjoavat kehittäjille mahdollisuuden luoda sovelluksia nopeasti ja tehokkaasti, samalla vastaten käyttäjien tarpeisiin ja vaatimuksiin. On hyvä huomioida, että kaikki mainitut ominaisuudet eivät välttämättä toteudu kaikilla low-code alustoilla. Alustojen ominaisuudet voivat vaihdella riippuen niiden suunnittelusta, tarkoituksesta ja käyttöympäristöstä.

2.1.2 Low-code alustojen kehitys, arkkitehtuuri ja ominaisuudet

Low-code kehitys (LCD) on lähestymistapa, joka on tullut yhä suositummaksi viime vuosina. Low-code kehityksen työkalut mahdollistavat ohjelmistojen kehityksen kirjoittamalla vähemmän koodia verrattuna perinteisiin ohjelmistokehityksen menetelmiin. LCD-työkalut auttavat kehittäjiä keskittymään enemmän ohjelman logiikkaan muiden teknologioiden sijaan. (Pinho et al., 2023)

Low-code sovelluskehitysalustan (LCDP) arkkitehtuuri voidaan jakaa karkeasti neljään eri osa-alueeseen (Liu et al., 2024):

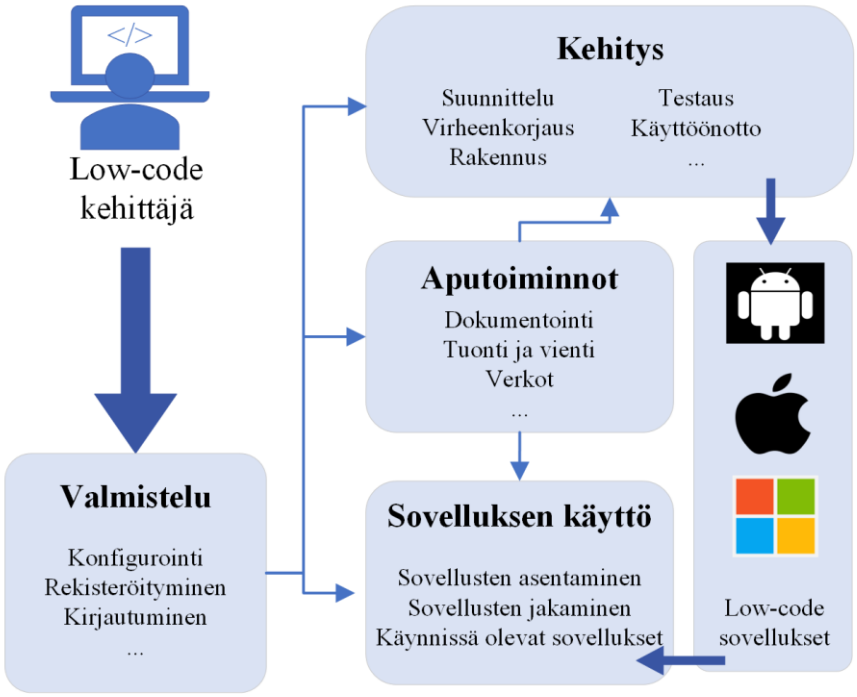
- *Valmistelu*: Valmistelu sisältää useita toimintoja, jotka liittyvät LCDP:iden käyttöönottoon. Koska LCDP:t toimivat usein pilviympäristössä, ne vaativat yleensä käyttäjiltä rekisteröitymisen ennen käyttöä. Lisäksi valmisteluun voi kuulua muita toimintoja kuten ympäristön määrittely, tarvittavien työkalujen asennus ja integraatioiden konfigurointi muihin järjestelmiin.

- *Kehitys:* Kehitys on LCDP:n ensisijainen toiminto. Kehityksen keskeinen vaihe on määritellä kehitettävät sovellukset mukaan lukien tietomallinnus, käyttöliittymän suunnittelu, liiketoimintalogiikan säännöt jne. Lisäksi jotkin LCDP:t voivat tarjota muita tukitoimintoja kehitetyille sovelluksille kuten virheenkorjauksen, testauksen ja kehitystyön tuen.

- *Sovelluksen käyttö:* Low-code alustat toimivat usein sekä kehitys- että käyttöympäristöinä. Ne tarjoavat tarvittavat palvelut, kuten tietokannat ja ulkoiset palvelurajapinnat (API), jotta kehitetyt sovellukset voidaan ottaa suoraan käyttöön alustalla. Tämä mahdollistaa nopean siirtymisen kehityksestä tuotantoon ja tekee sovellusten hallinnasta sujuvampaa.

- *Aputoiminnot:* Aputoiminnot tukevat LCDP:n käyttöä. Ne voivat sisältää dokumentoinnin, joka auttaa ymmärtämään ja seuraamaan sovelluksen kehitystä sekä tuonti- ja vientityökaluja, jotka mahdollistavat tiedon siirtämisen alustojen välillä. Lisäksi aputoiminnot voivat sisältää ulkoisten palveluiden ja verkkojen konfigurointia, joka mahdollistaa sovellusten integroinnin muihin järjestelmiin.

Kuva 2 esittää nämä neljä osa-aluetta ja niiden keskinäiset suhteet. Se havainnollistaa, kuinka eri osa-alueet liittyvät toisiinsa ja muodostavat kokonaisvaltaisen low-code sovelluskehitysalustan.

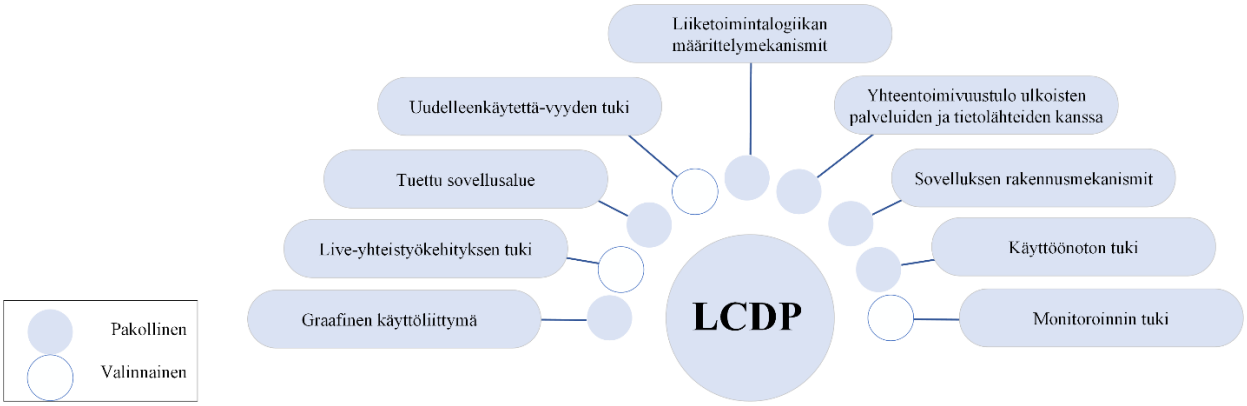


Kuva 2 Low-code kehitysalustojen yleinen arkkitehtuuri (mukaillen Liu, D. et al. 2024)

Kuva 3 esittää ylätasolla low-code sovellusalustojen keskeisiä ominaisuuksia. Ominaisuudet ovat Kuva 3:ssa jaettu kahteen eri tasoon: Pakolliset ominaisuudet sekä Valinnaiset ominaisuudet. LCDP:t erottuvat toisistaan tarjoamalla käyttäjille edistyneitä graafisia käyttöliittymiä, jotka sisältävät erilaisia työkaluja ja widgettejä. Näihin kuuluu esimerkiksi vedä ja pudota -komponentit, joiden avulla myös kansalaiskehittäjät voivat luoda haluamiaan sovelluksia. Lisäksi LCDP:t voivat tarjota käyttäjilleen erilaisia tukipalveluita, kuten esimerkiksi

live-yhteistyökehitystukea, joka auttaa maantieteellisesti hajallaan olevia kehittäjiä työskentelemään yhdessä samojen sovellusten parissa.

Toinen erottava piirre olemassa olevissa LCDP:issa liittyy tuettuun sovellusalueeseen, useimmat LCDP:t esimerkiksi ovat yleiskäyttöisiä ja tukevat minkä tahansa datavaltaisen sovelluksen kehittämistä. LCDP:t tarjoavat käyttäjille valmiita elementtejä, jotka toimivat lähtökohtina sovellusten kehittämiselle, mikä ilmentää niiden uudelleenkäytettävyyden tukiominaisuutta. (Di Rusci et al., 2022)



Kuva 3 Low-code kehitysalustojen keskeiset komponentit ja tukiprosessit (mukaillen Di Rusci et al., 2022)

Ohjelmistojärjestelmien toteuttaminen low-code kehitysalustoilla sisältää useita työkaluilla tuettua vaihetta, joita ovat muun muassa: domain mallinnus, käyttöliittymän määrittely, liiketoimintalogiikan määrittely, integraatiot ulkoisiin palveluihin, sovellusten luominen ja käyttöönotto sekä sovelluksen ylläpito. (Sahay, 2020)

2.2 Low-code alustojen taustat ja historia

Low-code työkalut ovat olleet olemassa konseptina jo muutaman vuosikymmenen ajan, ensimmäisen kerran ilmestyen 1980-luvulla. (Martin, 1982) Useiden vuosikymmenten ajan organisaatioilla oli kaksi eri vaihtoehtoa, kun tarvittiin uusia järjestelmiä; järjestelmä voitiin rakentaa omien kehittäjien avulla tai ostaa ulkoiselta toimittajalta. Nykyään myös kolmas lähestymistapa *Low-code sovellukset*, on kasvattanut suosiota. (Davenport & Johannessen, 2021)

Vuonna 2014 Forrester Reseach lanseerasi termin ”Low-code” raportissa, joka analysoi suurten toimittajien nykyisiä markkina-asemia. Raportin mukaan käsin ohjelmointi on liian hidasta kehittämään ja toimittamaan monia sovelluksia, joita yritykset käyttävät palvellakseen, voittaakseen ja säilyttääkseen asiakkaita. Tämän vuoksi osa yrityksistä on kääntynyt uusien low-code sovellusalustojen puoleen, jotka nopeuttavat sovelluskehitystä huomattavasti vähentäen käsin ohjelmoinnin määrää. (Richardson et al., 2014)

Low-code alustakehityksen muodostumiseen on vaikuttanut useita keskeisiä teknologioita. Mallipohjainen ohjelmistosuunnittelu (MDSE) suosittaa alakohtaisten mallien käyttöönottoa mikä on johtanut siihen, että low-code ohjelmistokehitysalustat ovat omaksuneet mallipohjaisen suunnittelun periaatteet. Loppukäyttäjäkehitys (EUD) on mahdollistanut ohjelmistojärjestelmien käyttäjille, jotka eivät ole ammattimaisia ohjelmistokehittäjiä, luoda, muokata tai laajentaa ohjelmistoartefakteja. Lisäksi on olemassa useita low-code/no-code sovelluskehitystyökaluja, jotka tarjoavat visuaalisia vedä ja pudota -mahdollisuuksia. Nämä teknologiat ovat pyrkineet parantamaan low-code ohjelmistokehityksen taustalla olevia teknologioita. (Alamin et al., 2023)

2.2.1 Low-code alustojen käyttöönotto

Low-code alustat ovat muuttaneet tapaa, jolla yritykset kehittävät ja ylläpitävät sovelluksiaan. Ne tarjoavat liiketoiminnan ammattilaisille mahdollisuuden osallistua sovellusten kehitykseen ilman syvällistä teknistä osaamista, mikä vähentää merkittävästi riippuvuutta ohjelmistokehittäjistä. Low-code alustojen käyttöönotto edistää liiketoiminnan ja IT:n välistä yhteensovittamista ja nopeuttaa uusien ratkaisujen markkinoille tuomista sekä ohjelmistosovellusten päivityksiä ja uusien versioiden julkaisua (Missikoff, 2024).

Low-code alustojen käyttöönotto vaatii organisaatiolta tiettyjä kyvykkyyksiä. Tämä sisältää ymmärryksen perusperiaatteista, kuten järjestelmän kehityksestä sekä kyvyn sovittaa yhteen käyttäjien ja kehittäjien erilaiset ajattelutavat ja tarpeet. Low-code alustojen käyttöönoton merkittävä este on teknologian ja sen käytön tuntemattomuus. Tämän esteen ylittämiseksi

organisaatioiden on investoitava koulutukseen ja tukeen, joka auttaa käyttäjiä ymmärtämään ja hyödyntämään alustaa tehokkaasti. Käyttöönotossa tulisi ottaa huomioon seuraavat seikat:

- Käyttäjien osallistaminen: Low-code alustojen käyttöönotossa on tärkeää ottaa huomioon loppukäyttäjät, sillä heidän odotetaan kykenevän kehittämään sovelluksia ja työnkulkuja ilman erityistä ohjelmointitaustaa tai ammattikehittäjien osallistumista.

- Organisaation kyvykkyydet: Onnistunut low-code alustojen käyttöönotto vaatii organisaatiolta tiettyjä kyvykkyyksiä. Tämä edellyttää perusperiaatteiden hallintaa, kuten järjestelmän kehitysprosessien ymmärtämistä ja taitoa yhdistää käyttäjän ja kehittäjän näkökulmat.

- Teknologian tuntemus: Low-code alustojen käyttöönoton merkittävä este on teknologian ja sen käytön tuntemattomuus. Tämän esteen ylittämiseksi organisaatioiden on investoitava koulutukseen ja tukeen, joka auttaa käyttäjiä ymmärtämään ja hyödyntämään alustaa tehokkaasti.

- Järjestelmän suunnittelu: Low-code alustojen avulla käyttäjät voivat suunnitella omat ajattelutapansa ja esittää ne järjestelmän kehitysmenetelmänä. Tämä vaatii kuitenkin ymmärrystä järjestelmän suunnittelusta ja kehityksestä.

Nämä tekijät ovat tärkeitä, kun otetaan käyttöön low-code alustoja, jotka mahdollistavat sovellusten nopean kehittämisen ja muutosten tekemisen vähemmällä ohjelmoinnilla. (Printz et al., 2024)

Merkittävä haaste low-code-alustojen käyttöönotossa on teknologian ja sen käyttömahdollisuuksien tuntemattomuus. Tämän haasteen voittamiseksi organisaatioiden on panostettava koulutukseen ja tukeen, jotta käyttäjät oppivat ymmärtämään ja hyödyntämään alustaa tehokkaasti. (Printz et al., 2024) Low-code alustojen käyttöönottoon liittyy myös erinäisiä haasteita, joita on kuvattu Taulukko 1:ssä. Low-code alustojen käyttöönotossa voi esiintyä haasteita, kuten tietämättömyys ja huoli alustojen lukkiutumisesta, joustavuudesta, skaalautuvuudesta ja turvallisuudesta. Ihmiset ovat myös usein luonnostaan muutosvastarintaisia ja low-code alustojen yleinen hyväksyntä voi heikentyä yritysten haluttomuuden vuoksi muuttaa toimintatapojaan. Kehitystyökalujen integrointi ja turvallisen koodianalyysin sisällyttäminen kehitysprosessiin ovat myös todettuja haasteita low-code alustojen käyttöönotossa. Eri kehitystyökalujen integrointi yhdeksi yhtenäiseksi ohjelmointiympäristöksi on

monimutkaista, mutta tärkeää käyttäjäkokemuksen kannalta. Turvallisen koodianalyysin integrointi voi olla puutteellista, mikä altistaa tietoturva-aukkoihin ja heikentää turvallisuutta. (Sanchis et al., 2020)

Taulukko 1 Low-code alustojen käyttöönoton haasteita (Sanchis et al., 2020)

HAASTE	KUVAUS
Muutosvastarinta	Ihmiset ovat usein luonnostaan vastahakoisia muutoksille ja tämä voi heikentää low-code- alustojen yleistä hyväksyntää yrityksissä.
Shadow IT / Varjo-IT	Varjo-IT tarkoittaa tilannetta, jossa yrityksen työntekijät ottavat käyttöön IT-ratkaisuja ilman IT-osaston hyväksyntää. Kärsimättömyys low-code alustojen käyttöönotossa voi johtaa varjo-IT:n syntyyn, kun yrityksen IT-osasto ei testaa ja hyväksy käytettävää teknologiaa. Tämä voi johtaa turvallisuusriskeihin ja noudattamisongelmiin.
Reaaliaikaisen datankäsittelyn haasteet	Sovellusten vaatima reaaliaikainen datankäsittely on keskeinen haaste low-code alustoilla.
Integraatiohaasteet	Eri kehitysympäristöjen integrointi yhteen yhtenäiseen ohjelmointiympäristöön (IDE) on haastavaa. Käyttäjäkokemuksen parantamiseksi on tarpeen tarjota korkea integraatiotaso eri kehitystyökalujen välillä.
Turvallisen koodianalyysin integrointi	Kehitystyökalut tarjoavat staattisia turvallisuuskoodianalyysiraportteja, mutta dynaamisten sovellusturvallisuusanalyysityökalujen, jotka testaavat haavoittuvuuksia suorituksen aikana integrointi puuttuu. Tämä

	voi johtaa turvallisuusaukkoihin sovelluksissa.

2.2.2 Alan sovellukset ja trendit

Sovellusten luominen on siirtymässä kohti modulaarisempaa lähestymistapaa, jossa sovellukset koostuvat ja yhdistyvät eri osista. Tämä muutos syrjäyttää perinteiset teknologiat ja organisaatiorakenteet, mikä lisää low-code alustojen ja kansalaiskehittämisen suosiota. Arvioiden mukaan vuoteen 2025 mennessä 70 % uusista sovelluksista luodaan low-code tai no-code teknologioiden avulla, kun vuonna 2020 tämä luku oli alle 25 %. Low-code alustojen suosion kasvu lisää myös kansalaiskehittämisen määrää. Erityisesti tämä koskee liiketoiminnan kehittäjiä, jotka työskentelevät IT-osastojen ulkopuolella ja luovat teknisiä tai analytiikkaan liittyviä ratkaisuja yrityksen sisäiseen tai ulkoiseen käyttöön. (Gartner, 2021)

Forresterin raportti ”The Low-Code And Digital Process Automation Market, 2023 To 2028” näkee kolme mahdollista tulevaisuutta low-code markkinalle: se joko jatkaa nykyisellä suunnallansa, kiihtyy tekoälyn avulla tai hidastuu tekoälyn vuoksi, kun kehittäjät tekevät ohjelmointitehtävänsä tekoälyavustajien kanssa eivätkä tarvitse low-coden tuottavuushyötyjä yhtä paljon. Forresterin mukaan todennäköisin skenaario tällä hetkellä on, että low-code jatkaa nykyistä kasvusuuntaansa. Tämän ennusteen mukaan low-code markkinat kasvaisivat 30 miljardiin dollariin vuoteen 2028 mennessä. (Bratincevic et al., 2024) Fortune Business Insights arvioi, että low-code alustojen markkinoiden arvo oli noin 22,25 miljardia Yhdysvaltain dollaria vuonna 2023 ja sen ennustetaan kasvavan 28,75 miljardiin dollariin vuonna 2024 ja edelleen 264,40 miljardiin dollariin vuoteen 2032 mennessä, mikä vastaa 32,0 % vuotuista kasvuvauhtia vuosina 2024–2032 (Fortune Business Insights, 2023).

Low-code alustoilla tekoälyä hyödynnetään monin tavoin. Tekoälyllä toimivat työkalut, kuten Copilot ja ChatGPT täydentävät low-code ohjelmointia. Yksi keskeinen tapa menetelmien hyödyntämiseen on luonnollisen kielen ohjelmointi (NLP), jossa tekoälyä käytetään generoimaan koodia luonnollisen kielen ohjeista. Tämä on erityisen hyödyllistä suurten palvelurajapintojen (APIs) kanssa työskenneltäessä, koska se vähentävät tarvetta muistaa suuria määriä API-kutsuja ja niiden argumentteja. Toinen tapa hyödyntää tekoälyä on koodin automaattinen optimointi ja virheiden korjaus. Tekoäly voi analysoida kirjoitettua koodia,

tunnistaa mahdolliset virheet tai tehottomuudet ja ehdottaa parannuksia. Lisäksi tekoälyä voidaan käyttää ohjelmistojen suorituskyvyn seurantaan ja optimointiin, analysoimalla suorituskykytietoja, tunnistamalla pullonkauloja ja ehdottamalla tapoja parantaa suorituskykyä. (Rao et al., 2024)

Robottiprosessiautomaatio (RPA) on teknologia, joka mahdollistaa toistuvien ja rutiininomaisten tehtävien automatisoinnin. Yhdessä low-code alustat ja RPA voivat tarjota työkaluja yritysten digitaaliseen muutokseen. Low-code alustat voivat helpottaa RPA-ratkaisujen kehittämistä ja käyttöönottoa, koska ne mahdollistavat nopean prototyypin luomisen ja iteroinnin. Toisaalta RPA voi laajentaa low-code alustojen kykyjä automatisoimalla tehtäviä, jotka olisivat muuten työläitä tai aikaa vieviä toteuttaa manuaalisesti. Molempia teknologioita hyödyntämällä yritykset voivat parantaa tuottavuutta, vähentää virheitä sekä vapauttaa henkilöstön aikaa keskittyä muihin korkeamman tason tehtäviin. (Antwiadjei & Huma, 2024)

2.3 Liiketoimintaprosessien hallinta ja sen merkitys liiketoiminnalle

Liiketoimintaprosessien hallinta (BPM) keskittyy siihen, miten työ suoritetaan organisaatiossa. Sen tavoitteena on tehdä yrityksen toiminnasta mahdollisimman sujuvaa ja tehokasta, parantaen yrityksen prosesseja mahdollisimman toimiviksi ja virheettömiksi. Lisäksi BPM:n avulla pyritään tarjoamaan asiakkaille entistä parempia palveluita. Näin yritys voi saavuttaa parhaan mahdollisen toiminnan eli operatiivisen erinomaisuuden. (Abdelwahab & Helal, 2023) BPM ei ole vain yksittäisten toimintojen parantamista, vaan se keskittyy hallitsemaan kokonaisia tapahtumaketjuja, toimintoja ja päätöksiä, jotka lopulta lisäävät arvoa organisaatiolle ja sen asiakkaille. Näitä tapahtumaketjuja, toimintoja ja päätöksiä kutsutaan myös prosesseiksi. (Dumaset al., 2018)

Yritysten on uudistuttava jatkuvasti kasvaakseen ja menestyäkseen pitkällä aikavälillä. Tämä edellyttää nykyisten parhaiden käytäntöjen ja tulevaisuuden liiketoimintakäytäntöjen yhdistämistä. Parhaat käytännöt takaavat tehokkuuden, kun taas uudet käytännöt tuovat todellisia kilpailuetuja. Seuraavan sukupolven prosessien automatisointi mahdollistaa uusien käytäntöjen toteuttamisen taloudellisesti hyväksyttävällä kustannustasolla. Tämä saavutetaan yhdistämällä joustava liiketoimintaprosessien määrittely ja ohjelmistosovellusten tuki.

Tämä korostaa liiketoimintaprosessien automaation merkitystä liiketoiminnalle. (Kirchmer & Scheer, 2004)

2.3.1 Liiketoimintaprosessien automaation tärkeys ja hyödyt

Liiketoimintaprosessien automatisointi (BPA) on innovatiivinen konsepti, joka hyödyntää teknologiaa liiketoiminnan toimintojen tai prosessien automaattiseen suorittamiseen. Tämä kattaa useita työkaluja, kuten robotiikan prosessiautomaation (RPA), liiketoimintaprosessien hallintajärjestelmät (BPMS) ja tekoälyratkaisut. BPA:n avulla organisaatiot voivat parantaa toiminnallista tehokkuuttaan merkittävästi. Tämä mahdollistaa uusien innovatiivisten liiketoimintamallien toteuttamisen ja uusille markkinoille laajentumisen entistä helpommin. Höytyjen toteutuminen riippuu useista tekijöistä, kuten oikeiden prosessien automatisoinnista oikeilla teknologioilla. (Martinek-Jaguszewska & Rogowski, 2023)

Taulukko 2 esittää liiketoimintaprosessien automatisoinnin etuja, kuten tehokkuuden lisäämistä ja virheiden vähentämistä. Liiketoimintaprosessien automatisointi voi parantaa myös työntekijöiden moraalia ja asiakaskokemusta sekä tukea hajautettuja tiimejä. (Davis, 2021)

Taulukko 2 Liiketoimintaprosessien automaation hyötyjä (Davis, 2021)

Hyöty	**KUVAUS**
Operatiivinen tehokkuus	Automaatio lisää henkilöstön tehokkuutta ja vähentää inhimillisiä virheitä.
Vähemmän fyysistä jätettä	Automaatio vähentää paperijätettä ja säästää varastointitilaa.
Datan eheys	Automaatio varmistaa tarkat ja ajantasaiset tiedot.
Standardisointi	Automaatio yhdenmukaistaa prosessit ja tiedon löytämisen.
Työntekijöiden moraali	Automaatio poistaa rutiininomaisia tehtäviä ja parantaa työntekijöiden moraalia.
Tarkemmat ennusteet	Automaatio helpottaa tietojen keräämistä ja tarkkojen ennusteiden tekemistä.

Hajautettujen tiimien tuki	Automaatio yhdistää etätiimejä ja nopeuttaa prosesseja.
Sääntelyn noudattaminen	Automaatio varmistaa sääntelyn ja lakien noudattamisen.
Parempi kokemus asiakkaille ja kumppaneille	Automaatio parantaa asiakas- ja kumppanikokemusta

2.3.2 Liiketoimintaprosessiautomaation mahdollisuudet ja haasteet

Monet yritykset ovat investoineet liiketoimintaprosessien automatisointiin, mutta useilla yrityksillä puuttuu tarvittava asiantuntemus näiden työkalujen hyödyntämiseen. Tästä huolimatta kuitenkin noin puolet Yhdysvaltalaisista automatisointityökaluja käyttävistä yrityksistä hyödyntää myös tekoälyä tai koneoppimista tehtävien käsittelyssä ja työnkulussa. IT-osastot ovat useimmiten ne, jotka vievät automatisointia eteenpäin yrityksissä. Yritysjohtajat ovat havainneet automatisoinnin suurimmiksi eduiksi kustannussäästöt, datavirheiden vähentämisen ja asiakastyytyväisyyden kasvun. (ICT Monitor Worldwide, 2023)

Koneoppimistekniikoita voidaan käyttää BPM:n parantamiseen ja organisaatioiden tavoitteiden saavuttamiseen. Ne voivat auttaa organisaatioita parantamaan tarjoamiensa palvelujen laatua tukemaan skaalautuvuutta automatisoinnin avulla ja vähentämään kustannuksia. Koneoppimista voidaan käyttää BPM:n parantamiseen esimerkiksi automatisoidun koneoppimisen (AutoML) tekniikoiden avulla. Koneoppimisen ja tekoälyn tekniikoiden viimeaikaiset edistysaskeleet ovat tarjonneet uusia mahdollisuuksia luoda dynaamisia ja sopeutuvia liiketoimintaprosessimalleja. Nämä tekniikat voidaan ottaa käyttöön ennakoivien mallien kehittämiseen, jotka ennakoivat liiketoimintaprosessin mukaisesti tulevia tapahtumia. Koneoppimisen ja tekoälyn tekniikoiden käyttö BPM:ssä tarjoaa uusia mahdollisuuksia luoda dynaamisia ja sopeutuvia liiketoimintaprosessimalleja. Onnistunut toteutus vaatii kuitenkin vahvaa yhteistyötä IT:n ja liiketoiminnan välillä sekä rakenteellista lähestymistapaa prosessien priorisointiin ja automatisointiin. (Abdelwahab & Helal, 2023)

Haasteita liiketoimintaprosessien automaatioon liittyen ovat prosessien jatkuva muutokset ja päivitystarpeet, jotka voivat tehdä automaatiosta haastavaa sillä automatisoidut järjestelmät ovat suunniteltu toimimaan tiettyjen sääntöjen mukaan. BPA voi myös aiheuttaa huolta

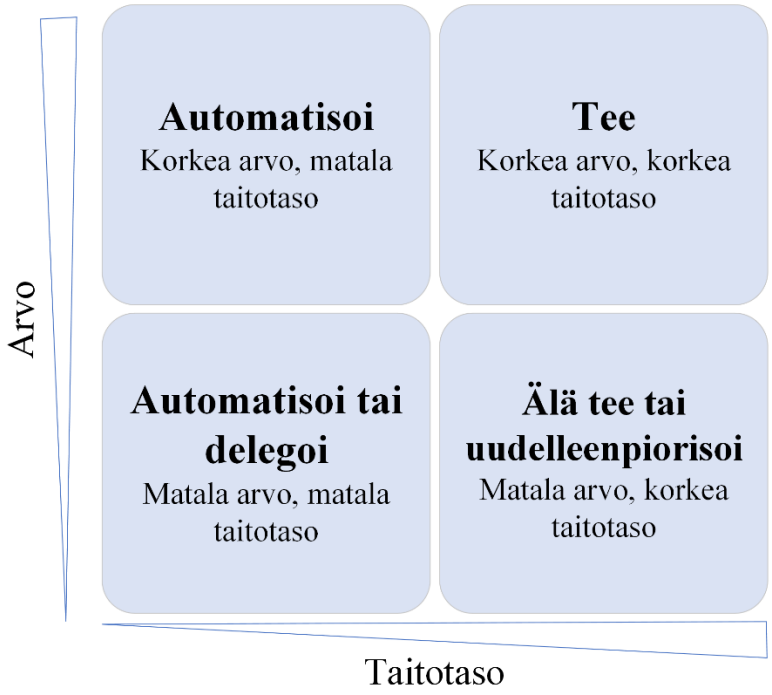
työntekijöissä, jotka pelkäävät työpaikkansa menetystä. Tämä voi johtaa automaation vastustukseen ja vaikeuttaa sen käyttöönottoa. Automatisoinnin käyttöönotto vaatii merkittäviä investointeja uuteen teknologiaan ja koulutukseen. Tämä voi olla este pienille ja keskisuurille yrityksellä, joille ei ole tarvittavia resursseja. BPA:n käyttöönottoon tarvitaan myös vahvaa yhteistyötä IT:n ja liiketoiminnan välillä. Tämä voi olla haasteellista organisaatioissa, joissa nämä kaksi osa-aluetta toimivat erillään toisistaan. (Mishra et al., 2019)

2.4 Low-code alustat liiketoimintaprosessien automaatiossa

Automaatio on keskeinen liiketoimintateknologia vähentää toistuvien, matalan taitotason tehtävien vaikutusta työvoimaan. Kun tarkastellaan liiketoimintaprosesseja ja niiden tehtäviä, ne voidaan usein jakaa seuraaviin neljään neljännekseen:

- Korkea arvo, matala taitotaso: Tehtävät vaativat vain vähäistä erikoisosaamista, mutta ovat kriittisiä tai tuottavat suuren arvon organisaatiolle.
- Korkea arvo, korkea taitotaso: Tehtävät vaativat ihmisen älykkyyttä tai käsittelyä, joita tarvitaan arvon, taitojen tai suhteiden määrittämiseksi.
- Matala arvo, matala taitotaso: Tehtävät vaativat vähäistä erikoisosaamista ja tuottavat vain pienen arvon.
- Matala arvo, korkea taitotaso: Tehtävät vaativat suuren keskittymisen tai taitoja, mutta tuottavat pienen arvon.

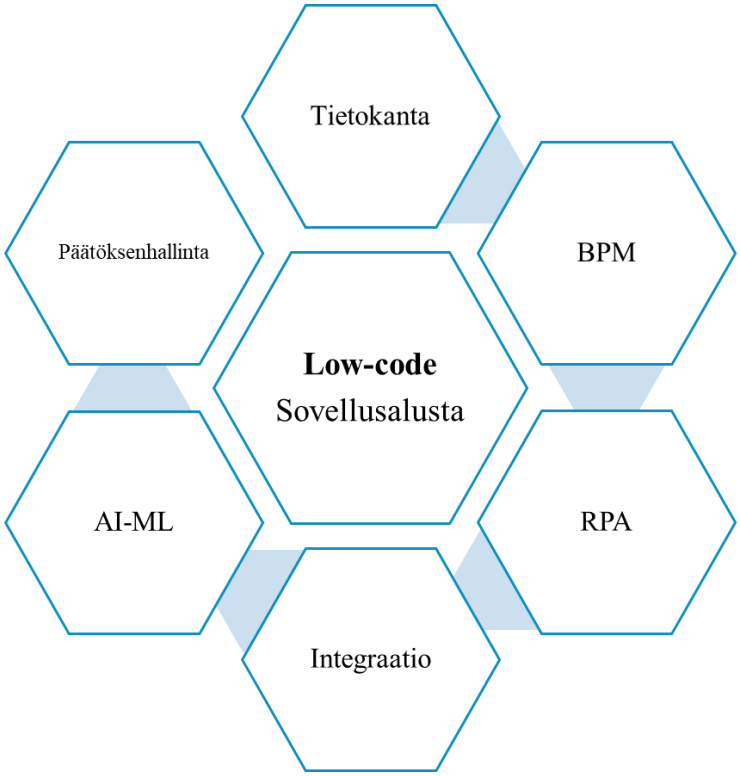
Kuva 4:ssa on esitetty nämä neljännekset. Tehtävät, jotka kuuluvat “matalan taitotason” neljänneksiin, ovat erittäin hyviä ehdokkaita automatisoitaviksi tehtäviksi. (Guilmette, 2022)



Kuva 4 Automaation ehdokkaat (Mukaillen Guilmette, 2022)

Low-code alustat (LCP) voivat auttaa organisaatioita siirtymään manuaalisista prosesseista digitaalisiin ja automatisoituihin prosesseihin. Low-code alustat voivat olla hyödyllisiä muun muassa liiketoimintajohtajille ja toiminnallisille päälliköille, koska ne mahdollistavat myös pienempien prosessien automatisoinnin ilman, että kustannukset ovat suuria. (Express Computer, 2019) Low-code alustat ovat tämän vuoksi nousseet merkittäväksi työkaluksi liiketoimintaprosessien automatisoinnissa. Ne tarjoavat ympäristön, jossa käyttäjät voivat automatisoida prosesseja vähäisellä tai ei lainkaan koodauksella. Tämä mahdollistaa nopeamman prosessien automatisoinnin, mikä parantaa organisaatioiden tuottavuutta ja joustavuutta. (Bock & Frank, 2021)

Low-code alustat koostuvat useista osa-alueista, kuten liiketoimintaprosessien hallinasta (BPM), tietokannoista, ohjelmistorobotiikasta (RPA), integraatioista, tekoälystä (AI) ja koneoppimisesta (ML) sekä päätöksenhallinta. Nämä osa-alueet toimivat yhdessä mahdollistaakseen monimutkaisten liiketoimintaprosessien automatisoinnin ja optimoinnin. (Waszkowski, 2022) Eri osa-alueita on kuvattu Kuva 5:ssa.



Kuva 5 Low-code sovellusalustojen osa-alueita

BPM on keskeinen osa low-code alustoja, koska se mahdollistaa liiketoimintaprosessien mallintamisen, suorittamisen, seurannan ja optimoinnin. Tietokannat taas mahdollistavat tietojen tallentamisen, hakemisen ja analysoinnin, mikä on olennainen osa monia liiketoimintaprosesseja. Ohjelmistorobotiikka (RPA), tekoäly (AI) ja koneoppiminen (ML) tuovat lisäarvoa low-code alustoihin mahdollistamalla tehtävien automatisoinnin, ennusteiden tekemisen ja päätöksenteon tehostamisen. (Bock & Frank, 2021)

2.4.1 Low-code alustojen rooli nykyaikaisessa liiketoiminta-automaatiossa

Low-code alustat ovat keskeisessä roolissa nykyaikaisessa liiketoiminta-automaatiossa. Ne mahdollistavat sovellusten nopean kehittämisen ja muutosten tekemisen, mikä on erityisen tärkeää liiketoiminta-automaatiossa. (Bock & Frank, 2021) Alustat tarjoavat ratkaisuja digitaalisen transformaation haasteisiin, joita monet yritykset kohtaavat erityisesti ohjelmistokehittäjien puutteen ja ohjelmistokehitysprojektien tehottomuuden tai epäonnistumisen osalta (Vincent et al., 2020).

Low-code alustat tarjoavat korkeaa tuottavuutta ja monitoimintoja keskitetysti ja ne voivat palvella laajaa joukkoa käyttäjiä organisaatiossa (Vincent et al., 2020). Alustat tehostavat liiketoimintaprosesseja yhdistämällä erilliset tehtävät saumattomiksi työnkuluiksi. Tämä parantaa automaatiota ja tarjoaa reaaliaikaisen näkymän toimintoihin, mikä lisää liiketoiminnan arvoa ja mahdollistaa low-code ratkaisujen maksimaalisen hyödyntämisen liiketoiminnan automaatiossa (Antwiadjei & Huma, 2021).

On kuitenkin tärkeää huomata, että termi ‘low-code’ voi olla ongelmallinen. Termiä käytetään tällä hetkellä usein epäjohdonmukaisella tavalla myymään erittäin heterogeenisiä kehitysympäristöjä (Waszkowski, 2022). Tämä epäjohdonmukaisuus voi johtaa väärinkäsityksiin ja sekaannuksiin siitä mitä low-code alustat todella ovat ja mitä ne voivat tarjota. Tästä huolimatta low-code alustat ovat tärkeä suuntaus ohjelmistokehityksessä, joka voi auttaa yrityksiä vastaamaan nopeasti muuttuviin markkinavaatimuksiin ja edistämään digitaalista transformaatiota. Low-code alustojen avulla voidaan kehittää sovelluksia vähemmällä määrällä ohjelmointia, mikä mahdollistaa sovellusten nopeamman kehityksen ja sovellusten automatisoinnin (Sanchis et al., 2020).

2.4.2 Prosessien mallintaminen low-code alustoilla

Prosessimallinnus on lähestymistapa, jolla havainnollistetaan visuaalisesti, kuinka yritykset harjoittavat toimintaansa. Prosessimallinnuksessa määritellään ja havainnollistetaan kokonaisuuksia, toimintoja, mahdollistajia ja niiden välisiä suhteita. Prosessimallinnusta käytetään laajasti organisaatioissa keinona lisätä tietoisuutta ja ymmärrystä liiketoimintaprosesseista sekä vähentämään organisaation monimutkaisuutta. (Bandara et al., 2005).

Yleisesti käytettyjä prosessimallinnuskieliä low-code alustoilla ovat esimerkiksi BPMN (Business Process Model and Notation) ja UML (Unified Modeling Language). (Kang et al., 2024) UML on laajalti tunnustettu mallinnuskieli, jota ohjelmistosuunnittelijat käyttävät ohjelmistojärjestelmän suunnittelun visualisoimiseen ja ilmaisemiseen. UML-luokkamallit edustavat tyypillisesti tietokantaskeemoja, mikä tekee niiden muuntamisesta tärkeän vaiheen kehityksessä. (Bubalo & Tankovic, 2023) Liiketoimintaprosessin mallinnusjärjestelmä (BPMN) on esimerkki mallinnustekniikasta, joka luotiin tarjoamaan merkintätapa, jonka kaikki liiketoimintaprosessin elinkaareen osallistuvat voivat ymmärtää. (Sahay et al., 2023)

Nämä kielet tarjoavat valmiita komponentteja ja teknologioita, joiden avulla sovelluksen kehittäjät voivat rakentaa sovelluksen toiminnallisuudet ja toimintalogiikat erilaisten visuaalisten elementtien avulla. Tämä nopeuttaa sovelluskehitystä huomattavasti, koska kaikkea ei tarvitse koodata alusta alkaen itse. Low-code alustat tarjoavat myös usein mahdollisuuden hyödyntää valmiita sisäänrakennettuja ominaisuuksia, jotka sujuvoittavat sovellusten kehittämistä huomattavasti. Tämä tekee niistä erittäin tehokkaita työkaluja liiketoimintaprosessien mallintamiseen. (Bucaioni et al., 2022)

Liiketoimintaprosessit ovat keskeisiä yritysstrategian toteuttamisessa ja niiden hallinta (BPM) keskittyy parantamaan liiketoiminnan suorituskykyä. BPMN on graafinen liiketoimintaprosessien mallinnuskieli, jonka OMG on kehittänyt. Sen tavoitteena on standardisoida liiketoimintaprosessit ja se on muihin prosessimallinnuskieliin verrattaen käyttäjäystävällinen mallinnuskieli. Liiketoimintaprosessit esitetään prosessikaavioina ja jokainen kaavio koostuu joukosta mallinnuselementtejä. (Kang et al., 2024) BPMN:ssä liiketoimintaprosesseja kutsutaan usein työnkuluiksi ja ne määritellään käyttämällä spesifiä merkintätyyliä, johon kuuluu monia mallintamisen rakenteita toiminnoille, tapahtumille, porteille, artefakteille ja niin edelleen. (Sahay et al., 2023) Kaavamainen prosessimallinnus tarjoaa käyttäjälle tarkan kuvauksen siitä, miten prosessin eri osat toimivat ja vuorovaikuttavat, varmistaen että mallinnettu prosessi vastaa todellisen prosessin toimintaa. (Kang et al., 2024)

2.5 Aiempi kirjallisuus low-code alustojen käytöstä liiketoimintaprosessiautomaatiossa

Liiketoimintaprosessien automatisointi ei ole uusi käsite ja useat organisaatiot harkitsevat uudelleen prosessien automatisointistrategiaansa uusien työskentelytapojen, kuten etätyön tai taloudellisten haasteiden keskellä. Organisaatiot eri toimialoilla kiihdyttävät digitaalista muutosta parantaakseen asiakaskokemusta, lisätäkseen työntekijöiden tuottavuutta, vähentääkseen kustannuksia ja varmistaakseen kilpailukykynsä markkinoilla. (PR Newswire, 2021) Low-code alustat helpottavat liiketoimintaprosessien kehittämistä vähemmällä ohjelmointityöllä sekä ne mahdollistavat liiketoimintaprosessien nopean digitalisoinnin. (Express Computer, 2019)

2.5.1 Low-code alustojen hyödyt liiketoimintaprosessiautomaatiossa

Low-code alustat vähentävät laajan koodausosaamisen tarvetta, mikä mahdollistaa laajemman käyttäjäjoukon aktiivisen osallistumisen liiketoimintaprosessien optimointiin. (Khorram et al, 2020) Low-code alustat vähentävät rutiinitehtävien vaatimaa työmäärää, mikä parantaa tuottavuutta ohjelmistokehitysprojekteissa. Lisäksi low-code alustat sisältävät kaikki tarvittavat työkalut ja komponentit yhdessä ympäristössä, mikä tekee kehitystyöstä sujuvampaa. (Bock & Frank, 2021)

Krishnaraj et al. (2022) tutkivat low-code-kehitysalustoja ja niiden vaikutusta ohjelmistokehitykseen vertailevassa tutkimuksessaan ”Comparative Study on Various Low Code Business Process Management Platforms”. Tutkimuksen päätavoitteena oli vertailla ja erottaa erilaisia low-code kehitysalustoja, mukaan lukien Appian, SalesForce Lightning ja Caspio. Tutkimuksen mukaan low-code kehitysalustat nopeuttavat sovellusten toimitusta sekä antavat kehittäjille mahdollisuuden rakentaa nopeasti sovelluksia tiettyihin liiketoimintakäyttötapauksiin. Tämän lisäksi low-code alustojen rajoitukset ovat vähäisempiä verrattuna perinteiseen kehitysmenetelmään ja nämä rajoitukset katoavat, kun low-code alustat julkaisevat tulevat versiopäivityksensä. Yritysasiakkaat voivat hyödyntää low-code alustoja, jotka eroavat toisistaan ominaisuuksien ja suorituskyvyn osalta. Toisaalta kaikilla low-code alustoilla on samanlaiset rajoitukset. Nämä rajoitukset voivat liittyä esimerkiksi käyttöliittymien kustomointiin, integraatioihin, jotka vaikuttavat yritysten vanhoihin järjestelmiin sekä kehittäjien puutteeseen työskennellä low-code alustojen parissa. (Krishnaraj et al., 2022)

Ciucan-Rusu et al. (2023) tarkastelivat tutkimuksessaan '”Low-code solutions integration for digital process automation - application in management” low-code alustojen käyttöä prosessiautomaatiossa ja niiden vaikutusta projektinhallintaan. Tutkimusryhmä käytti Microsoft Power Platformin low-code ratkaisua projektin toteutuksessa parantaakseen projektiryhmän suorituskykyä. Low-code ratkaisulla saavutettiin seuraavia hyötyjä:

- Paperityön vähentäminen: ratkaisujen avulla voitiin vähentää paperityötä, joka teki prosesseista tehokkaampia.
- Toistuvien tehtävien poistaminen: Automaation avulla voitiin poistaa toistuvia ja aikaa vieviä tehtäviä, mikä vapautti työtekijöiden aikaa muihin tehtäviin.

- Projektiryhmän motivaation lisääminen: kun projektin jäsenet pystyivät keskittymään merkityksellisempiin tehtäviin, se lisäsi heidän motivaatiotaan.
- Henkilökohtaisten digitaalisten taitojen parantaminen: työskentely low-code alustojen parissa auttoi työntekijöitä kehittämään heidän digitaalisia taitojaan.
- Tehokkaampi ajanhallinta: automaation ja digitaalisten työkalujen avulla voitiin hallita aikaa tehokkaammin.
- Avoimempi ja parempi dokumentoitu taloudellinen hallinta: low-code ratkaisun avulla voitiin parantaa taloudellisen tiedon läpinäkyvyyttä ja dokumentointia. (Ciucan-Rusu et al., 2023)

Waszkowski (2019) artikkelissa ”Low-code platform for automating business processes in manufacturing” kuvataan liiketoimintaprosessien automaation käyttöä low-code alustalla tuotantoprosessien automatisoinnissa. Artikkelin mukaan liiketoimintaprosessien automaation hyötyinä ovat muun muassa nopea kehitys, kustannussäästöt sekä vähempi määrä ohjelmointia. Lisäksi low-code alustat mahdollistavat yrityksen omien työntekijöiden hyödyntämisen sovellusten kehityksessä. Tämä voi johtaa parempiin tuloksiin, koska työntekijät ymmärtävät yrityksen tarpeet ja voivat suunnitella ratkaisuja, jotka vastaavat näitä tarpeita. (Waszkowski, 2019)

Low-code alustat mahdollistavat prosessien automatisoinnin, jotka vielä äskettäin toimivat sähköpostin tai paperin välityksellä. Tämä voi parantaa tehokkuutta päivittäisissä työprosesseissa ja tuoda merkittäviä positiivisia vaikutuksia pitkällä aikavälillä. Lisäksi työntekijät voivat siirtää toistuvat ja tylsät tehtävänsä low-code sovelluksen tehtäväksi. Tämä vapauttaa työntekijät keskittymään työnsä mielenkiintoisempiin ja monipuolisempiin osiin sekä rikastuttaa heidän työtään. Työprosessien digitalisointi ja automatisointi voivat parantaa niiden laatua poistamalla inhimilliset virheet. Työntekijät voivat myös optimoida työprosessejaan ja parantaa niitä. Lisäksi, kun kehitetään sovellusta, prosessin logiikka on määriteltävä, mikä mahdollistaa työvaiheiden tarkistamisen ja uusien lisäämisen tai poistamisen. Tämä auttaa myös seuraamaan prosesseja jälkikäteen. Low-code alustoilla voidaan nopeuttaa liiketoimintaprosessien automatisointia ja lyhentää kehitysaikoja. (Elshan et al., 2023)

Taulukko 3 kuvaa näitä kirjallisuudessa havaittuja low-code alustojen hyötyjä liiketoimintaprosessiautomaatiossa kootusti.

Taulukko 3 Low-code alustojen hyötyjä liiketoimintaprosessiautomaatiossa

HYÖTY	KUVAUS
Sovellusten nopeampi toimitus	Mahdollistaa sovellusten nopean kehittämisen ja toimituksen, mikä parantaa yritysten reaktiokykyä ja tehokkuutta. (Krishnaraj et al., 2022)
Koodausosaamisen tarpeen vähentäminen	Vähentää tarvittavan koodausosaamisen määrää, jolloin myös ei-tekniset käyttäjät voivat osallistua kehitystyöhön. (Khorram et al., 2020)
Tuottavuuden parantaminen	Vähentää rutiininomaisia tehtäviä ja yhdistää työkalut yhteen ympäristöön, mikä tekee kehitystyöstä sujuvampaa ja tehokkaampaa. (Bock & Frank, 2021)
Työkalujen ja komponenttien integrointi	Tarjoaa kokonaisvaltaisen ympäristön, jossa kaikki tarvittavat työkalut ovat yhdessä paikassa, parantaen kehitystyön sujuvuutta. (Bock & Frank, 2021)
Työprosessien tehostus	Parantaa prosessien tehokkuutta poistamalla virheitä ja tarjoamalla mahdollisuuksia prosessien optimointiin. (Elshan et al., 2023)

2.5.2 Low-code alustojen haasteet liiketoimintaprosessiautomaatiossa

Low-code alustat ovat muodostuneet tehokkaiksi työkaluiksi, jotka mahdollistavat yritysten prosessien kehittämisen ketterämmiksi ja paremmiksi. Ne tarjoavat mahdollisuuden luoda laajan erilaisia ratkaisuja, aina yksinkertaisista automaatiosovelluksista monimutkaisiin toteutuksiin. Kuitenkin low-code alustojen soveltuvuudessa on rajoituksia, sillä ne eivät välttämättä kykene vastaamaan kaikkien monimutkaisten liiketoimintaprosessien vaatimuksiin. Tästä syystä on suositeltavaa aloittaa automatisoimalla ensin yrityksen päätoimialan prosessit ketterämmiksi ja nopeammiksi. Kun nämä prosessit on optimoitu, voidaan laajentaa automatisointia muihin osa-alueisiin. (Elshan et al., 2023)

Low-code alustojen heterogeenisyys on yksi merkittävä haaste. Alustat eroavat toisistaan huomattavasti toiminnallisuuden, teknologioiden ja sovellusmahdollisuuksien suhteen. Tämä tarkoittaa, että tietyt alustat soveltuvat paremmin yksinkertaisiin tietojen hallintatehtäviin, kun taas toiset pystyvät tukemaan laajempia ja monimutkaisempia liiketoimintasovelluksien konfiguraatioita, integrointia ja kehittämistä. (Bock & Frank, 2021)

Vaikka low-code alustat voivat tuottaa tuottavuushyötyjä vähentämällä rutiinitehtävien vaatimaa työmäärää, ne voivat olla rajoitettuja monimutkaisemmissa projekteissa. (Bock & Frank, 2021) Yksi keskeinen haaste on tämän vuoksi monimutkaisten liiketoimintaprosessien tehokas automatisointi. Vaikka useat low-code alustat tarjoavat työkaluja ja valmiita komponentteja, näiden käyttömahdollisuudet voivat olla rajoitettuja monimutkaisemmissa projekteissa. Esimerkiksi monimutkaiset työvirrat ja päätöksentekomallit voivat vaatia enemmän räätälöintiä ja integraatiota kuin mitä low-code alustat standardinomaisesti tarjoavat. (KPMG, 2023).

Ratkaisuina on kehitetty alustoja, jotka tukevat laajennettua ohjelmointia ja mukautuksia. Esimerkiksi JourneyApps on omaksunut "code-first" -lähestymistavan, jossa kehittäjät voivat kirjoittaa mukautettua koodia monimutkaisten tehtävien suorittamiseksi, mikä lisää alustan joustavuutta ja kykyä käsitellä monimutkaisia työprosesseja. (JourneyApps, 2023)

Monimutkaisemmissa projekteissa low-code alustojen suorituskyky ja skaalautuvuus voivat nousta haasteiksi. Nämä haasteet ovat ratkaistu kehittämällä alustoja, jotka kykenevät käsittelemään suuria tietomääriä ja vaativia prosesseja tehokkaasti. Esimerkiksi valmistusteollisuudessa low-code alustat ovat onnistuneet automatisoimaan monimutkaisia prosesseja, mikä on parantanut tehokkuutta ja vähentänyt manuaalisen työn tarvetta. (Waszkowski, 2019)

Datanhallinnan osalta haasteita voi ilmetä, kun mallinnettavat prosessit ovat suuria ja monimutkaisia. Erityisesti BPMN mallinnuskieltä käyttäessä, kun diagrammit kasvavat suuriksi eli niihin liittyy useita tiedostoja, aliprosesseja, kutsutoimintoja ja päätöstaulukoita, datanhallinnasta voi tulla merkittävä ongelma. (Funk et al, 2023)

Low-code alustojen käyttö voi tuoda mukanaan myös hallinnollisia haasteita. Kansalaiskehittäjät, jotka mallintavat uusia prosesseja alustoilla saattavat luoda ratkaisuja, jotka eivät toimi tai skaalaudu hyvin ja haasteita liittyy myös tukeen ja ylläpitoon. Jos kansalaiskehittäjä lähtee yrityksestä, voi syntyä tilanne, jossa kukaan ei tiedä miten järjestelmää ylläpidetään.

Hallinnollisia haasteita voidaan hallita asianmukaisella suunnittelulla ja koulutuksella, esimerkiksi järjestämällä sovelluksen tiedonsiirron kansalaiskehittäjältä ammattilaiselle tai kehittämällä ratkaisuja hybridimallilla, jossa kansalaiskehittäjä tekee suurimman osan ratkaisusta ja siirtää sen kehittäjälle viimeistelyä varten. (Davenport & Johannessen, 2021)

Liiketoimintaprosessien testaus on olennainen osa sovelluskehitystä, jossa kansalaiskehittäjien rooli sekä teknisen tietämyksen puute testaustoiminnoissa tuovat omat haasteensa. Kansalaiskehittäjät tarvitsevat apua testikohtaisten elementtien oikeaan käyttöön testimalleja suunniteltaessa ja korkean tason testiautomaation toteuttaminen voi olla haastavaa (Khorram et al., 2020).

Turvallisuus on myös keskeinen huolenaihe sovellusten kehityksessä. Ei-teknisten käyttäjien mahdollistaminen rakentamaan low-code alustoilla voi johtaa puutteelliseen näkyvyyteen organisaatiossa käytössä oleviin sovelluksiin ja lisätä haavoittuvuutta turvallisuusriskeille. Ratkaisuna on valita alusta, jolla on sisäänrakennettuja turvallisuusominaisuuksia, kuten pääsynvalvonta, salaus ja haavoittuvuustestaus (Bunce, 2024).

Taulukko 4 kuvaa näitä low-code alustojen haasteita liiketoimintaprosessiautomaatiossa kootusti.

Taulukko 4 Taulukko 3 Low-code alustojen haasteita liiketoimintaprosessiautomaatiossa

HAASTE	KUVAUS
Heterogeenisyys	Eri low-code alustojen suuri eroavuus toiminnallisuuden ja teknologioiden osalta, mikä vaikeuttaa alustojen vertailua ja valintaa. (Bock & Frank, 2021)
Monimutkaisempien prosessien rajoitukset	Alustat voivat olla rajoitettuja monimutkaisissa projekteissa, joissa tarvitaan enemmän räätälöintiä ja integraatiota. (Bock & Frank, 2021)
Skaalautuvuus ja suorituskyky	Haasteet suurempien tietomäärien ja vaativien prosessien käsittelyssä voivat rajoittaa suorituskykyä. (Waszkowski, 2019)

Turvallisuus	Ei-teknisten käyttäjien luomat sovellukset voivat lisätä turvallisuusriskejä, ellei alustalla ole riittäviä turvatoimia. (Bunce, 2024)
Hallinnolliset haasteet	Kansalaiskehittäjien luomat ratkaisut voivat johtaa hallinnollisiin haasteisiin ja tuen puutteeseen, erityisesti jos kehittäjä vaihtuu. (Davenport & Johannessen, 2021)

3 Tutkimusmenetelmät

Tutkimuksen tavoite ja tutkimuskysymykset toimivat lähtökohtina tutkimusmenetelmien valinnalle ja ohjaavat tutkimuksen rakennetta (Hirsjärvi & Hurme, 2022). Tässä pääluvussa esitellään työn tutkimuslähestymistapa, tutkimusmenetelmät sekä syyt tutkimusmenetelmien valintaan. Lisäksi käydään läpi tarkemmin tiedonkeruuprosessin kulkua sekä analysoidaan kerättyä aineistoa sekä sen luotettavuutta.

3.1 Tutkimuslähestymistapa

Tutkimuksen tavoitteena on ymmärtää low-code alustojen haasteet ja hyödyt liiketoimintaprosessien automatisoinnissa. Tätä varten suoritettiin kirjallisuuskatsaus, jonka avulla saatiin yleiskuva low-code alustoista, niiden taustoista ja historiasta, liiketoimintaprosessien hallinnasta ja sen merkityksestä sekä low-code alustojen roolista liiketoimintaprosessien automaatiossa. Lisäksi tarkasteltiin aiempia tutkimuksia low-code alustojen hyödyistä ja haasteista liiketoimintaprosessien automatisoinnissa. Kirjallisuuskatsauksen tavoitteena oli luoda selkeä ja kattava yleiskuva siitä, miten low-code alustat ja liiketoimintaprosessien automaatio liittyvät toisiinsa ja miten näitä teknologioita voidaan hyödyntää yhdessä. Tämä kokonaisvaltainen ymmärrys auttaa yrityksiä tunnistamaan low-code alustojen potentiaalin ja niiden tarjoamat mahdollisuudet liiketoimintaprosessien tehostamisessa. Samalla se auttaa tunnistamaan mahdolliset haasteet ja rajoitukset, jotta näiden teknologioiden käyttöönotto ja hyödyntäminen voidaan suunnitella ja toteuttaa mahdollisimman tehokkaasti.

Kirjallisuuskatsauksen tiedonhaku tehtiin yliopiston tarjoamien resurssien avualla. Tiedonhaku aloitettiin tarkastelemalla aiheeseen liittyviä artikkeleita yleisellä tasolla, minkä jälkeen keskityttiin tarkemmin avainsanoihin ja niiden yhdistelmiin. Hakemiseen käytettiin seuraavia tietokantoja ja hakukoneita: Google Scholar, LUT Primo, Scopus, ProQuest, Wiley Online Library, ResearchGate, SpringerLink, IEEE Xplore ja JSTOR. Seuraavia avainsanoja ja niiden yhdistelmiä käytettiin tiedonhakuun: low-code platforms, process automation, business process management, business process automation, low-code development, business process modeling ja workflow automation.

Empiirinen tutkimus toteutettiin käyttämällä laadullisia eli kvalitatiivisia menetelmiä. Kvalitatiivinen tutkimus pyrkii kontekstuaaliseen tulkintaan ja toimijoiden näkökulman ymmärtämiseen. Kvalitatiivisen tutkimuksen malli tulee humanistisista tieteistä ja tulkitsevasta tutkimusotteesta (Glesne & Peshkin, 1992). Hammersleyn (1992) mukaan kvalitatiivinen tutkimus keskittyy merkityksiin, kun taas kvantitatiivinen tutkimus tutkii käyttäytymistä. Kvalitatiivinen suuntaus olettaa, että tutkimuksen kohde ja tutkija ovat vuorovaikutuksessa. Kvalitatiivinen tutkimus tuo esille tutkittavien havainnot tilanteesta ja antaa mahdollisuuden heidän menneisyytensä ja kehitykseensä liittyvien tekijöiden huomioimiseen (Hirsjärvi & Hurme, 2022). Kvalitatiivinen lähestymistapa valittiin tukemaan tutkimuksen tavoitteita, jotta voidaan saada syvällinen ymmärrys low-code alustojen hyödyistä ja haitoista prosessien automatisoinnissa valitun kohdeyrityksen kontekstissa. Kvalitatiivinen tutkimus myös auttaa ymmärtämään käytännön haasteita ja mahdollisuuksia low-code alustojen käytöstä, mitkä mahdollisesti tukevat liiketoimintapäätöksiä.

Empiirisessä tutkimuksessa käytettiin pääasiallisena tutkimusmenetelmänä asiantuntijahaastatteluja, jotka suoritettiin eri roolien edustajille yrityksessä. Asiantuntijahaastattelut eivät ole oma erillinen haastattelumenetelmänsä, vaikka asiantuntijat ovatkin erityinen ryhmä haastateltavia ja heidän haastattelemisessaan on omat erityispiirteensä (Hyvärinen et al., 2017). Haastattelut tehtiin puolistrukturoituina asiantuntijahaastatteluina. Puolistrukturoiduissa haastatteluissa kysymykset on määrätty ennalta, mutta haastattelija voi vaihdella niiden sanamuotoa. Menetelmälle on ominaista, että osa haastattelun näkökohdista on lyöty lukkoon, mutta ei kaikkia (Hirsjärvi & Hurme, 2022). Haastattelut käytiin teemahaastatteluina, joissa haastatteluiden teema-alueet ja aihepiirit olivat kaikille haastateltaville samat. Teemahaastattelu on puolistrukturoitu haastattelumenetelmä, jossa haastattelu kohdennetaan tiettyihin teemoihin. Oleellisinta teemahaastattelussa on, että yksityiskohtaisten kysymysten sijaan haastattelu etenee ennalta määriteltyjen keskeisten teemojen varassa. Menetelmä ottaa huomioon sen, että ihmisten tulkinnat asioista ovat keskeisiä ja merkitykset syntyvät vuorovaikutuksessa. (Hirsjärvi & Hurme, 2022) Haastattelut perustuivat etukäteen suunniteltuihin teemoihin ja roolikohtaisiin kysymyksiin, joita muokattiin ja niihin syvennettiin haastattelun aikana. Haastatteluilla haluttiin saada syvällistä tietoa ja ymmärrystä low-code alustojen haasteista ja hyödyistä tässä tutkitussa ekosysteemissä.

3.2 Aineiston keruu

Tutkimuksessa käytettiin pääasiallisena menetelmänä puolistrukturoituja teemahaastattelu asiantuntijahaastatteluja, jotka suoritettiin asiantuntijaroolissa sekä toteuttavassa roolissa työskenteleville henkilöille, jotka työskentelivät kohdeyrityksessä.

Tutkimuksessa haastateltiin viittä eri kohdeyrityksessä työskentelevää asiantuntijaa. Hyvärinen et al. (2017) kuvaa asiantuntijuutta monimuotoiseksi ilmiöksi ja asiantuntijoiden olevan henkilöitä, joilla erityistä tietoa tutkittavasta asiasta, jota ei usealla muulla ole. Asiantuntijalle on usein asemansa tai tehtäviensä myöstä karttunut erityisiä taitoja ja tietoa. (Hyvärinen et al., 2017) Haastateltavat jaoteltiin kahteen eri kategoriaan heidän työtehtäviensä mukaan: johtaviin ja toteuttaviin. Riippuen henkilön roolista, haastateltaville esitettiin hieman erityyppisiä haastattelukysymyksiä. Johtavan aseman henkilöiden työtehtäviin kuului liiketoimintavastuita sekä seuraamista, että johdettavan yksikön toiminta on linjassa yrityksen strategiaan. Toteuttavassa roolissa työskentelevät haastateltavat suunnittelivat ja määrittelivät ratkaisuja low-code alustoilla asiakkaille ja yrityksen omiin tarpeisiin sekä toteuttivat ratkaisuja. Molempien kategorioiden haastateltavat toimivat myös asiakasrajapinnassa. Haastateltavien tiedot low-code alustoita ja prosessiautomaatiosta olivat pääosin karttuneet työkokemuksen käytännön työn sekä oman kiinnostuneisuuden kautta. Kokemusta low-code alustoista haastateltavilla oli 3–15 vuoden ajalta.

Haastattelukysymykset johdettiin teoreettisesta viitekehyksestä, joka esiteltiin luvussa 2 Teoreettinen viitekehys. Haastatteluissa kysytyt kysymykset löytyvät *liitteestä 1 Asiantuntijahaastattelujen puolistrukturoidut kysymykset.* Asiantuntijahaastattelukysymyksissä keskityttiin seuraaviin teemoihin:

- Teema 1: Monimutkaiset prosessiautomaatiot
- Teema 2: Tuotteen soveltuvuus ja ominaisuuksien puutteet
- Teema 3: Suorituskyky ja skaalautuvuus
- Teema 4: Kansalaiskehityksen haasteet ja hyödyt
- Teema 5: Turvallisuus ja tiedonsiirto
- Teema 6: Low-code prosessiautomaation hyödyt

Kysymyslista kattoi monta low-code alustojen liiketoimintaprosessien automaatioon liittyvää osa-aluetta, mahdollistaen joustavan ja tutkivan lähestymistavan aineiston keruuseen. Vaikka teemat ja kysymykset olivat määritelty jo ennen haastatteluita, haastateltavilta

kysyttiin myös tarkentavia kysymyksiä sekä lisäkysymyksiä. Lisäkysymyksiä esitettiin haastateltaville, mikäli haastatteluvastauksissa nousi uusia teemoja tai aihealueita, joita ei ollut kysymyslistauksissa mukana.

Haastattelukysymykset esitettiin haastateltaville PowerPoint-esityksen avulla. Jokaiselle teemalle oli oma dia, mikä helpotti keskittymistä kyseiseen aiheeseen. Koska kysymyksissä esiintyi useita termejä, kuten *low-code alusta, kansalaiskehitys ja liiketoimintaprosessiautomaatio*, nämä termit määriteltiin dioissa. Määritelmät perustuivat aiemmin tehtyyn kirjallisuuskatsaukseen. Haastattelun aikana kerrottiin myös, että termeihin liittyviä kysymyksiä saa esittää, jotta kaikilla olisi yhteinen ymmärrys teemoista. Tämä teki eri vastauksista vertailukelpoisempia keskenään. Tutkija vältti näyttämästä ennakkoasenteita kommentoimalla vastauksia tai käyttämällä ei-verbaalisia eleitä.

Haastateltaville annettiin mahdollisuus ohjata keskustelua omien kiinnostuksen kohteidensa ja keskeisten aiheiden suuntaan haastattelun aikana. Haastattelun lopussa haastateltavat saivat mahdollisuuden nostaa esille tärkeitä, aiemmin käsittelemättömiä aiheita tai lisätä kommentteja jo käsiteltyihin teemoihin. Erityistä huomiota kiinnitettiin aiheisiin, joita haastateltavat korostivat tai joista heillä oli erityisen paljon tietämystä.

Haastattelut toteutettiin etäyhteydellä Teams-alustalla elokuussa ja syyskuussa vuonna 2024. Kaikki haastattelut nauhoitettiin ja niiden kesto vaihteli keskustelun laajuuden ja sujuvuuden mukaan. Keskimäärin haastattelut kestivät 50–70 minuuttia. Haastatteluiden tallenteet litteroitiin ja analysoitiin teemallisen analysoinnin menetelmin. Analyysin avulla pyrittiin tunnistamaan keskeiset teemat, trendit ja havainnot low-code alustojen käytöstä, hyödyistä ja haasteista eri roolien näkökulmasta.

3.3 Aineiston analyysi ja luotettavuus

Laadullisessa eli kvalitatiivisessa tutkimuksessa kerättyä aineistoa tarkastellaan ainoastaan tietyn teoreettisen ja metodologisen viitekehyksen kautta. Aineistoa analysoidessa kiinnitetään huomiota siihen mikä on teoreettisen viitekehyksen ja tutkimuksen kysymysasettelun kannalta olennaisinta, vaikka samassa tutkimuksessa aineistoa on mahdollista tarkastella eri näkökulmista. (Alasuutari, 2011) Tässä laadullisessa tutkimuksessa tutkitaan tietyn ekosysteemin asioita ja sen tulokset ovat suosituksia sekä havaintoja kerätystä aineistosta.

Haastatteluaineiston luottavuus riippuu sen laadusta. Jos esimerkiksi litterointi ei ole yhdenmukaista tai tallenteiden kuuluvuus huono, ei aineistoa voida sanoa luotettavaksi. Kvalitatiivisessa tutkimuksessa tutkija myös saattaa vaikuttaa saatavaan tietoon jo tietojen keruuvaiheessa sekä siinä on kyse tutkija tulkinnoista. (Hirsjärvi & Hurme, 2022) Tämän työn aineisto pyrittiin litteroimaan aina heti käytyjen asiantuntijahaastatteluiden jälkeen, jotta käytyjen keskustelujen kieli ja vuorovaikutus saatiin litteroitua yhdenmukaisesti. Myös haastattelut tallennettiin kahdella eri laitteella, jotta teknisten haasteiden riskit saatiin minimoitua.

Myös validius on käsite luotettavuuden ja uskottavuuden arvioimiseksi. Tutkimuksen aineisto uskottavuuden määrittämiseksi on osoittaa vastaavuus sekä tutkijan tulkintojen että tutkittavien tulkintojen välillä. Tässä menetelmässä haastateltavat saavat itse myös tutustua tutkijan tulkintoihin. (Hirsjärvi & Hurme, 2022) Kun kaikki haastattelut oli pidetty sekä tulokset muodostettu, käytiin tulokset kahden asiantuntijan kanssa läpi. Läpikäynnissä asiantuntijoille annettiin vielä mahdollisuus kommentoida saatuja tuloksia.

4 Tulokset ja analyysi

Tässä pääluvussa esitellään työn kohdeyritys, analysoidaan haastatteluaineistoa sekä esitellään haastattelujen keskeiset tulokset. Haastatteluaineisto analysoitiin teemallisen analyysin menetelmin, jossa analysoitiin keskeisiä teemoja sekä etsittiin niille mahdollisia alateemoja. Analyysissä tarkastellaan haastattelujen teemojen havaintoja myös eri roolien näkökulmasta.

4.1 Kohdeyrityksen kuvaus

Kohdeyrityksenä työssä oli teknologiayritys, joka keskittyy tarjoamaan räätälöityjä digitaalisia ratkaisuja eri toimialoille. Kohdeyrityksen liiketoimintayksiköllä on oma low-code alusta, joka mahdollistaa nopean ja joustavan sovelluskehityksen. Yrityksen tarjoama alusta sisältää laajan valikoiman työkaluja ja ominaisuuksia, kuten käyttäjäystävällisen käyttöliittymän, liiketoimintaprosessien mallinnuksen, integroidut analytiikkatyökalut sekä mahdollisuuden integroitua eri järjestelmiin. Alustan avulla yritykset voivat luoda ja hallita monimutkaisia liiketoimintasovelluksia nopeasti ja kustannustehokkaasti.

Yrityksen asiakaskuntaan kuuluu laajasti eri toimialojen yrityksiä ja organisaatioita, mukaan lukien julkisen sektorin toimijoita. Julkisen sektorin asiakkaat hyödyntävät yrityksen tarjoamaa alustaa tehostaakseen ja automatisoidakseen hallinnollisia prosesseja, mikä parantaa palveluiden laatua ja tehokkuutta sekä vähentää kustannuksia ja manuaalista työtä.

Yritys mukauttaa ratkaisujaan asiakkaiden erityistarpeisiin, mikä on tehnyt siitä luotettavan kumppanin monille organisaatioille, jotka etsivät innovatiivisia ja kustannustehokkaita ratkaisuja digitalisaation haasteisiin.

4.2 Analyysi ja havainnot low-code alustojen käytöstä ja haasteista

Kerätty haastatteluaineisto analysoitiin teemoittelun menetelmällä, jossa haastatteluissa esiin nousseita piirteitä verrattiin keskenään ja luokiteltiin teemoihin ja alateemoihin. Haastatteluista nousi esiin sekä hyötyjä että haasteita low-code alustojen käytössä ja vastaukset erosivat osittain riippuen siitä, toimivatko vastaajat toteuttavassa vai johtavassa roolissa.

Haastatteluiden pohjalta muodostuivat seuraavat pääteemat ja alateemat, jotka on esitelty Taulukko 5:ssa.

Taulukko 5 Kerätyn aineiston teemat ja alateemat

TEEMA	ALATEEMAT
Monimutkaiset prosessiautomaatiot	Prosessien kompleksisuus, poikkeustapaukset
Tuotteen soveltuvuus ja puutteet	Ominaisuuksien rajallisuus, räätälöinnin tarve
Suorituskyky ja skaalautuvuus	Suorituskykyongelmat, skaalaus isommille prosesseille
Kansalaiskehityksen haasteet ja hyödyt	Käyttäjien osaaminen, yksinkertaisuuden tarve, laatuongelmat
Turvallisuus ja tiedonsiirto	Käyttöoikeuksien hallinta, tietoturva, tiedonsiirron haasteet

4.2.1 Monimutkaiset prosessiautomaatiot

Monimutkaisten prosessien automatisointi low-code alustojen avulla osoittautui haastavaksi erityisesti silloin, kun prosessit sisältävät useita poikkeustapauksia tai vaativat integraatiota muihin järjestelmiin. Haastateltavat toivat esiin, että vaikka alustoilla on mahdollista automatisoida monimutkaisia prosesseja, tämä voi edellyttää huomattavaa suunnittelua ja testausta. Tämä tuo esiin low-code alustan rajoitteet monimutkaisissa automatisointiprosesseissa.

Prosessien poikkeustapausten ja niiden hallinta nousi asiantuntijahaastatteluissa erityisesti toteuttavaa työtä tekevillä haastateltavilla yhtenä haasteena liittyen monimutkaisiin prosessiautomaatiototeutuksiin. Haastateltavat kuvasivat poikkeustapausten tekevän prosessista helposti monimutkaisen sekä vievän enemmän aikaa suunnitteluun sekä prosessin toteutukseen.

> *"Ylipäätään erilaiset poikkeustapaukset ja niiden hallinta monimutkaistaa. Usein tekee ”happy casen” mutta sen jälkeen työ vasta alkaakin, kun alkaa*

> *miettimään poikkeuksia. Se tekee prosessista heti monimutkaisemman ja vaatii enemmän suunnittelua." (Haastateltava 3, toteuttava)*

Toteuttavaa työtä tekevät haastateltavat myös korostivat, että kompleksisuus oli yksi osa-alue, joka monimutkaisti prosessiautomaatioita. Prosessin kompleksisuutta nosti esimerkiksi erilaiset integraatiot tai ylläpidon haasteet kokonaisuuden kannalta. Pitkään käytössä oleviin prosesseihin tehdyt muutokset voivat luoda haasteita, kun kokonaisuus ei enää toimi johdonmukaisesti. Jatkuvat pienetkin muutokset prosessiin voivat tehdä siitä monimutkaisen, mikä vaatii sen perusteellista läpikäyntiä ja mahdollisesti prosessin uudelleenrakentamista.

> *”On sellaisia haasteita, että joissain tilanteissa prosessi on ollut pitkään käytössä ja siihen on tehty koko ajan pikkuhiljaa muutoksia lisää ja lisää, niin siitä on tullut sellainen, että se ei enää se kokonaisuus toimi. Siellä ei ehkä ole huomioitu jotain jossain toisessa prosessivaiheessa olevia muuttujan asettamia tai muuttujia. Tällaisissa prosesseissa on mahdollisesti paljonkin dataa sillä vanhalla prosessilla ja sitä ei voida vaan kerralla räjäyttää kokonaan ja tehdä uudestaan vaan ne täytyy pystyä ne vanhatkin asiat käsittelemään edelleen.” (Haastateltava 4, toteuttava)*

Haastateltavat kertoivat, että he ovat kompleksisuuden haasteita ratkoneet tekemällä erillisitä työtä myös ylläpidossa olleiden prosessien yksinkertaistamiseen. Tässä haastateltavat korostivat sitä, että prosessimallinnuksen työkalut eivät yksin riitä tekemään prosessista selkeää vaan kehittäjällä on suuri vastuu työn siisteydestä.

> *”Itse viime projektissa juuri tein ison homman, kun yksinkertaistin prosessia kun oli muuttunut ”prosesiviidakoksi”. Oli paljon muun muassa päällekkäisiä viivoja kaikkiin suuntiin. Kuvaa katsomalla oli todella vaikea katsoa mitä prosessissa tapahtuu. Työkalu ei itsessään tee siistiä jälkeä eli täytyy myös tekijän pitää siitä huoli, että prosessi on siisti ja luettavissa.” (Haastateltava 3, toteuttava)*

Monimutkaisia prosessiautomaatioita haastatteluissa käsiteltäessä johtavassa asemassa työskentelevät haastateltavat painottivat tuotteistuksen näkökulmaa vahvasti. Haastateltavat korostivat, että monimutkaistakin prosessia tulisi katsoa geneerisestä näkökulmasta, jotta toteutus voidaan myös tuotteistaa. Tavoitetila on, että monimutkainen toteutus on mahdollista

myydä myös muille asiakkaille, mikä tekee niiden toteutuksesta liiketoiminnan kannalta kannattavampaa.

> *”Kun on yhdelle asiakkaalle tehty monimutkaisempi prosessi niin pyrimme tunnistamaan sen geneerinen prosessi ja toisaalta etsimään kohdeasiakkuuksia, joille voimme myydä tätä samaa ratkaisua, jonka olemme jo tehneet kertaalleen. Silloin se tarkoittaa sitä, kun jos meillä on tallainen monimutkainen toteutettu prosessi niin meidän kannattaa aina tuotteistuksen näkökulmasta miettiä, että kenelle muille me voitaisiin tätä kertaalleen toteutettua prosessia lähteä myymään.” (Haastateltava 2, johtava)*

Johtavan aseman haastateltavat korostivat kuitenkin myös toteutusprojekteissa kuluvaa aikaa monimutkaisissa prosessiautomaatiototeutuksissa. Kun kyseessä on monimutkaisempi prosessi, joka tulee automatisoida, myös prosessin kehitys vie enemmän aikaa. Usean toteuttajan samanaikainen työskentely voi olla vaikeaa monimutkaisten prosessien kohdalla. Kun prosessi on monimutkainen, ratkaisussa on tärkeää miettiä kokonaisuutta toteutuksessa ja usean henkilön on tätä vaikeaa helposti yhtäaikaisesti tehdä.

> *”Monimutkaisissa prosessien toteutusprojekteissa kalenteriaikaa tarvitsee varata riittävästi, koska prosessin kehitys ottaa oman aikansa. Monimutkaisia prosessien tekemistä on vaikeaa myös rinnakkaistaa koska kaikki vaikuttaa kaikkeen. Jotta kokonaisuus pysyy hallinnassa voi työtä tehdä ainoastaan yks henkilö kerrallaan.” (Haastateltava 5, johtava)*

4.2.2 Tuotteen soveltuvuus ja puutteet

Tuotteen soveltuvuuden ja puutteiden yhteydessä haastateltavat korostivat, että low-code alustojen ominaisuuksien rajallisuus ja räätälöinnin tarve vaikuttavat niiden käytettävyyteen ja tehokkuuteen.

Ominaisuuksien rajallisuus esiintyi toteuttavien henkilöiden haastatteluissa tarpeena kehittää uusia ominaisuuksia tai kehittää edelleen vanhoja toiminnallisuuksia asiakkaan tarpeiden mukaan. Low-code alustoilla on usein rajoitetut ominaisuudet, joten osa asiakkaiden toiveista saattaa vaatia koodimuutoksia ja lisäkehitystä. Tämä voi viivästyttää kehitysprosessia ja kasvattaa projektin kustannuksia.

> ”*Olen kehittänyt myös täysin uusia ominaisuuksia mutta usein jotain vanhaa ominaisuutta kehitetään lisää. Kun tarvitaan koodimuutoksia, niin silloin toki aina prosessin kehitys seisahtuu siksi aikaa, kun kehittäjät luovat uuden muutoksen. Myös asiakkaan kannalta tallaiset tapaukset ovat kalliimpia sekä vie enemmän aikaa että ominaisuus tai muutos saadaan heille käyttöön.” (Haastateltava 1, toteuttava)*

Sekä johtavassa että toteuttavassa asemassa työskentelevät haastateltavat korostivat myös räätälöinnin tarvetta. Vaikka asiakkaalta tuleva tarve on usein spesifi ja uusi ominaisuus tulee määritellä tarkasti tärkeää on, että uudet ominaisuudet suunnitellaan uudelleenkäytettäviksi eri asiakkaille. Yksittäisen todella räätälöidyn ratkaisun sijaan parempi vaihtoehto on luoda geneerisiä ja asiakaskohtaisesti konfiguroitavia ratkaisuja.

> *”Määrittelyvaiheessa on myös tärkeää tehdä uudesta ominaisuudesta uudelleenkäytettävä eli ei vain jotain tosi spesifiä tietylle asiakkaalle. Tärkeää on myös, että se määrittely on tehty tarpeeksi tarkalla tasolla, jotta ei sitä sitten tarvitse korjata ja tehdä useaan kertaan. Eli kaikki sellaiset asiat mitä voisi ajatella itsestäänselvyyksinä mutta ei saa kuitenkaan tehdä oletuksia.” (Haastateltava 4, toteuttava)*

Johtavassa asemassa työskentelevät haastateltavat painottivat myös strategista näkemystä vastauksissaan. Yksittäisiä ominaisuuksia kehittäessä tärkeää on arvioida, tukeeko kyseinen ratkaisu myös yrityksen strategisia tavoitteita. Uusien investointien kohdalla arvioidaan ovatko kehitykseen käytettävät resurssit ja aika perusteltuja pitkällä aikavälillä. Pitkän aikavälin hyödyt ovat keskeisiä, jotta investoinnit tuottavat lisäarvoa sekä tukevat yrityksen kilpailukykyä.

> *”Usein kehitystarpeet lähtee yksittäisistä asiakastarpeista. Näitä tarpeita arvioidaan kuitenkin laajemmassa kontekstissa, jotta varmistetaan, että ratkaisut tukevat kokonaisstrategiaamme. Näin voidaan varmistaa, että investoinnit uusiin ominaisuuksiin on perusteltuja ja tuovat pitkän aikavälin hyötyjä.” (Haastateltava 5, johtava)*

Kilpailukyvyn parantamista haastateltavien mukaan tehtiin myös jatkuvalla tuotekehityksellä, joka mahdollistaa yrityksen erottumisen kilpailevista tuotteista markkinoilla. Uusien

ominaisuuksien kehittäminen voi olla kuitenkin haasteellista ja mukana on riskejä sillä kaikki ideat eivät ole automaattisesti menestyksekkäitä tai markkinoilla toimivia.

> *”Koko ajan tehdään myöskin sellaista sisäistä tuotekehitystä, joka pyrkii löytämään sellaisia ominaisuuksia, joilla saavutettaisiin selkeä kilpailuetu alalla oleviin kilpaileviin tuotteisiin verrattuna. Sinänsä tuon tyyppinen tuotekehityksen ideointi ei ole helppoa ja siihen liittyy aina myös riskejä.” (Haastateltava 2, johtava)*

4.2.3 Suorituskyky ja skaalautuvuus

Suorituskyky ja skaalautuvuus ovat tärkeitä teemoja erityisesti suurten ja vaativien prosessien yhteydessä. Asiantuntijahaastattelut toivat esiin, että monimutkaiset prosessit, joissa tapahtuu paljon taustalla olevia toimintoja tai joissa on laajoja käyttäjämääriä voivat kärsiä suorituskykyongelmista. Tämä vaikuttaa suoraan asiakaskokemukseen ja liiketoiminnan tehokkuuteen.

Suorituskykyongelmia erityisesti toteuttavat haastateltavat kuvasivat ilmenevän, kun järjestelmässä on monimutkaisia prosesseja, paljon käyttäjiä tai suuria datamääriä. Prosessien suorituskyky on ratkaiseva tekijä myös asiakaskokemuksen kannalta. Mikäli prosessin eri toimenpiteissä kestää kauan, asiakkaiden tyytymättömyys kasvaa ja tämä voi vaikuttaa negatiivisesti käyttökokemukseen. Suorituskykyongelmat voivat johtua myös siitä, että prosessi on turhan monimutkainen tai se sisältää tarpeettomia toimintoja, joita voisi optimoida. Toteuttajat korostivatkin käytännön ratkaisuja suorituskykyongelmien ratkaisemiseksi eli prosessien optimoimisen.

> *” Kun on monimutkaisempi prosessi ja tapahtuu esimerkiksi paljon konvertointeja sekä on paljon käyttäjiä niin suorituskykykin voi hidastua. Erityisesti jos on paljon taustalla tapahtuvia prosessitoimenpiteitä niin asiakkailta on välillä tullut kommenttia, että näiden pyörähtämiseen menee liikaa aikaa. Näissä tapauksissa ollaan katottu onko kaikki toimenpiteet oikeasti tarpeellisia ja voisiko prosessia optimoida eri parametrein.” (Haastateltava 1, toteuttava)*

Skaalautuvuuden hallinta on myös keskeinen haaste suurten prosessien kohdalla. Toteuttajien haastatteluissa koettiin ongelmalliseksi skaalautuvuuden testaaminen ja varmistaminen,

että järjestelmä toimii hyvin myös suuremmalla kuormituksella ja datamäärillä. Ratkaisuna näihin ongelmiin tuotiin esille se, että testiautomaation vielä laajempi hyödyntäminen voisi helpottaa skaalautuvuuden testaamista. Tämä oli toteuttajien näkökulmasta kriittistä, sillä ilman riittäviä testejä järjestelmä ei välttämättä pystytä vastaamaan käyttäjien tarpeisiin tuotantoympäristössä.

> *”Testaus on hankalaa oikean käyttötapauksen, oikean datamäärän ja tiedostokoon mukaisesti. Se ei aina ole ihan itsestään selvää, miten pystyy skaalautuvuutta tai kuormitusta testaamaan. Olisi hyvä olla käytössä vielä parempia työkaluja esimerkiksi testiautomaation hyödyntäminen vielä lisää. Eli näin saataisiin tuotannon kaltaisuutta ja poikkeuksia testattua paremmin” (Haastateltava 3, toteuttava)*

Johtavan aseman haastateltavat korostivat, että mahdollisiin suorituskykyongelmiin ei voida aina vastata lisäämällä teknisiä resursseja, kuten palvelimia. Sen sijaa suorituskyvyn parantaminen vaatii huomiointia jo suunnitteluvaiheessa sekä alustan teknisen arkkitehtuurin valinnoissa.

> *”Kaikkia suorituskykyongelmia ei voida ratkaista pelkästään lisäämällä resursseja, kuten palvelimia. On tärkeää, että kehitystiimi ottaa nämä näkökohdat huomioon jo suunnitteluvaiheessa ja varaa riittävästi aikaa ja resursseja niiden ratkaisemiseen projektin aikana. Tämä auttaa varmistamaan, että projekti pysyy aikataulussa ja budjetissa.” (Haastateltava 5, johtava)*

4.2.4 Kansalaiskehityksen haasteet ja hyödyt

Kansalaiskehityksen osalta asiantuntijahaastatteluissa korostuivat osaamisen taso, yksinkertaisuus sekä tuki ja dokumentaatio.

Yksi kansalaiskehityksen suurista haasteista on käyttäjien tekninen osaaminen, jonka sekä toteuttavassa että johtavassa asemassa työskentelevät haastateltavat mainitsivat. Usein kansalaiskehittäjillä ei ole tarvittavaa teknistä näkemystä, mikä voi johtaa huonoihin tai tehottomiin ratkaisuihin. Kansalaiskehityksessä käyttäjien rajallinen tekninen osaaminen voi johtaa puutteellisiin ratkaisuihin, vaikka ne toimisivatkin periaatteessa.

> *”Haasteita on usein se, ettei asiakas täysin ymmärrä mitä on tekemässä ja tekee sellaisia ratkaisuja, jotka vaan toimii. Itse kuitenkin koittaa miettiä toimivuuden lisäksi onko ratkaisu hyvä tai voisiko se olla paremmin tehty.” (Haastateltava 1, toteuttava)*

Mahdollisia laatuongelmia haastateltavat ratkaisivat muun muassa kattavalla ratkaisujen testaamisella. Ratkaisujen testaaminen vastuutetaan projektikohtaisesti, mutta tärkeää on, että ne testataan alemmissa ympäristöissä ennen tuotantoympäristöön siirtoa. Virheiden ja ongelmien sattuessa tärkeää on myös, että aikaisempi versio saadaan helposti palautettua käyttöön. Näin ongelmatilanteen kesto pysyy mahdollisimman lyhyenä ja tuotantoympäristöön ei aiheudu käyttökatkoja.

> *”Tärkeintä on, että asiakkaan ratkaisut olisi syytä testata hyvin. Toki meillä myös saadaan edelliset versiot aina talteen. Asiakkaat joko itse testaa omat tekemänsä ratkaisut tai me testaamme heidän puolestansa. Alustassa on myös lokitustietoa kyllä saatavilla, eli jos asiakas kirjautuu omilla tunnuksillaan, niin sieltä näkyy kyllä, kuka on tehnyt muutoksia ja tallentanut uusia prosessiversioita.” (Haastateltava 4, toteuttava)*

Haastatteluissa myös painotettiin, että kansalaiskehityksessä on tärkeää ratkaisujen yksinkertaisuus ja helposti ymmärrettävyys. Työkalujen käytettävyyden tulee olla korkealla tasolla, jotta käyttäjät pystyvät toteuttamaan ratkaisuja ilman suuria virheitä. Myös käyttöoikeuksien rajoittaminen koettiin hyväksi toimintatavaksi estämään, ettei liian suuria muutoksia kansalaiskehittäjät edes pysty tekemään.

> *”Vaatii kyllä työkalulta ja toteutukselta sitä, että ratkaisut ovat mahdollisimman yksinkertaisia ja yksiselitteisiä. Pitää pystyä tekemään mahdollisimman paljon yksinkertaista ja käytettävyys mahdollisimman hyvä. Pitää myös pystyä varmistumaan siitä, ettei asiakkaalle ole liikaa oikeuksia, että pystyy mitään hajottamaan.” (Haastateltava 3, toteuttava)*

Onnistuneen kehitystyön kannalta sekä toteuttajat että johtajat painottivat kansalaiskehittäjille järjestettävää tukea sekä koulutusta. Koulutuksissa on tärkeää tarjota ”hands-on” -harjoituksia sekä tukea dokumentaation kautta. Kattava tuki ja koulutus ovat keskeisiä onnistumisen kannalta, koska kansalaiskehittäjillä on usein rajoittunut tekninen osaaminen.

> *”Pidämme usein koulutuspakettiin kuuluvan hands on -session kansalaiskehittäjille. Näissä käydään läpi perusteita. Kaikki perustuu tietysti kattavaan dokumentaatioon ja tukeen.” (Haastateltava 2, johtava)*

Vaikka kansalaiskehityksessä on haasteita, haastatteluissa nousi eri rooleilla myös merkittäviä hyötyjä. Kansalaiskehittäjät tuntevat usein parhaiten omat liiketoimintaprosessinsa, mikä mahdollistaa sen, että he voivat tuottaa ratkaisuja suoraan vastaamaan heidän vaatimuksiansa ja odotuksia. Kansalaiskehitys myös voi vähentää väärinymmärrysten riskiä sekä lisätä asiakastyytyväisyyttä.

> *”Asiakkaalla on usein paras tietämys minkälaisen lopputuloksen haluaa. Eli esimerkiksi kun ratkaisuja määritellään ei asiakas aina itse osaa sanoittaa mitä haluaa. Itse kun asiakas toteuttaa niin näkee sitten samalla, onko se toteutus se mitä haluaa vai onko tahtotila sittenkin jokin muu.” (Haastateltava 1, toteuttava)*

Kansalaiskehittäjien onnistuminen low-code työkaluilla myös sitouttaa heitä vahvemmin käytettyyn teknologiaan ja organisaation kehitysprojekteihin. Tämä näkyy erityisesti silloin, kun kansalaiskehittäjät saavat aikaan toimivia sekä hyödyllisiä ratkaisuja.

> *”Kun kansalaiskehittäjät onnistuvat niin se sitouttaa heitä kyseiseen ratkaisuun vahvasti mukaan.” (Haastateltava 2, johtava)*

Johtavassa roolissa toimivat haastateltavat toivat esiin kuitenkin sen, että kansalaiskehityksen merkitys liiketoimintastrategiassa ole vielä riittävän suuri, että siihen panostettaisiin merkittäväksi. Kohdeyrityksen strateginen painopiste on enemmän omien prosessien tehostamisessa ja low-code työkalujen hyödyntämisessä projektien nopeuttamiseksi ja kustannustehokkuuden parantamiseksi.

> *”Kansalaiskehittäjien panoksen merkitys yrityksen liiketoimintastrategiassa ei tällä hetkellä ole riittävän suuri, jotta siihen kannattaisi panostaa merkittävästi. Keskitymme tehostamaan omaa toimintaamme low-code työkaluilla, jotta voimme toimittaa projekteja nopeasti ja kustannustehokkaasti. Tämä strategia auttaa meitä saamaan uusia asiakkaita ja kehittämään liiketoimintaamme.” (Haastateltava 5, johtava)*

4.2.5 Turvallisuus ja tiedonsiirto

Turvallisuus ja tiedonsiirto ovat keskeisiä teemoja erityisesti low-code alustoilla, jotka mahdollistavat erityyppisten käyttäjien tekemät muutokset ja konfiguroinnit. Haastatteluissa nousi esille, että tietoturva ja käyttöoikeuksien hallinta ovat keskeisiä huolenaiheita molemmissa rooleissa niin toteuttavassa kuin johtavassa.

Low-code alustojen käytössä tietoturva on keskeinen huolenaihe, erityisesti kun kansalaiskehittäjät tekevät muutoksia ja konfigurointeja itse. Toteuttavassa roolissa työskentelevät haastateltavat korostivat, että oikein suunnitellut käyttöoikeudet ja prosessien dokumentointi ovat olennaisia turvallisuuden varmistamisessa. Tärkeää on seurata ja lokittaa kaikki järjestelmässä tapahtuvat toiminnot. Käyttäjien henkilökohtaisten tunnusten käyttö mahdollistaa sen, että jokainen toimenpide sekä muutos voidaan jäljittää tiettyyn käyttäjään. Tämä parantaa tietoturvan tasoa ja vähentää väärinkäytösten riskiä.

> *"Meillä toki lokitetaan järjestelmää hyvin ja esim. tuotannossa kaikilla käyttäjillä on aina henkilökohtaiset tunnukset eli mitään yhteiskäyttötunnuksia ei koskaan käytetä." (Haastateltava 1, toteuttava)*

Haastateltavat nostivat esille myös sen, että tietoturvariskejä on tärkeää käsitellä koulutuksissa, jotta kansalaiskehittäjät ymmärtävät miten heidän muutoksensa voivat vaikuttaa turvallisuuteen. Tietoturvakysymykset eivät ole pelkästään teknisten tiimien vastuulla, vaan kaikkien kehittäjien on oltavat tietoisia toimistaan ja niiden vaikutuksista turvallisuuteen.

> *"Tällaiset asiat (tietoturva) on tärkeää tuoda esiin asiakaskoulutuksessa. Asiakkaiden täytyy ymmärtää, mitä tietoja he käyttävät muuttujissa ja miten ne voivat vaikuttaa tietoturvaan." (Haastateltava 4, toteuttava)*

Toteuttavan roolin haastateltavat nostivat esiin myös tiedonsiirtoon liittyviä haasteita, jotka voivat ilmetä erityisesti suurten tietomäärien tai monimutkaisten integraatioiden yhteydessä. Tietoturvariskit voivat myös kasvaa, jos siirrettävä dataa sisältää henkilökohtaisia tai arkaluontoisia tietoja. Tiedonsiirron hallinta ja suojaaminen välttämätöntä turvallisuuden takaamiseksi.

> *"Jos ajatellaan esimerkiksi näitä erilaisia integraatioita, niin niissä täytyy myös aina huomioida myös tietoturvan näkökulma. Eli minkälaista dataa*

> *liikkuu ja miten sitä voidaan siirtää tietoturvallisesti minkäkin rajapinnan kautta. Se, että esimerkiksi jotakin tehdään jotain SFTP siirtoa jostakin materiaalista, joka sisältäisi selkokielisenä henkilötunnuksia ja tällaisia, niin se ei ole enää nykypäivänä kovin hyvä ratkaisu. Täytyy sitten miettiä, että millaisella ratkaisulla ja miten niitä voidaan sitten tarvittaessa, jos sellaista tarvetta tulee." (Haastateltava 4, toteuttava)*

Johtavien roolien näkökulmasta tietoturva on myös strateginen kysymys. Kohdeyrityksellä on käytössä sertifikaatteja, kuten ISO9001 ja ISO2700, jotka osoittavat, että organisaatio noudattaa tietoturvan ja laadunhallinnan parhaita käytäntöjä. Sertifikaatit antavat asiakkaille varmuuden siitä, että tietoturva on suunniteltu ja toteutettu huolellisesti toimittajaorganisaatiossa. Tämä strateginen lähestymistapa tietoturvaan auttaa yrityksiä varmistamaan, että ne toimivat standardien mukaisesti ja täyttävät sekä omat että asiakkaidensa vaatimukset.

> *"Jos puhutaan ohjelmistokehityksen ja prosessien tietoturvallisuudesta niin siinä avainasemassa on sertifikaatit mitä toimittajaorganisaatiolla voi olla. Ja silloin puhutaan esim. ISO9001 tai ISO27001 sertifikaateista (laatusertifikaatti ja tietoturvasertifikaatti). Kun organisaatiolla on nämä sertifikaatit, se osoittaa, että näitä asioita on mietitty ja se toimintatapa on silloin sen tyyppinen, että se noudattaa hyviä käytäntöjä ja dokumentointi on kunnossa." (Haastateltava 2, johtava)*

4.3 Haastatteluista saadut tulokset

Tässä osiossa esitetään haastatteluista saadut keskeiset tulokset tutkimuskysymysten valossa yhteenvetona. Haastatteluista saadut tulokset ovat jaettu kolmeen osioon: hyödyt, haasteet ja parhaat käytännöt.

4.3.1 Hyödyt

Low-code alustojen keskeisimmät hyödyt asiantuntijahaastatteluissa liittyvät visuaalisen suunnittelun ja käyttäjälähtöisyyteen, prosessien ymmärrettävyyteen ja hallintaan, nopeuteen ja kustannustehokkuuteen sekä prosessien parantamiseen.

Yksi low-code alustojen hyödyllisistä ominaisuuksista on niiden tarjoama visuaalinen suunnitteluympäristö. Erityisesti toteuttavan roolin haastateltavat korostivat, että visuaalinen suunnittelu mahdollistaa monimutkaisten prosessien ymmärtämisen ja kehittämisen helpommin erityisesti niille käyttäjille, joilla ei ole syvällistä teknistä osaamista. Visuaalinen suunnittelu tuo prosessit selkeämmin käyttäjille näkyviin ja tekee niistä helpommin hallittavia. Tämä vähentää virheiden riskiä ja parantaa yhteistyötä eri tiimien välillä, mikä nopeuttaa kehitysprosesseja.

> *”Graafinen käyttöliittymä tekee prosessien hallinnasta ja kehittämisestä intuitiivista ja helposti ymmärrettävää. Asiakkaat voivat helposti seurata prosessien kulkua ja nähdä, kuka käsittelee mitäkin vaihetta.” (Haastateltava 4, toteuttava)*

Johtajat puolestaan korostivat, että graafinen käyttöliittymä auttaa koko tiimiä sitoutumaan prosessien kehittämiseen ja tarkistamaan, että prosessit toimivat oikein. Graafinen käyttöliittymä parantaa prosessien käytettävyyttä ja mahdollistaa loppukäyttäjien osallistumisen kehitykseen, mikä lisää ratkaisujen hyväksyttävyyttä ja tehokkuutta.

> *”*T*ämäntyyppinen lähestymistapa missä sinä lähtökohtaisesti teet sen tyyppisiä graafisia kuvauksia, jotka ovat loogisia niin ne auttavat helpommin sitouttamaan enemmän porukkaa toteamaan, että näinkö me halutaan, että tämä prosessi toimii” (Haastateltava 2, johtava)*

Keskeiset low-code alustojen hyödyt haastatteluissa liittyivät myös nopeuteen ja kustannustehokkuuteen sekä johtavan että toteuttavan roolin henkilöillä. Low-code alustat vähentävät raskaan koodaustyön tarvetta ja mahdollistavat resurssien tehokkaan uudelleenkäytön. Tämä johtaa merkittäviin kustannussäästöihin erityisesti silloin, kun useat asiakkaat hyötyvät samasta ominaisuudesta.

> *”Suuria etuja ovat ainakin alustan nopeus ja toistettavuus. Jos kaikki pitäisi koodata alusta lähtien, se olisi huomattavasti hitaampaa. Meillä on käytössä prosessimoottori, joka muuntaa datan käyttökelpoiseksi. Tämä nopeuttaa prosessien ja muutosten toteuttamista verrattuna siihen, että kaikki koodattaisiin käsin.” (Haastateltava 4, toteuttava)*

Nopeuden merkitys korostuu myös asiakasprojekteissa, joissa muutokset ja uudet ominaisuudet voidaan ottaa nopeasti käyttöön ilman merkittäviä käyttökatkoja. Low-code

työkalujen avulla prosessien toteuttaminen nopeutuu ja iteratiivinen kehitys toimii nopeammin, mikä voi tuoda kilpailuetua sekä parantaa asiakastyytyväisyyttä.

> *”Low-code ominaisuuksien ansiosta voimme tehdä muutoksia tuotantoympäristöön nopeasti ja ilman katkoja. Tämä on erittäin tärkeää asiakkaille, sillä muutokset voidaan toteuttaa ilman, että ne näkyvät tuotantokatkoina tai vaativat asennuksia. Uuden prosessin version voi vain tiputtaa paikalleen.” (Haastateltava 5, johtava)*

Johtavassa asemassa työskentelevät haastateltavat nostivat hyötynä esiin prosessien parantamisen ja hallittavuuden. Prosessien automaatio low-code alustoilla parantaa organisaation hallittavuutta ja vähentää henkilöriskejä. Haastateltavien mukaan, kun prosessit ovat automatisoituja ja roolit selkeästi jaettuja, yrityksen toiminta on hallitumpaa ja virheiden määrä vähenee. Automatisoidut prosessit vähentävät manuaalisen työn tarvetta, mikä vähentää myös riskiä siitä, että työntekijän poissaolo vaikuttaisi kriittisiin liiketoimintaprosesseihin. Tämä parantaa prosessien jatkuvuutta sekä liiketoiminnan tehokkuutta.

> *”Se (liiketoimintaprosessien automatisointi) parantaa hallittavuutta. Kun tiedämme missä asiat menevät ja kenen vastuulla ne ovat, mikään asia ei niin sanotusti putoa väliin. Kun meillä on eri rooleja ja tehtäviä, niin silloin pystytään aina tarkistamaan mikä kyseisen casen tila on. Toisaalta tietysti, kun automatisoidaan prosesseja ja varsinkin sellaisia joissa on manuaalisia vaiheita niin sillä vähennetään aina henkilöriskiä joka voi konkretisoitua kun henkilö esimerkiksi lähtee pois yrityksestä. Näin asiat eivät ole yhden henkilön varassa.” (Haastateltava 2, johtava)*

Johtajien näkökulmasta low-code teknologian hyödyntäminen mahdollistaa prosessien kriittisen tarkastelun ja parantamisen. Low-code työkalujen avulla voidaan varmistaa, että prosessit toimivat oikein ja tehokkaasti, mikä tuo kustannus- ja aikasäästöjä. Haastateltavat korostivat, että low-code alustojen tuoma lisäarvo ei rajoitu pelkästään olemassa olevien prosessien nopeuttamiseen, vaan se mahdollistaa myös prosessien kehittämisen ja uusien toimintatapojen omaksumisen.

> *”Nämä (low-code) työvälineet mahdollistavat myöskin sen prosessin kehittämisen ja sitä pitäisikin aina hyvinkin kriittisesti tehdä määrittelyvaiheessa. Eli tulisi miettiä tehdäänkö asioita oikein. Teknologia kuitenkin mahdollistaa*

täysin uusia ratkaisuja joiden avulla voidaan esimerkiksi läpimenoaikaa, tehokkuutta ja kustannussäätöjä saada paljonkin kun vaan mietitään pyöriikö asia oikeassa järjestyksessä ja tekeekö oikeat ihmiset oikeita asioita oikeaan aikaan. Eli tavallaan prosessien kehittäminen samalla kun automatisoidaan on keskeinen tavoite myöskin." (Haastateltava 2, johtava)

4.3.2 Haasteet

Vaikka low-code alustat tarjoavat merkittäviä etuja, haastatteluissa nousi esiin useita haasteita. Haasteet liittyvät monimutkaisten prosessien hallintaan, teknisiin rajoituksiin, suorituskykyyn ja skaalautuvuuteen sekä muutosvastarintaan. Nämä haasteet liittyvät usein siihen, että low-code alustat eivät pysty täysin tukemaan kaikkia prosessin monimutkaisuuksia ja vaativat usein teknistä osaamista sekä resursseja.

Monimutkaiset liiketoimintaprosessit ja niiden integrointi low-code alustoille osoittautuivat merkittäväksi haasteeksi, joka vaativat huomattavaa teknistä asiantuntemusta ja resursseja. Toteuttavassa roolissa työskentelevät kertoivat, että eritysesti monimutkaiset prosessitoteutukset, kuten integraatiot sekä poikkeuksia sisältävät prosessit ovat olleet vaikeimpia toteuttaa. Monimutkaisuus kasvaa, kun low-code alustoja yhdistetään muihin järjestelmiin. Tämä vaatii usein lisäresursseja sekä teknistä osaamista, mikä voi hidastaa projektin etenemistä. Integraatiot voivat edellyttää räätälöityjen ratkaisujen kehittämistä, mikä monimutkaistaa prosessia sekä kehitystä.

"Monimutkaisimmat prosessitoteutukset ovat olleet integraatiot prosesseissa sekä monimutkaiset allekirjoitusprosessit." (Haastateltava 1, toteuttava)

Sekä toteuttavassa että johtavassa työskentelevät haastateltavat korostivat, että low-code alustoihin liittyy teknisiä rajoituksia. Vaikka low-code alustat vähentävät ohjelmoinnin tarvetta, ne eivät poista sitä kokonaan. Tekninen ymmärrys on välttämätöntä, sillä prosessien toteuttaminen vaatii tietämystä siitä, miten alustaa voidaan käyttää tehokkaasti. Low-code alustat pyrkivät yksinkertaistamaan kehitystä, mutta ne eivät poista täysin teknisen osaamisen tarvetta. Kehittäjien on ymmärrettävä alustan tekniset konseptit sekä ymmärrettävä miten alusta toimii, jotta prosessitoteutukset saadaan toimimaan oikein.

> *”Low-code alustoilla on aika paljon rajoituksia. Eli tulee paljon vastaan tilanteita, että tiettyjä asioita ei voinutkaan niin suoraviivaisesti tehdä. Eli esimerkiksi kun toteutetaan prosessia niin ei se olekaan niin yksinkertaista, että voi vaan drag and dropata laatikkoja ja prosessi toimisi saman tien. Eli tarvitsee myös olla ymmärrystä mitä tekee ja miten.” (Haastateltava 4, toteuttava)*

Suorituskykyongelmat ja alustan kyky skaalautua laajoihin prosesseihin nousivat esiin erityisesti suurissa ja vaativissa toteutuksissa toteuttavassa roolissa työskentelevillä. Paljon resursseja vaativien prosessien suorituskyky saattoi hidastua, mikä näkyy käyttäjälle hitautena. Suorituskykyongelmat voivat johtaa viiveisiin ja heikentää asiakaskokemusta.

> *”Yleinen hitaus ja tahmeus ovat konkreettisimpia ongelmia mitä huomaa. Usein ne (suorituskyky ja skaalautuvuus haasteet) on sellaisia asioita, joita ei ole osannut huomioida, eli jokin asia kestääkin kauemmin kuin on ajatellut.” (Haastateltava 3, toteuttava)*

Johtavassa asemassa työskentelevät mainitsivat haastatteluissa yhtenä haasteena organisaation sisäisen muutosvastarinnan. Automatisointiprojektit voivat kohdata vastustusta henkilöstön keskuudessa erityisesti niiltä työntekijöiltä, joiden työtehtäviä automatisaatio vähentää. Tämä voi johtaa siihen, että työntekijät kokevat ammattitaitonsa vähenevän. Vastustus voi hidastaa automatisaatioprojektien etenemistä sekä vaatii johtajilta tarkkaa suunnittelua ja viestintää. Työntekijöiden on organisaatiossa tärkeä ymmärtää, että automatisointi ei vähennä heidän osaamistaan, vaan mahdollistaa heidän keskittymisensä strategisempiin ja muihin tehtäviin.

> *”Näillä automatisoinneilla pyritään vähentämään tämän (hiljainen dokumentoimaton tieto) määrää ja roolia. Mikä toisaalta taas aiheuttaa mahdollisesti vastustusta. Eli henkilöt, joiden ammattitaito perustuu siihen että he osaavat tehdä tiettyjä asioita ja automatisoinnin ansiosta voidaan sellaisia tehtäviä poistaa. Henkilöt voivat kokea, että heidän ammattitaitonsa vähenee, kun prosessi ei olekaan enää henkilöriippuvainen.” (Haastateltava 2, johtava)*

Haastatteluissa johtavan näkökulman haastateltavat myös toivat esille, että low-code alustojen hyödyntämisessä on usein mahdollisuuksia, joita ei hyödynnetä vielä täysin. Automatisointia olisi mahdollista laajentaa ja integroida muihin järjestelmään, mikä lisäisi prosessien tehokkuutta ja tuottavuutta. Tämä laajentaminen ei kuitenkaan ole aina mahdollista

rajallisten resurssien ja liiketoimintatavoitteiden vuoksi. Tämä hidastaa low-code alustojen täysimittaista hyödyntämistä erityisesti, kun sisäiset kehityshankkeet kilpailevat muiden korkeammalle priorisoitujen asiakasprojektien kanssa. Tämä johtaa siihen, että automatisaation laajentaminen ja järjestelmien integrointi ei aina etene toivotulla tavalla.

> *”Näen, että automatisointia voisi laajentaa ja integroida muihin järjestelmiin vielä lisää, mikä toisi lisäarvoa. Eri prosesseissamme olisi mahdollisuuksia laajentaa ja parantaa prosesseja, mutta tällä hetkellä toiminta ei ole täysin systemaattista. Kuitenkin tätä loppuun saattamista rajoittavat eniten liiketoiminnan tavoitteet ja resurssit. Koska olemme konsulttitalo, sisäiseen kehitykseen käytettävä aika ja raha ovat rajallisia ja se kilpailee usein muiden hankkeiden kanssa, ja päätökset tehdään sen perusteella, mikä tuo eniten arvoa liiketoiminnalle. Tämä priorisointi voi johtaa siihen, että kaikki prosessikehityshankkeet eivät etene loppuun asti.” (Haastateltava 5, johtava)*

4.3.3 Parhaat käytännöt

Haastatteluista nousi esiin useita parhaita käytäntöjä, jotka ovat osoittautuneet hyödyllisiksi low-code alustojen käytössä. Näitä käytäntöjä voidaan soveltaa laajemmin organisaatioissa, jotka pyrkivät automatisoimaan liiketoimintaprosessejaan. Keskeisiä teemoja esiin nousseissa parhaissa käytännöissä olivat yksinkertaisuuden merkitys, prosessien valinta, käyttäjien koulutus ja ohjeistus sekä ulkoisen kumppanin rooli alustojen käyttöönotossa.

Yksinkertaisuuden korostaminen oli keskeinen teema, jota erityisesti toteuttavassa asemassa työskentelevät haastateltavat korostivat. Haastateltavat painottivat, että prosessien tulisi olla mahdollisimman yksinkertaisia ja selkeitä, jotta ne ovat helposti ymmärrettäviä ja hallittavia. Monimutkaisten ratkaisujen sijaan tulisi pyrkiä yksinkertaiseen kokonaisratkaisuun, joka voidaan jakaa sopiviin osiin.

> *”Korostaisin yksinkertaisuutta. Eli pyrkii mahdollisimman simppeliin ratkaisuun ja suunnitteluun. Prosessikehityskin on prosessi eli etenee jaksoittain. Tekeminen on hyvä jakaa sopiviin osiin, jotta saadaan laadukasta jälkeä.” (Haastateltava 3, toteuttava)*

Yksinkertaisuus parantaa prosessien hallittavuutta ja mahdollistaa nopeamman ja tehokkaamman kehityksen. Prosessien mallinnuksessa on haastateltavien mukaan hyödyllistä aloittaa yleiskuvasta ja siirtyä vähitellen yksityiskohtiin, jotta monimutkaisuus ei kasva hallitsemattomaksi. Yksinkertaiset ratkaisut eivät ainoastaan helpota prosessien kehittämistä, vaan mahdollistavat myös asiakkaiden paremman ymmärryksen siitä, miten prosessit toimivat. Tämä parantaa prosessien läpinäkyvyyttä ja jo käyttöönottovaiheessa varmistaa, että asiakkaat voivat seurata kehitystyötä sekä osallistua siihen mukaan.

> *”Tärkeää on kuitenkin, että prosessi pidetään tarpeeksi yksinkertaisena, jotta asiakas pystyy sitä lukemaan ja ymmärtää sitä.” (Haastateltava 4, toteuttava)*

Johtavassa asemassa työskentelevät haastateltavat korostivat, että automatisoitavan prosessin valinta on kriittinen vaihe low-code alustojen käytön aloittamisessa. Parhaat tulokset haastateltavien mukaan saavutetaan, kun valitaan automatisointiin sellaiset prosessit, joissa manuaalivaiheita on mahdollista vähentää ja prosessiin osallistuu useita henkilöitä eri rooleista. Prosessin automatisaatiolla voidaan saavuttaa prosessien keskitetympää hallintaa sekä koordinointia.

> *”Kannattaa aloittaa sellaisista prosesseista, joissa hyötynäkökulma on huomioitu. Eli jos on sellaisia prosesseja, joissa on manuaalivaiheita ja koet että homma tökkii koko ajan. Joko niin, että asiat kestää ja asioita tippuu väliin ja asiakkaat valittaa kun homma ei pelitä. Ja varsinkin jos prosessiin osallistuu organisaatiosta eri henkilöitä eri rooleilla sekä prosessia tulisi saada keskitetysti koordinoitua koko prosessia.” (Haastateltava 2, johtava)*

Haastatteluissa myös painotettiin, että on tärkeää dokumentoida ja ymmärtää omat liiketoimintaprosessit ennen low-code alustalla tehtävää prosessiautomaatiota, jotta voidaan priorisoida oikeat prosessit sekä aloittaa helpoimmasta. Tämä auttaa varmistamaan, että oppimiskäyrä ei ole liian jyrkkä ja alustan käyttö alkaa sujuvasti.

> *”Ennen alustan käyttöönottoa, dokumentoi ja ymmärrä omat liiketoimintaprosessisi. Eli mitä prosesseja haluat automatisoida ja miten ne toimivat tällä hetkellä. Tämä auttaa priorisoimaan ja aloittamaan helpoimmista prosesseista, jotta oppimiskäyrä ei ole liian jyrkkä.” (Haastateltava 5, johtava)*

Valittaessa automatisoitavia prosesseja on haastateltavien mukaan tärkeää huomioida prosessein monimutkaisuus, kuten useat rinnakkaiset vaiheet tai integraatiot muihin

järjestelmiin. Monimutkaisiakin prosesseja voidaan hallita tehokkaasti, kun kokonaiskuva säilyy selkeänä.

> *”Prosessin ei välttämättä tarvitse olla edes kovin monimutkainen, mutta yleensä hyviä tuloksia saadaan sellaisissa prosesseissa missä on useita peräkkäisiä tai rinnakkaisia vaiheita ja joilla on eri vastuuhenkilöitä sekä eri tiimejä, jotka tekevät näitä asioita. Mahdollisesti myös sellaiset, että meillä on integraatioita muihin järjestelmiin prosessin aikana. Myös sen tyyppiset, joissa prosessi voi haarautua useammiksi aliprosesseiksi ja pitää pystyä pitämään kuitenkin kokonaisuus hallinnassa.” (Haastateltava 2, johtava)*

Käyttäjien koulutuksen merkitys korostui molempien roolien vastauksissa. Haastateltavat korostivat, että alustan tekniset käsitteet sekä toimintalogiikka tulee ymmärtää hyvin, jotta käyttäjät osaavat hyödyntää alustaa tehokkaasti. Nämä olisi hyvä käydä käyttäjien kanssa yhdessä läpi sekä tarjota heille kattava dokumentaatio.

> *”Low-code alustoilla on omat käsitteensä, kuten ehdolliset tai automaattisesti käynnistyvät toimenpiteet. Kannattaa käyttää aikaa ymmärtääksesi, mitä tapahtuu missäkin vaiheessa. Hyvä tapa on ottaa olemassa oleva prosessi esimerkiksi ja käydä se läpi kokeneemman suunnittelijan kanssa.” (Haastateltava 4, toteuttava)*

Lisäksi haastateltavien mukaan tiimityön merkitys on tärkeä. Vastuu low-code alustan käytöstä ei tulisi olla ainoastaan yhden henkilön varassa, sillä tämä lisää riskiä, mikäli henkilöstö vaihtuu. Useammalle henkilölle tulisi kouluttaa alustan käyttö, mikä pienentää henkilöriskejä sekä parantaa osaamisen jakautumista organisaatiossa.

> *”Varmista, kuitenkin myös että vastuu alustan käytöstä ei ole vain yhden henkilön varassa. Tiimiytä vastuu, jotta useampi henkilö ymmärtää, miten alustaa käytetään ja kehitetään. Tämä vähentää riskiä, jos henkilöstö vaihtuu.” (Haastateltava 5, johtava)*

Johtavassa roolissa työskentelevät haastateltavat korostivat myös ulkoisen asiantuntija-avun merkitystä low-code alustojen käyttöönotossa. Alkuvaiheessa on hyvä tukeutua ammattilaiseen, joka voi auttaa prosessikehityksen alkuun ja varmistaa, että alusta otetaan käyttöön tehokkaasti. Tämä vähentää mahdollisia riskejä ja varmistaa, että alusta palvelee liiketoiminnan tavoitteita mahdollisimman hyvin.

> *”Kannattaa alkuvaiheessa ottaa ammattilainen avuksi, joka kertoo miten prosessikehittämistä tehdään. Kannattaa myös miettiä mikä on hyvä ja luotettava kumppani, jonka kanssa kehitystä lähdetään tekemään. Ominpäin ei kannata ainoastaan tilata low-code alustaa ja koittaa itse lähteä opettelemaan, se tulee usein kalliiksi.” (Haastateltava 2, johtava)*

Ammattilaisten käyttö voi haastateltavien mukaan lyhentää projektien kestoa ja investointina olla hyödyllinen. Tämä auttaa myös voi auttaa varmistamaan projektin onnistumisen.

> *”Suosittelen käyttämään ammattilaista alustan käyttöönotossa. Tämä lyhentää projektin kestoa ja pienentää riskejä. Investointi ammattilaisen apuun on kannattavaa, sillä se varmistaa sujuvan käyttöönoton ja tehokkaan käytön” (Haastateltava 5, johtava)*

5 Tulosten soveltaminen

Tässä pääluvussa esitellään empiirisessä tutkimuksessa esiin nousseita löydöksiä liittyen low-code alustojen hyödyntämiseen liiketoimintaprosessien automatisoinnissa. Tärkeimmät empiirisen tutkimuksen löydökset esitellään erikseen ja niitä analysoidaan syvällisemmin. Tuloksia analysoidaan myös teoreettisen viitekehyksen valossa. Tulokset vahvistavat aiemmassa kirjallisuudessa esitettyjä havaintoja, mutta tuovat myös uusia näkökulmia sekä käytännön haasteita. Lisäksi luvussa käsitellään empiirisen tutkimuksen aineiston rajallisuutta.

5.1 Tulosten tarkempi tarkastelu teoreettisen viitekehyksen valossa

Empiirisessä tutkimuksessa nousi low-code alustojen yhdeksi keskeiseksi eduksi niiden visuaaliset kehitystyökalut. Graafinen käyttöliittymä asiantuntijahaastattelujen mukaan parantaa käytettävyyttä sekä mahdollistaa eri tasoisten käyttäjien osallistumisen mukaan kehittämiseen. Tämä parantaa kehittämisen tehokkuutta sekä hyväksyttävyyttä. Teoreettisessa viitekehityksessä esimerkiksi Khorram et al., (2020) mukaan nähtiin myös low-code alustojen mahdollistavan laajemman osallistujajoukon osallistumisen kehitykseen sekä vähentävän koodausosaamisen tarvetta. Myös Krishnaraj et al., (2022) mukaan low-code alustojen hyöty on niiden nopeampi ratkaisujen toteutus. Empiirisessä tutkimuksessa kuitenkin painotettiin teoreettista viitekehitystä enemmän visuaalisuuden merkitystä erityisesti ratkaisujen hyväksyttävyyden näkökulmasta, joka ei teoreettisessa viitekehyksessä noussut esiin. Kun prosessi on ymmärrettävä, se myös sitouttaa yrityksen henkilöitä enemmän mukaan kehitykseen ja käyttöönottoon.

Empiirisessä haastattelututkimuksessa saadut tulokset tukevat aiempaa kirjallisuutta, jonka mukaan low-code alustat tarjoavat merkittäviä etuja yrityksille, jotka pyrkivät tehostamaan liiketoimintaprosessejaan ilman, että niiden tarvitsee investoida raskaisiin ohjelmistokehitystiimeihin (Kirikova & Rokis, 2023). Asiantuntijahaastatteluiden tulokset osoittavat, että kohdeyrityksessä low-code alustojen käyttöönotto tehosti liiketoimintaprosessien automatisointia erityisesti toistuvissa ja manuaalisissa tehtävissä. Tämä näkyi konkreettisesti, että prosessien automatisoinnin myötä prosessit ovat hallitumpia sekä henkilöriskit pienenivät.

Myös teoreettisessa viitekehityksessä kuvattiin automaation poistavan rutiininomaisia tehtäviä, mutta toisaalta myös parantavan työntekijöiden moraalia (Davis, 2021).

Kohdeyrityksessä asiantuntijahaastatteluissa havaitut joustavuuden edut low-code alustojen käyttöönotossa näkyivät siinä, että yritys pystyi nopeasti mukautumaan muuttuviin liiketoiminnan tarpeisiin jopa ilman ohjelmistopäivityksiä ja tuotannon käyttökatkoja. Teoreettisessa viitekehityksessä myös kuvattiin low-code alustojen rajoitusten olevan vähäisempiä kuin perinteisten kehitysmenetelmien sekä olevan joustavampi versiopäivitysten osalta (Krishnaraj et al., 2022). Teoriassa yhtenä joustavuuteen liittyvänä hyötynä kuvattiin alustojen yhtenäinen ympäristö, joka tarjoaa työkalut yhdessä paikassa selkeästi (Bock & Frank, 2021). Asiantuntijahaastatteluissa tämä etu ei kuitenkaan noussut esiin.

Kustannustehokkuus nousi yhdeksi low-code alustojen hyödyistä sekä teoreettisessa viitekehyksessä että empiirisen tutkimuksen asiantuntijahaastatteluissa. Asiantuntijahaastatteluissa korostettiin low-code alustojen kykyä vähentää koodaamisen tarvetta sekä olemassa olevien toteutuksien tehokkaan uudelleenkäyttämisen. Uudelleenkäyttö johti kustannussäästöihin erityisesti niissä tapauksissa, kun monet asiakkaat hyötyivät samasta ominaisuudesta. Teoreettisessa viitekehyksessä liiketoimintaprosessien automatisoinnin suureksi eduksi myös mainittiin kustannussäästöt (ICT Monitor Worldwide, 2023). Teoriassa myös tuotiin esille sama havainto, että low-code alustojen käyttö mahdollistaa sovellusten kehityksen vähemmällä määrällä koodia (Sanchis et al., 2020). Resurssien uudelleenkäyttö kuitenkin korostui enemmän kohdeyrityksien haastatteluissa kuin teoreettisessa viitekehityksessä.

Teoreettisen viitekehyksen mukaan low-code alustat soveltuvat hyvin yksinkertaisten ja korkean arvon tuovan tehtävien automatisointiin (Guilmette, 2022). Alustojen kyky hallita monimutkaisia prosesseja voi kuitenkin olla rajoitettua. Monimutkaiset toteutukset tarvitsevat usein myös enemmän räätälöintiä sekä mahdollisesti integraatioita muihin järjestelmään. (Bock & Frank, 2021) Kohdeyrityksen asiantuntijahaastatteluissa havaittiin myös, että erityisesti integraatioita ja useita poikkeuksia sisältävät prosessitoteutukset olivat haastavimpia toteuttaa. Vaikka low-code alustat tarjoavat rajapintoja sekä työkaluja integraatioiden toteuttamiseksi, kohdeyrityksen asiantuntijoiden mukaan monimutkaisimmat toteutukset vaativat usein uusia ominaisuuksia ja kehitystä, mikä poikkeaa low-code alustojen ominaisesta yksinkertaisuudesta ja helppokäyttöisyydestä. Monimutkaisuuden välttämiseksi kohdeyrityksen asiantuntijahaastatteluissa painotettiin, että prosessin toteuttajalla on suurin vastuu varmistaa toteutuksen yksinkertaisuus ja välttää tarpeetonta prosessin monimutkaistamista.

Empiirinen tutkimus osoitti, että koulutuksella on kriittinen rooli low-code alustojen onnistuneessa käyttöönotossa. Kohdeyrityksen asiantuntijahaastatteluissa ilmeni, että ilman riittävää koulutusta ja tukea käyttäjät eivät pystyneet hyödyntämään alustojen mahdollisuuksia täysin. Erityisesti kansalaiskeittäjien osallistuminen prosessien kehittämiseen edellyttää, että heille tarjotaan riittävä koulutus alustan käyttöön sekä prosessiautomaation periaatteisiin. Tämä johtaa siihen, että organisaation tulee investoida riittävästi sekä tekniseen että toiminnalliseen koulutukseen, jotta automatisointiprojektit voidaan toteuttaa onnistuneesti. Myös teoreettisessa viitekehyksessä esimerkiksi Printz et al. (2024) korostaa, että low-code alustojen tehokas käyttö vaatii organisaatioilta investointeja koulutukseen, jotta alustoja voidaan hyödyntää täysimääräisesti.

Teoreettinen viitekehys nostaa tietoturvaan liittyvä riskin, joka liittyy ei-teknisten käyttäjien eli kansalaiskehittäjien luomiin toteutuksiin low-code alustoilla. Bunce (2024) huomauttaa, että vaikka low-code alustat mahdollistavat laajemman käyttäjäjoukon osallistumisen prosessikehitykseen, tämä voi lisätä tietoturvariskejä, mikäli alustoilla ei ole riittäviä turvatoimia. Erityisesti ei-tekniset käyttäjät voivat tahattomasti luoda ratkaisuja, jotka sisältävät haavoittuvuuksia. Empiirisessä tutkimuksessa tunnistettiin myös tämä riski. Asiantuntijahaastatteluissa tuli kuitenkin myös esille koulutuksen kriittinen rooli tietoturvan hallinnassa. Haastatteluissa korostettiin, että tietoturvariskejä on tärkeää käsitellä yhdessä kansalaiskehittäjien kanssa, jotta ymmärretään miten eri muutokset voivat vaikuttaa tietoturvaan.

Teoreettisessa viitekehyksessä ratkaisuina turvallisuuden huolenaiheille esitettiin muun muassa alustan sisäänrakennettuja tietoturvaominaisuuksia, sisältäen esimerkiksi pääsynvalvonnan ja haavoittuvuustestauksen (Bunce, 2024). Myös asiantuntijahaastatteluissa näihin haasteisiin ratkaisuna esitettiin lokitietojen tallennusta ja hyödyntämistä sekä tarkkaa käyttäjienhallintaa.

Empiirisen tutkimuksen tulokset osoittavat, että low-code alustojen käytön hyödyt ovat suurelta osin yhdenmukaisia aiemman teoreettisen tutkimuksen kanssa, mutta myös uusia käytännön haasteita on noussut esiin. Kohdeyrityksen asiantuntijahaastatteluissa havaittiin, että vaikka low-code alustat mahdollistavat sovellusten nopean kehittämisen ja tuovat merkittävää joustavuutta liiketoimintaprosessien automatisointiin, käytännön toteutuksessa törmättiin haasteisiin muun muassa teknisen valmiuden ja koulutustarpeen osalta. Sovellusten nopea kehitys vaati jatkuvaa koulutusta ja teknistä tukea, jotta käyttäjät pystyivät hyödyntämään alustoja tehokkaasti ja välttämään mahdollisia virheitä sovellusten luomisessa.

Käytännössä low-code alustojen laajentaminen monimutkaisempien prosessien automatisointiin osoittautui vaikeaksi ilman syvempää teknistä osaamista. Vaikka alusta tarjoaa kattavasti erilaisia ratkaisuja, empiirinen tutkimus toi esiin, että yrityksissä tarvitaan riittävää teknistä osaamista, jotta automatisointi voidaan toteuttaa ilman ongelmia.

Low-code alustojen tarjoamien mahdollisuuksien onnistunut hyödyntäminen edellyttää yrityksiltä huolellista suunnittelua, teknistä osaamista ja koulutukseen investoimista. Koulutuksen avulla varmistetaan, että yrityksen eri osastot sekä eritasoiset käyttäjät voivat osallistua sovellusten kehittämiseen, mikä nopeuttaa päätöksentekoa ja vähentää yritysten riippuvuutta IT-asiantuntijoista. Tämä osallistava lähestymistapa tukee liiketoiminnan ketteryyttä ja varmistaa sen, että prosessitoteutuksessa tehdään oikeita asioita oikeaan aikaan.

Tutkimuksen tulokset tarjoavat yrityksille arvokasta tietoa siitä, miten low-code alustoja voidaan hyödyntää tehokkaasti. Alustojen kyky skaalautua liiketoimintaprosessien mukana mahdollistaa yrityksille asteittaisen etenemisen; automaatiota voidaan aloittaa yksinkertaisista työnkuluista ja laajentaa vähitellen monimutkaisempiin prosesseihin. Tämä antaa yrityksille joustavan ja kustannustehokkaan tavan kasvattaa automatisoitujen prosessien määrää samalla, kun ne sopeutuvat muuttuviin liiketoimintatarpeisiin.

5.2 Tutkimuksen aineiston rajallisuus

Tutkimuksen aineistossa oli tiettyjä rajoituksia, jotka voivat vaikuttaa tulosten yleistettävyyteen ja sovellettavuuteen.

Kvalitatiivinen haastattelu perustuu usein vain muutamaan tapaukseen ja siksi sitä ei voida yleistää. Kvalitatiiviset tutkimukset eivät voi kuten kvantitatiiviset tutkimukset osoittaa generalisoitavuutta tilastollisesti, vaan niiden on käytettävä analyyttistä yleistämistä. (Brinkmann, 2013) Tutkimuksen aineisto koostui yhden kohdeyrityksen haastatteluista, mikä voi rajoittaa tulosten yleistettävyyttä muihin organisaatioihin. Tutkimuksessa saatiin syvällistä tietoa kyseisen yrityksen low-code alustojen käytöstä, mutta eri organisaatioilla voi olla erilaiset lähtökohdat ja resurssit, jotka vaikuttavat onnistuneeseen käyttöönottoon.

Low-code teknologia kehittyy nopeasti ja tämän tutkimuksen tulokset kuvaavat tietyn organisaation tilannetta tietyllä hetkellä. Tulevaisuudessa alustoihin voi tulla uusia ominaisuuksia, jotka muuttavat niiden käyttötapoja sekä hyötyjä (Yan, 2021). Tämän vuoksi tulokset

saattavat vanhentua suhteellisen nopeasti, mikä on tyypillistä nopeasti kehittyvälle teknologialle.

Kvalitatiiviset haastattelut perustuvat subjektiivisiin kokemuksiin ja tulkintoihin. Vaikka täydellistä objektiivisuutta ei kvalitatiivisissa haastatteluissa voida saavuttaa, voidaan tutkimuksessa pyrkiä siihen esimerkiksi arvioimalla eri tulkintoja kriittisesti. (Brinkmann, 2013) Tässä tutkimuksessa käytetyt mittarit, kuten tehokkuus ja joustavuus, perustuivat subjektiivisiin arvioihin haastatteluissa. Tämä saattaa tuoda tiettyä tulkinnanvaraisuutta, vaikka tutkimuksessa pyrittiin huolellisesti analysoimaan vastauksia ja varmistamaan niiden luotettavuus.

6 Johtopäätökset

Tämän tutkimuksen tavoitteena oli tutkia low-code alustojen käytön hyötyjä ja haasteita liiketoimintaprosessien automatisoinnissa, erityisesti parhaiden käytäntöjen ja teknisten rajoitteiden osalta. Tavoitteena tutkimuksella oli tarjota tietoa yrityksille, jotka harkitsevat low-code alustojen käyttöönottoa liiketoimintaprosessien automatisoinnissa. Työssä asetettiin kolme eri tutkimuskysymystä, joiden pohjalta tutkimusta alettiin tekemään. Seuraavaksi vastataan tutkimuskysymyksiin työn kirjallisuuskatsauksen sekä empiirisen tutkimuksen avulla saatujen tulosten pohjalta.

Päätutkimuskysymys: *Mitkä ovat hyödyt ja haasteet low-code alustojen käytössä liiketoimintaprosessien automatisoinnissa?*

Työn kirjallisuuskatsauksen sekä empiirisen tutkimuksen perusteella tulokset osoittavat, että low-code alustojen hyödyt kuten nopeus, kustannustehokkuus, joustavuus, prosessien skaalautuvuus ja visuaalisen suunnittelun helppous ovat merkittäviä erityisesti yksinkertaisten prosessien automatisoinnissa. Toisaalta monimutkaisemmat prosessit, turvallisuus, suorituskyky ja alustojen rajoitukset toivat esiin haasteita, jotka vaativat käyttäjiltä sekä kehittäjiltä syvempää teknistä osaamista ja koulutusta.

Yksi merkittävimmistä eduista low-code alustoilla on niiden kyky nopeuttaa liiketoimintaprosessien kehitystä ja automatisointia. Alustat tarjoavat käyttäjille graafisia työkaluja, kuten valmiiksi määriteltyjä vedä-ja-pudota komponentteja, jotka mahdollistavat ratkaisujen nopean toteutuksen ilman syvällistä ohjelmointiosaamista. Uudelleenkäytettävät komponentit myös mahdollistavat niiden hyödyntämisen erilaisissa ratkaisuissa, mikä tuo kustannussäästöjä. Graafiset työkalut lyhentävät ratkaisujen kehitysnopeutta verrattuna perinteisiin ohjelmistokehitysmenetelmiin, joissa manuaalinen ohjelmointi vie enemmän aikaa.

Low-code alustojen keskeinen hyöty on niiden käyttäjäystävällisyys ja mahdollisuus osallistaa henkilöitä mukaan ratkaisun kehittämiseen myös ilman syvällistä teknistä osaamista. Low-code alustoilla ei-tekniset käyttäjät eli kansalaiskehittäjät voivat osallistua ratkaisujen suunnitteluun ja prosessien automatisaatioon. Kansalaiskehittäjien osallistuminen lisää

organisaatioiden joustavuutta sekä vähentää riippuvuutta ohjelmistokehittäjistä. Tämä parantaa IT-osaston ja liiketoiminnan välistä yhteistyötä ja nopeuttaa kehittämistä.

Joustavuus on low-code alustojen tärkeä etu, sillä ne mahdollistavat ratkaisujen nopean iteroinnin ja ratkaisujen mukauttamisen käyttäjien tarpeisiin. Low-code alustat mahdollistavat kehittäjien tekemien muutosten nopean käyttöönoton jopa ilman käyttökatkoja, mikä tekee kehitystyöstä sujuvampaa. Alustojen skaalautuvuus myös mahdollistaa sen, että voidaan aloittaa pienistä ja yksinkertaisista automaatiohankkeista ja kasvattaa niitä monimutkaisempiin toteutuksiin organisaation tarpeiden mukaan. Vaiheittainen lähestymistapa on auttanut organisaatioita hallitsemaan kehitysprojekteja paremmin sekä pienentänyt riskejä.

Low-code alustojen käyttö liiketoimintaprosessien automatisoinnissa tuo hyötyjä organisaation tuottavuuden parantamiseen. Liiketoimintaprosessien automatisointi vähentää rutiininomaisia ja manuaalisia työtehtäviä ja parantaa työntekijöiden ajankäytön tehokkuutta. Kun manuaalisten työtehtävien määrä vähenee myös henkilöriskit yrityksessä pienenevät, sillä prosessit eivät ole riippuvaisia yksittäisten työntekijöiden osaamisesta tai läsnäolosta. Tämä johtaa liiketoiminnan parempaan tuottavuusteen, kun toistuvat tehtävät hoidetaan tehokkaammin ja inhimillisten virheiden määrä vähenee.

Turvallisuus on low-code alustoilla merkittävä haaste erityisesti, kun alustoilla eritasoiset käyttäjät kehittävät ratkaisuja. Kansalaiskehittäjät eivät välttämättä ole tietoisia parhaista tietoturvakäytännöistä, mikä voi lisätä riskiä tietoturvahaavoittuvuuksille. Kansalaiskehityksen tietoturvallisuuden riskit on otettava huomioon erityisesti silloin, kun kehitettävä ratkaisu käsittelee luottamuksellisia tietoja. Kansalaiskehittäjien ratkaisut vaativat usein lisäkatselmointeja sekä testausta ennen niiden käyttöönottoa mikä vaatii lisäresursseja.

Low-code alustojen tekninen rajoittuneisuus on yksi merkittävä haaste. Low-code alustoilla tekniset mahdollisuudet voivat olla rajallisia erityisesti monimutkaisemmissa liiketoimintaprosesseissa. Tämä voi johtaa tilanteeseen, jossa organisaatio joutuu tekemään kompromisseja sovelluksen toiminnallisuuksien suhteen tai turvautumaan kustomoituun ratkaisuun.

Liiketoimintaprosessien automatisoinnin haasteena low-code alustoilla on myös suorituskyvyn ja skaalautuvuuden ongelmat. Vaikka low-code alustat tarjoavat työkaluja nopeaan kehitykseen ja prosessien automatisointiin, niiden suorituskyky voi heiketä, kun käsiteltävät prosessit ja tietomäärät kasvavat. Näissä tapauksissa kehittävät voivat joutua tekemään

ylimääräistä optimointia, kuten prosessien uudelleenorganisointia ja suunnittelua, jotta ongelmat saadaan ratkaistua.

Monimutkaisten prosessien automatisoinnissa low-code alustojen käyttö voi olla rajoittuneempaa. Monimutkaisemmat prosessit vaativat syvempää teknistä osaamista sekä esimerkiksi integraatioita, mikä rajoittaa kansalaiskehittäjien kykyä toteuttaa ratkaisuja itsenäisesti. Tämä vähentää low-code alustan tarjoamia hyötyjä kuten nopeutta ja helppoutta.

Osatutkimuskysymys 1*: Mitkä ovat suurimmat haasteet ja tekniset rajoitukset, joita organisaatiot kohtaavat pyrkiessään hyödyntämään low-code alustoja monimutkaisten liiketoimintaprosessien automatisoinnissa?*

Low-code alustat on suunniteltu nopeuttamaan kehitysprosesseja tarjoamalla käyttäjälle visuaalisia työkaluja sekä valmiita komponentteja. Alustojen kyky käsitellä monimutkaisempia liiketoimintaprosesseja on kuitenkin rajallinen ja prosessien onnistunut automatisointi vaatii syvempää teknistä osaamista ja usein myös räätälöintiä.

Low-code alustojen käyttöä rajoittaa niiden rajallisuus mukautua monimutkaisiin liiketoimintaprosesseihin, jotka sisältävät paljon eri sääntöjä, poikkeuksia tai integraatioita. Low-code alustat tarjoavat usein rajalliset mahdollisuudet ratkaisujen räätälöintiin mikä voi johtaa siihen, että ratkaisuja joudutaan täydentämään manuaalisella ohjelmoinnilla ja uusilla ominaisuuksilla. Monimutkaiset liiketoimintaprosessiratkaisut vaativat usein myös integraatioita eri järjestelmien ja tietolähteiden välillä. Integraatiot edellyttävät usein ohjelmointiosaamista, mikä monimutkaistaa toteutusta sekä kehitystä.

Kuten päätutkimuskysymyksen vastauksessa aiemmin mainittiin, monimutkaisten liiketoimintaprosessien automatisoinnissa kansalaiskehittäjät kohtaavat haasteita. Kun prosessien monimutkaisuus kasvaa, kansalaiskehittäjien tekninen osaaminen on usein riittämätöntä onnistuneen toteutuksen aikaansaamiseksi. Tämä johtaa siihen, että monimutkaisemmissa ratkaisuissa teknisemmät kehitystiimit ottavat suuremman vastuun ratkaisun toteuttamisesta ja kanaslaiskehittäjin toteutukset rajoittuvat yksikertaisempiin automaatioratkaisuihin.

Osatutkimuskysymys 2: *Mitä parhaita käytäntöjä ja oppeja voidaan saada organisaatioilta, jotka ovat onnistuneesti hyödyntäneet low-code alustoja prosessien automatisoinnissa?*

Onnistuneissa prosessiautomaatioprojekteissa low-code alustoilla korostuu huolellinen suunnittelu, kattava käyttäjien koulutus, ulkopuolisen avun hyödyntäminen sekä eritasoisten käyttäjien laaja osallistaminen.

Onnistuneessa prosessiautomaatioprojektissa huolellinen suunnittelu on välttämätöntä. Suunnitteluvaiheessa on tärkeää ottaa huomioon organisaation nykyiset prosessit ja valikoida niistä ensimmäiseksi automatisoitaviksi sellaiset, jotka sisältävät paljon manuaalivaiheita. Lisäksi on tärkeää dokumentoida sekä ymmärtää omat liiketoimintaprosessit ennen low-code alustalla tehtävää prosessiautomatisaatiota, jotta voidaan priorisoida oikeat prosessit ja aloittaa oikeanlaisesta. Valittaessa automatisoitavia prosesseja on tärkeää huomioida prosessien monimutkaisuus, kuten useat rinnakkaiset vaiheet. Monimutkaisiakin prosesseja voidaan hallita ja automatisoida tehokkaasti, jos kokonaiskuva säilyy selkeänä.

Prosesseja mallintaessa on hyödyllistä aloittaa yleiskuvasta ja siirtyä vähitellen yksityiskohtiin, jotta monimutkaisuus ei kasva hallitsemattomaksi. Yksinkertaiset ratkaisut eivät ainoastaan helpota prosessien kehittämistä, vaan mahdollistavat myös asiakkaiden paremman ymmärryksen, miten prosessit toimivat. Tämä mahdollistaa sen, että asiakkaat voivat seurata kehitystyötä sekä osallistua siihen mukaan.

Lisäksi käyttäjien koulutus on avainasemassa. Low-code alustojen tehokas hyödyntäminen edellyttää, että käyttäjät ymmärtävät alustan tarjoamat mahdollisuudet ja osaavat käyttää niitä oikein. Koulutuksen tulisi kattaa sekä tekniset taidot että prosessien hallinta, jotta käyttäjät voivat itsenäisesti kehittää ja ylläpitää prosesseja.

Ulkopuolisen avun hyödyntäminen voi olla erittäin hyödyllistä käyttöönotossa. Asiantuntijat, joilla on kokemusta low-code alustojen käyttöönotosta, voivat tarjota arvokasta tietoa ja tukea projektin eri vaiheissa. Asiantuntijat voivat auttaa välttämään yleisiä ongelmia ja varmistaa, että projekti etenee suunnitellusti.

Eritasoisten käyttäjien osallistaminen on tärkeää. Low-code alustojen yksi suurimmista eduista on niiden helppokäyttöisyys, mikä mahdollistaa laajemman osallistumisen kehitysprojekteihin. Graafinen käyttöliittymä parantaa käytettävyyttä ja tekee alustan käytöstä intuitiivista myös niille, joilla ei ole syvällistä teknistä osaamista. Tämä edistää yhteistyötä ja parantaa ratkaisun hyväksyttävyyttä.

Tämän tutkimuksen empiirisessä osuudessa keskityttiin ainoastaan yhden IT-alalla toimivan yrityksen näkemyksiin low-code alustojen käytöstä liiketoimintaprosessien

automatisoinnissa. Tämä rajaus tarjoaa syvällisen ymmärryksen kyseisen yrityksen ekosysteemissä, mutta samalla se rajoittaa tulosten yleistettävyyttä muihin toimialoihin ja organisaatioihin. Jatkotutkimus voisi laajentaa näkökulmaa tutkimalla low-code alustojen käyttöä eri toimialoilla, kuten terveydenhuollossa, rahoitusalalla ja julkisella sektorilla. Tällainen laajempi tutkimus antaisi kattavamman kuvan low-code alustojen soveltuvuudesta ja haasteista erilaisissa toimintaympäristöissä.

Potentiaalinen jatkotutkimusaihe on low-code alustojen käytön pitkäaikaiset vaikutukset liiketoimintaprosessien tehokkuuteen ja skaalautuvuuteen. Nykyinen tutkimus keskittyi lyhyen aikavälin hyötyihin ja haasteisiin, mutta pitkäaikaisvaikutusten tutkiminen voisi paljastaa miten low-code alustat vaikuttavat organisaatioiden toimintaan pidemmällä aikavälillä. Lisäksi pitkäaikaisvaikutusten tutkiminen voi tuoda esiin mahdollisia piileviä kustannuksia tai riskejä, jotka eivät ole ilmeisiä lyhyellä aikavälillä.

Tämä tutkimus ei ottanut huomioon kohdeorganisaation tai sen asiakkaiden teknisiä valmiuksia low-code alustojen prosessiautomaatioprojektien toteutuksessa. Jatkotutkimusaiheena olisi mahdollista keskittyä eri organisaatioiden teknisiin valmiuksiin ja niiden vaikutukseen low-code alustojen hyödyntämisessä. Organisaatioiden tekniset valmiudet, kuten henkilöstön tekninen osaaminen ja resurssien saatavuus voivat merkittävästi vaikuttaa low-code alustojen liiketoimintaprosessien automatisoinnin käyttöönottoon ja käytön onnistumiseen.

Tässä tutkimuksessa havaittiin, että monimutkaisissa prosessitoteutuksissa low-code alustoilla tarvitaan usein myös ohjelmointia. Jatkotutkimuskohde olisi vertailla monimutkaisten liiketoimintaprosessien toteutuksia low-code alustoilla ja perinteisellä ohjelmoinnilla. Tutkimuksessa olisi mahdollista tarkastella kumpi lähestymistapa on pitkällä aikavälillä kustannustehokkaampi ja kestävämpi. Huomiota tulisi kiinnittää alustojen suorituskykyyn, ylläpitokustannuksiin sekä prosessien muokattavuuteen ja skaalautuvuuteen. Vertailun avulla voidaan saada selkeämpi kuva siitä, milloin low-code alustojen käyttö on perusteltua verrattuna perinteiseen ohjelmointiin.

Lähteet

Abdelwahab, M. B. M. & Helal, I. M. A. (2023) “Advanced Techniques for Business Process Automation: Insights and Challenges”, Intelligent Methods, Systems, and Applications (IMSA). [Verkkojulkaisu]. 2023 IEEE. pp. 303–308.

Alamin, M. A. A. et al. (2023) “Developer discussion topics on the adoption and barriers of low code software development platforms”, Empirical software engineering : an international journal [Verkkojulkaisu]28 (1), 4–4.

Alasuutari, P. (2011) Laadullinen tutkimus 2.0. Tampere: Vastapaino.

Antwiadjei, L. & Huma, Z. (2021) “Comparative analysis of low-code platforms in automating business processes”, Asian Journal of Multidisciplinary Research & Review, Volume 3 Issue 5 [Verkkojulkaisu]

Antwiadjei, L. & Huma, Z. (2024) “Evaluating the Impact of ChatGPT and Advanced Language Models on Enhancing Low-Code and Robotic Process Automation”, Journal of Science & Technology, Volume 5 Issue 1 [Verkkojulkaisu]

Bandara, W., G. G. Gable, and M. Rosemann (2005) "Factors and Measures of Business Process Modelling: Model Building Through a Multiple Case Study", European Journal of Information Systems (14)4, pp. 347-360

Bock, A. C. & Frank, U. (2021) “In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms”, ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). [Online]. 2021 IEEE. pp. 57–66.

Bock, A. C. & Frank, U. (2021) “Low-Code Platform. Business & information systems engineering” [Verkkojulkaisu]63 (6), 733–740.

Bubalo, A. & Tankovic, N. (2023) ”Modeling low-code databases with executable UML”, Human Systems Engineering and Design, Vol. 112, 436–445

Bucaioni, A. et al. (2022) “Modelling in low-code development: a multi-vocal systematic review”, Software and systems modeling. [Verkkojulkaisu] 21 (5), 1959–1981.

Bunce, C. (2024). “Low-Code Challenges (And Solutions)” bizagi.com. Saatavilla: https://www.bizagi.com/en/blog/low-code-challenges

Bratincevic, J., Taylor-Huot R., Stone Z. (2024) “The Low-Code And Digital Process Automation Market, 2023 To 2028”, Tech. Rep., Forrester

Brinkmann, S. (2013) “Qualitative interviewing”, 1st ed. Oxford, England: Oxford University Press.

Ciucan-Rusu, L. et al. (2023) “Low-code solutions integration for digital process automation - application in management”, 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet). [Verkkojulkaisu]. 2023 IEEE. pp. 1–5.

Davenport, T. & Johannessen, C. (2021) ”When Low-Code/No-Code Development Works — and When It Doesn’t.”, Harvard Business Review [Verkkojulkaisu] Saatavilla: https://hbr.org/2021/06/when-low-code-no-code-development-works-and-when-it-doesnt

Davis, T. (2021) “The 60 Second Marketer: 9 Benefits of Business Process Automation”, Chatham: Newstex.

Desmond, M. et al. (2022) A No-Code Low-Code Paradigm for Authoring Business Automations Using Natural Language. arXiv.org. [Verkkojulkaisu]

Di Ruscio, D. et al. (2022) “Low-code development and model-driven engineering: Two sides of the same coin?”, Software and systems modeling. [Online] 21 (2), 437–446.

Dumas, M. et al. (2018) “Fundamentals of Business Process Management”, 2nd ed. 2018 [Verkkojulkaisu]. Berlin, Heidelberg: Springer Berlin Heidelberg.

Elshan, E., Dickhaut, E. & Ebel, P. (2023) “An Investigation of Why Low Code Platforms Provide Answers and New Challenges”, Proceedings of the 56th Hawaii International Conference on System Sciences, pp. 6159-6165.

Express Computer (2019) “How to achieve true business process automation with a low code platform”, Express Computer.

Fortune Business Insights (2023) “Low Code Development Platform Market Size, Share & Industry Analysis, By Component (Platform and Services), By Deployment (Cloud and On-premises), By Enterprise Size (Large Enterprises and SMEs), By Application Type (Web & Cloud Based, Mobile Based, and Desktop Based), By Industry (BFSI, Healthcare,

Education, IT and Telecommunication, Media & Entertainment, Manufacturing, Government, Retail, and Others), and Regional Forecast”

Fortune Business Insights (2023) “Business Process Management (BPM) Market Size, Share & Industry Analysis, By Component (Solution/Tools/Platform, Services), By Deployment (On-Premise, Cloud), By Organization Size (Large Enterprises, Small and Medium Sized Enterprises), By Functionality (Accounting & Finance, Sales & Marketing, Human Resource, Supply Chain Management, Operations Management & Support, Others), By Industry (BFSI, IT & Telecommunication, Healthcare, Manufacturing, and Others) and Regional Forecast, 2024-2032”

Funk, D. et al. (2023) “Creating a Low-Code Business Process Execution Platform With Python, BPMN, and DMN”, IEEE software. [Verkkojulkaisu] 40 (1), 9–17.

Gartner, (2021) “Gartner says cloud will be the centerpiece of new digital experiences,” 2021. [Verkkojulkaisu] Saatavilla: https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences

Glesne, C. & Peshkin, A. (1992) “Becoming qualitative researchers: An introduction”, London: Longman.

Guilmette, A. (2022) “Workflow automation with Microsoft Power Automate : use business process automation to achieve digital transformation with minimal code”, 2nd ed. Birmingham, UK: Packt Publishing, Limited.

Hammersley, M. (1992) “Deconstructing the qualitative-quantitative divide” Teoksessa J. Brannen (toim.) Mixing methods: Qaulitative and quantitative reseach. Aldershot: Avebury, 39-58

Hirsjärvi, S. & Hurme, H. (2022) ”Tutkimushaastattelu : teemahaastattelun teoria ja käytäntö”, [2. painos]. Helsinki: Gaudeamus.

Horváth, B. et al. (2020) ‘Towards the next generation of reactive model transformations on low-code platforms: three research lines’, Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. [Verkkojulkaisu]. New York, NY, USA: ACM. pp. 1–10.

Hyvärinen, M. et al. (2017) ”Tutkimushaastattelun käsikirja”, Tampere: Vastapaino.

ICT Monitor Worldwide. (2023) “Lack of know-how slows down business process automation benefits”, Amman: Disco Digital Media, Inc.

JourneyApps (2023) “10 Challenges Facing Low-Code Platforms (And How We Solve Them)”, Saatavilla: https://journeyapps.com/blog/10-challenges-facing-low-code-platforms-and-how-we-solve-them

Kang, G. et al. (2024) “Business Process Modeling for Industrial Internet Application via BPMN Extension”, IEEE transactions on automation science and engineering. [Online] 1–17.

Khorram, F. et al. (2020) “Challenges & opportunities in low-code testing”, Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. [Verkkojulkaisu]. 2020 New York, NY, USA: ACM. pp. 1–10.

Kirchmer, M., Scheer, AW. (2004) "Business Process Automation — Combining Best and Next Practices", Springer, Berlin, Heidelberg

Kirikova, M., Rokis, K. (2023) “Exploring Low-Code Development: A Comprehensive Literature Review”, Complex Systems Informatics and Modeling Quarterly (CSIMQ)

Kovalev, S. et al. (2022) “Approach to the Implementation of Intelligent Low-Code Platforms”, Proceedings of the Sixth International Scientific Conference Intelligent Information Technologies for Industry (IITI’22). Switzerland: Springer International Publishing AG. p.

KPMG (2023) “Low-code: The unifying fabric of the digital enterprise”, Saatavilla: https://kpmg.com/kpmg-us/content/dam/kpmg/pdf/2022/low-code-unifying-fabric.pdf

Krishnaraj, N. et al. (2022) “Comparative Study on Various Low Code Business Process Management Platforms”, 2022 International Conference on Inventive Computation Technologies (ICICT). [Verkkojulkaisu]. 2022 IEEE. pp. 591–596.

Lethbridge, T. C. (2021) “Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering”, Leveraging Applications of Formal Methods, Verification and Validation. [Verkkojulkaisu]. Cham: Springer International Publishing. pp. 202–212.

Liu, D. et al. (2024) “What’s Wrong With Low-Code Development Platforms? An Empirical Study of Low-Code Development Platform Bugs”, IEEE transactions on reliability. [Verkkojulkaisu] 73 (1), 1–15.

Martin, J. (1982) “Application Development Without Programmers”, first ed., PrenticeHall, Paramus, NJ,

Martinek-Jaguszewska, K. & Rogowski, W. (2023) “Development and Validation of the Business Process Automation Maturity Model: Results of the Delphi Study”, Information systems management. [Verkkojulkaisu] 40 (2), 169–185

Mishra, S., Kandampully Kailas, S.D. ja M.K., B.N. (2019) “People & Process Dimensions of Automation in Business Process Management Industry”, 1. International Journal of Engineering and Advanced Technology, [Verkkojulkaisu] 8(6).

Pinho, D. et al. (2023) “What about the usability in low-code platforms? A systematic literature review”, Journal of computer languages. [Verkkojulkaisu]

PR Newswire (2021) “Bizagi Delivers 288% ROI According to Independent Research Firm: The Total Economic Impact Of Bizagi’s Low-Code Intelligent Process Automation Platform showcases an average of $33.3M in business benefits over three years”

Prinz, N., Huber, M., Leonhardt, J. & Riedinger, C. (2024) “Unleash the Power of Citizen Development: Leveraging Organizational Capabilities for Successful Low-Code Development Platform Adoption”, 1. Proceedings of the 57th Hawaii International Conference on System Sciences

Sahay, A. et al. (2020) “Supporting the understanding and comparison of low-code development platforms”, 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). [Verkkojulkaisu]. 2020 IEEE. pp. 171–178.

Sahay, A. et al. (2023) “Analyzing business process management capabilities of low-code development platforms”, Software, practice & experience. [Verkkojulkaisu] 53 (4), 1036–1060.

Rao, N. et al. (2024) “AI for Low-Code for AI”, in Proceedings of the 29th International Conference on Intelligent User Interfaces. [Verkkojulkaisu]. New York, NY, USA: ACM. pp. 837–852.

Richardson, C., Rymer J. R., Mines C., Cullen A., Whittaker D. (2014) “New Development Platforms Emerge for Customer-Facing Applications”, Tech. Rep., Forrester, p. 16.

Rokis, J. & Kirikova, M. (2023) “Low-code development platforms: Reducing the need for deep programming skills and enabling rapid application development and business process automation”, Journal of Software Engineering and Applications, 16(2), pp. 123-135.

Sanchis, R., García-Perales, Ó., Fraile, F. ja Poler, R., 2020 “Low-Code as Enabler of Digital Transformation in Manufacturing Industry”, Applied Sciences, 10(1), s.12. Saatavilla: https://doi.org/10.3390/app10010012

Thaker, B. K. (2022) “Low-Code Platforms and Business Process Transformation”, International Journal of Technology Engineering Arts Mathematics Science Vol. 2, No. 2, December 2022, pp. 20~23

tetratech (2022) 3BL Blogs - 3BL Media: Leveraging No/Low Code Platforms for Business Process Management. Chatham: Newstex

Vincent P, Natis Y, Iijima K, Wong J, Ray S, Jain A, Leow A (2020) “Magic quadrant for enterprise low-code application platforms”, Gartner Report September 2020, Gartner

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

Waszkowski, R. (2022) “Low-Code Development Platform for Business Process Automation: Aurea BPM Production Management and Process Control”

Yan, Z. (2021) “The Impacts of Low/No-Code Development on Digital Transformation and Software Development”, arXiv.org. [Verkkojulkaisu]

Liite 1. Asiantuntijahaastattelujen puolistrukturoidut kysymykset

1. Voisitko kertoa roolistasi yrityksessä ja pääasiallisista työtehtävistäsi?
2. Kauan olet työskennellyt nykyisessä roolissa ja onko sinulla aikaisempaa kokemusta low-code alustoista?
3. Millainen koulutus sinulla on taustalla? Mihin perustat ymmärryksesi low-code alustoista? Onko se enemmän käytännön kokemusta, koulutusta, vai itseopiskelua?
4. Miten ensimmäisen kerran tutustuit low-code alustoihin? Millaisia olivat ensikokemuksesi?
5. Miten näkemyksesi low-code alustojen käytöstä on muuttunut ajan myötä? Onko jokin erityinen kokemus tai projekti vaikuttanut tähän?
6. Miten arvioisit low-code alustojen hyödyllisyyttä yrityksessänne ja asiakkaiden ratkaisuissa? Ovatko ne helpottaneet työtäsi tai parantaneet prosesseja?
7. Miten low-code alustat ovat muuttaneet tapaasi lähestyä liiketoimintaprosessien automatisointia? Voisitko antaa esimerkin?
8. Miten kuvailisit liiketoimintaprosessien merkitystä yrityksesi toiminnassa? Mikä rooli sinulla on näiden prosessien kehittämisessä ja ylläpidossa?
9. Miten henkilökohtaiset kokemuksesi low-code alustoista ovat muokanneet käsitystäsi niiden hyödyllisyydestä ja soveltuvuudesta liiketoimintaprosessien automatisointiin?

Teemakohtaiset kysymykset

Teemakohtaiset kysymykset on jaettu roolikohtaisesti kahteen eri tasoon: Johtava -ja Toteuttava toiminta.

Teema 1: Monimutkaiset prosessiautomaatiot – Johtava toiminta

a. Miten varmistatte ja priorisoitte riittävät resurssit (esim. aika, raha, osaaminen) monimutkaisten prosessiautomaatioprojektien onnistuneeseen toteutukseen?
b. Mitä hyötyjä ja haasteita näet monimutkaisten low-code prosessiautomaatiototeutusten tuovan liiketoiminnalle?

Teema 1: Monimutkaiset prosessiautomaatiot – Toteuttava toiminta

a. Miten lähestyt monimutkaisten low-code prosessiautomaatiototeutusten suunnittelua ja toteutusta low-code alustalla? Mitkä ovat suurimmat tekniset ja toiminnalliset haasteet, joita kohtaat?
b. Miten alusta tukee monimutkaisten automaatioiden toteutusta? Onko käytössä joitakin erityisiä työkaluja tai ominaisuuksia, jotka ovat erityisen hyödyllisiä?

Teema 2: Tuotteen soveltuvuus ja ominaisuuksien puutteet – Johtava toiminta

a. Mitä osa-alueita arvioitte, jos low-code alustalta puuttuu jokin keskeinen ominaisuus?
b. Miten varmistatte, että alustan nykyiset ja tulevat ominaisuudet tukevat liiketoimintastrategiaanne pitkällä aikavälillä?

Teema 2: Tuotteen soveltuvuus ja ominaisuuksien puutteet – Toteuttava toiminta

a. Mitä teet, jos low-code alustan liiketoimintaprosessiautomaation alusta ei sisällä tarvittavia ominaisuuksia?
b. Oletko joutunut kehittämään omia ratkaisuja? Miten tämä on vaikuttanut projektin toteutukseen?

Teema 3: Suorituskyky ja skaalautuvuus – Johtava toiminta

a. Miten arvioitte ja varaudutte suorituskykyyn ja skaalautuvuuteen liittyviin haasteisiin ennen projektin aloittamista? Miten resursoitte ja allokoitte aikaa näiden ongelmien ratkaisemiseen?
b. Miten priorisoitte suorituskykyyn ja skaalautuvuuteen liittyvät kehitystarpeet projektin ollessa käynnissä, jotta projekti saadaan maaliin ajallaan ja budjetissa?
c. Miten suorituskyvyn ja skaalautuvuuden vaatimukset ohjaavat low-code alustanne kehitystä? Kuinka ne tukevat yrityksen strategisia tavoitteita?

Teema 3: Suorituskyky ja skaalautuvuus – Toteuttava toiminta

a. Miten huomioit suorituskyvyn ja skaalautuvuuden prosessien suunnittelussa ja toteutuksessa? Miten varmistat, että ratkaisut toimivat tehokkaasti myös suurissa ja vaativissa prosesseissa?
b. Millaisia haasteita olet kohdannut suorituskyvyn ja skaalautuvuuden osalta? Miten nämä haasteet ovat ratkaistu?

Teema 4: Kansalaiskehityksen haasteet ja hyödyt– Johtava toiminta

a. Miten kansalaiskehittäjien osallistuminen vaikuttaa projektin resurssien hallintaan ja aikatauluihin? Miten varmistatte, että he saavat riittävästi tukea ja koulutusta?
b. Miten arvioitte kansalaiskehittäjien vaikutusta projektin laatuun? Miten tasapainottelette kehittäjien resurssien käytön ja kansalaiskehittäjien tuottaman lisäarvon välillä?
c. Miten arvioitte kansalaiskehittäjien panoksen merkitystä yrityksen liiketoimintastrategiassa? Miten tämä näkyy käytännön päätöksenteossa?

Teema 4: Kansalaiskehityksen haasteet ja hyödyt– Toteuttava toiminta

a. Miten kansalaiskehittäjien (esim. asiakkaiden) luomat ratkaisut vaikuttavat koko järjestelmän arkkitehtuuriin? Mitä hyötyjä tai haasteita olet havainnut näissä ratkaisuissa?
b. Miten varmistat, että kansalaiskehittäjien luomat ratkaisut ovat laadukkaita? Millaisia käytäntöjä olet kehittänyt näiden ratkaisujen hallintaan?

Teema 5: Turvallisuus ja tiedonsiirto – Johtava toiminta

a. Miten varmistatte, että low-code projektien tietoturva on hoidettu asianmukaisesti ilman, että se hidastaa projektin etenemistä tai kuluttaa liikaa resursseja?
b. Miten jaatte resursseja tietoturvaan ja tiedonsiirtoon liittyvien ongelmien ratkaisemiseksi? Miten aikataulutatte nämä tehtävät projektinhallinnassa?
c. Miten tietoturva- ja tiedonsiirtoasiat on otettu huomioon yrityksen strategiassa low-code projektien osalta? Kuinka nämä tekijät vaikuttavat päätöksiinne alustan kehityksessä?

Teema 5: Turvallisuus ja tiedonsiirto – Toteuttava toiminta

a. Miten huomioit tietoturvan low-code ratkaisujen kehittämisessä? Millaisia erityisiä haasteita kohtaat tiedonsiirrossa ja tietoturvassa?
b. Miten huolehdit siitä, että kansalaiskehittäjien luomat ratkaisut täyttävät tietoturvavaatimukset? Millaisia menetelmiä käytät varmistamaan tiedonsiirron turvallisuuden?

Teema 6: Low-code prosessiautomaation hyödyt – Johtava toiminta

a. Mitkä ovat suurimmat hyödyt, joita olette saavuttaneet ottamalla käyttöön low-code alustan ratkaisuja omiin prosesseihin?
b. Mitä etuja olet huomannut low-code alustan käytöstä prosessien automatisoinnissa ja ratkaisujen kehittämisessä? (esim. asiakasprojekteissa asiakkailla)
c. Onko prosessien automatisointi low-code alustan avulla parantanut yrityksenne kilpailukykyä tai tehokkuutta?

Teema 6: Low-code prosessiautomaation hyödyt – Johtava toiminta

a. Mitä etuja olet huomannut low-code alustan käytöstä prosessien automatisoinnissa ja ratkaisujen kehittämisessä? (esim. asiakasprojekteissa asiakkailla tai käyttäessäsi itse)

10. Mitä neuvoja antaisitte muille yrityksille, jotka harkitsevat low-code alustan käyttöönottoa prosessien automatisointiin? (Johtava toiminta)
11. Mitä neuvoja antaisit muille ratkaisusuunnittelijoille/prosessitoteuttajille, jotka alkavat käyttää low-code alustaa prosessien automatisointiin? (Toteuttava toiminta)