

Otto Juntunen

# API-RAJAPINNAT OHJELMISTOROBOTIIKASSA



JYVÄSKYLÄN YLIOPISTO  
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA  
2023

# TIIVISTELMÄ

Juntunen, Otto

API-rajapinnat ohjelmistorobotiikassa

Jyväskylä: Jyväskylän yliopisto, 2023, 30 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Seppänen, Ville

Gradun tavoitteena on pyrkiä selvittämään, millaisissa hyötyjä API-rajapintojen käyttämisellä voidaan saavuttaa ohjelmistorobotiikassa. Yleensä ohjelmistorobotiikassa hyödynnetään olemassa olevia käyttöliittymiä ja niitä automatisoidaan. Työni kuitenkin perehtyy API-rajapintojen hyödyntämisen mahdollisuuksiin.

Asiasanat: ohjelmistorobotiikka, API

## **ABSTRACT**

Juntunen, Otto

APIs in robotic process automation

Jyväskylä: University of Jyväskylä, 2023, 30 pp.

Information Systems, Master's Thesis

Supervisor: Seppänen, Ville

The aim of the master's thesis is to try to find out, what kind of benefits one can get from using APIs in robotic process automation. Usually we use existing user interfaces in robotic process automation. My master's thesis try to find out how to use APIs.

Keywords: robotic process automation, API

## KUVIOT

KUVIO 1	Ohelmistorobotiikan mahdollisuudet (Aalst ym., 2018) .....	12
---------	--	----

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVIOT JA TAULUKOT

1	JOHDANTO.....	6
2	OHJELMISTOROBOTIIKKA.....	9
	2.1 Liiketoimintaprosessien automaatio.....	9
	2.2 Ohjelmistorobotiikan määritelmä .....	10
	2.3 Ohjelmistorobotiikan hyödyt.....	12
	2.4 Ohjelmistorobotiikan haasteet .....	13
3	API-RAJAPINNAT JA NIIDEN HYÖDYNTÄMINEN OHJELMISTOROBOTIIKASSA.....	15
	3.1 API-rajapinnat .....	15
	3.2 API-rajapinnat ohjelmistorobotiikassa .....	16
	3.3 Hyperautomaatio.....	17
4	TUTKIMUSMENETELMÄ .....	19
5	ANALYYSI.....	21
	5.1 Muutosten hallinnan helppous.....	21
	5.2 Toteutuksen helppous.....	22
	5.3 Luotettavuus.....	23
	5.4 Tehokkuus .....	23
	5.5 Testauksen helppous .....	24
6	POHDINTA JA YHTEENVETO .....	25
	LÄHTEET .....	28
	LIITE 1 SISÄLTÖANALYYSIN KOODIT .....	30

# 1 JOHDANTO

Ohjelmistorobotiikka on suhteellisen uutena toimintamallina nostamassa päätään liiketoimintaprosessien automatisoinnissa. Automatisointia tehdään pitkälti matkimalla ihmisen toimintaa ja hyödyntämällä erilaisia käyttöliittymiä. Tällaisella toiminnalla pystytään automatisoimaan monia prosesseja, joihin perinteiset tietojärjestelmien integraatiot ei välttämättä pystyisi tai perinteisten integraatioiden toteuttaminen ei syystä tai toisesta ole kannattavaa. Ohjelmistorobotiikassa hyödynnettävät menetelmät eivät sinänsä ole uusia, mutta niiden yhdistäminen ohjelmistorobotiikan kaltaiseksi kokonaisuudeksi on kuitenkin suhteellinen uusi asia.

Monissa aiemmissa tutkimuksissa ohjelmistorobotiikkaa ja API-rajapintojen hyödyntämistä on vertailtu keskenään ja ne nähdään välillä jopa vastakohtina. Kummassakin on selvästi hyvät ja huonot puolensa ja tavoitteenani tutkimuksesani onkin pyrkiä edistämään näiden kahden yhdistämistä tavalla tai toisella. Tällöin molemmista maailmoista voitaisiin tarvittaessa ottaa hyvät puolet ja käyttää kulloiseenkin tapaukseen parhaiten soveltuvaa teknologiaa. En siis itse näe API-rajapintoja ja ohjelmistorobotiikkaa niinkään vastakohtina vaan ennemminkin toisiaan sopivassa suhteessa täydentävinä teknologioina.

Gartnerin tutkimuksessa API-rajapintojen hyödyntämisen on havaittu olevan vahvasti osa liiketoimintaprosessien automatisointia tulevaisuudessa. Vuoteen 2024 mennessä 95 % ohjelmistorobotiikan toimijoista ajatellaan tarjoavan perinteisen käyttöliittymillä tapahtuvan automatisoinnin lisäksi myös API-rajapintojen hyödyntämistä (Ray ym., 2022). On siis selvästi nähtävissä, että API-rajapintojen hyödyntäminen liiketoimintaprosessien automatisoinnissa on kasvussa. Alalla on siis nähtävissä kehitystä, jonka myötä halutaan pystyä kattavasti hyödyntämään erilaisia menetelmiä automatisoinnissa eikä pysytellä pelkästään yhdessä menetelmässä. Prosesseja pystytäänkin jatkossa automatisoimaan kattavammin, kun prosesseissa pystytään hyödyntämään kattavasti erilaisia menetelmiä.

Tutkimuskysymystä asetellessani pyrin miettimään sitä, millä tavoin voisin parhaiten edistää tätä kahden eri teknologian yhdistämistä. Mitään kovin yleispätevää vastausta en missään vaiheessa edes kuvitellut pystyväni antamaan

gradussani, mutta vähän pyrin antamaan suuntaa asioille. Tutkimuskysymyksestäni muodostuikin seuraavanlainen:

Minkälaisia hyötyjä voidaan saavuttaa hyödyntämällä API-rajapintoja ohjelmistorobotiikan viitekehyksessä?

Gartnerin tutkimusraportissa hyperautomaatiosta mainitaan API-rajapintojen ja käyttöliittymien kautta tapahtuvan automaation yhdistämisen olevan tärkeä osa hyperautomaatiota. Eri teknologiat myös helposti siiloutuvat, mutta olisi tärkeää päästä tarkastelemaan eri teknologioita kokonaisuutena, jotta päästään automatisoimaan myös vähän haastavampia prosesseja (Ray, Vincent, Guttridge & Karamouzis, 2021). Hyperautomaatio onkin ohjelmistorobotiikan ohella yksi mielenkiintoisimmista aiheista liiketoimintaprosessien automatisoinnissa, kun varsinainen käyttöliittymien kautta tapahtuva automatisointi on alkanut saamaan vakinaisempaa asemaa. Tämän takia tutkimukseni API-rajapintojen hyödyntämisestä on mielenkiintoinen ja tarpeellinen. Koitan tutkimukseni avulla rakentaa jonkinlaista käsitystä siitä, mitä ohjelmistorobotiikassa voidaan saavuttaa siirtymällä hiukan hyperautomaation suuntaan. Kuitenkaan tutkimuksen tarkoituksena ei ole pelkästään keskittyä hyperautomaatioon.

API-rajapintojen käyttämisen muodostuessa tärkeäksi osaksi liiketoimintaprosessien automatisointia onkin mielenkiintoista päästä selvittämään, mitä hyötyjä niiden avulla pystytään saavuttamaan. Eri tekniikoita hyödyntäen pystytään jatkossa automatisoimaan enenevässä määrin prosesseja ja perinteisesti hankalasti automatisoitavissa olleet prosessitkin saattavat tulla olemaan järkevästi automatisoitavissa. Kun helposti automatisoitavat prosessit on jo automatisoitu niin pitääkin panostaa myös vähän haastavampiin automatisointikohteisiin. Hyperautomaatiossa pyritäänkin yhdistämään monenlaisia eri tapoja, joilla hankalaksi luultuja prosesseja pystytään paremmin hallitsemaan ja automatisoimaan ja sitä kautta tehostamaan toimintaa. Hyperautomaation voidaankin ajatella olevan tärkeä osa liiketoimintaprosessien automatisointia tulevaisuudessa.

Tutkimuksessani tarkastelen yhtä asiakkaalle tehtävää liiketoimintaprosessin automatisointia, jossa on tarkoitus käyttää osittain API-rajapintoja ja osittain tarkoituksena on hyödyntää myös käyttöliittymien kautta tapahtuvaa automatisointia. Tapaus on tutkimuksen kannalta mielenkiintoinen, kun siinä päästään tarkastelemaan erilaisia automatisoinnin tapoja ja vertailemaan niiden eroja. Projektin oltua jonkin aikaa tuotannossa pidin ryhmähaastattelun, jonka avulla pyrin selvittämään API-rajapintojen hyödyntämisellä saavutettuja etuja. Haastateltua käytetään tämän tutkielman aineistona.

Graduni muodostuu neljästä pääluvusta, joista ensimmäisessä käsittelen ohjelmistorobotiikkaa ja toisessa API-rajapintoja ja niiden hyödyntämistä ohjelmistorobotiikassa. Pysin määrittelemään ohjelmistorobotiikkaa ja tarkastelemaan sen hyötyjä ja haasteita. API-rajapinnat myös määrittelen lyhyesti, jonka jälkeen tarkastelen niiden hyödyntämistä ohjelmistorobotiikassa. Lopuksi luon lyhyen vilauksen hyperautomaatioon, jossa ohjelmistorobotiikkaan tuodaan mukaan tekoälyä ja muita älykkäitä ominaisuuksia. Tekoäly kuitenkin on rajattu

tutkimukseni ulkopuolelle. Kahdessa seuraavassa luvussa käyn läpi käytettyjä tutkimusmenetelmiä ja tarkastelen työni lopputuloksia.



## 2 OHJELMISTOROBOTIIKKA

Tässä luvussa käsitellään liiketoimintaprosessien automatisointitapana ohjelmistorobotiikkaa. Liiketoimintaprosesseja on kyllä automatisoitu jo pidemmän aikaa, mutta suhteellisen uutena menetelmänä graduni keskittyy ohjelmistorobotiikkaan. Luvun tavoitteena on esitellä ohjelmistorobotiikkaa yleisellä tasolla ja määrittellä käsitettä. Tarkastellaan samalla hiukan ohjelmistorobotiikkaan liittyviä käsitteitä ja sen lähellä olevia toimintoja. Katsotaan myös, mitä kaikkea ohjelmistorobotiikka on ja myös sitä, mitä se ei ole. Pyrin myös avaamaan ohjelmistorobotiikkaan liittyviä hyötyjä ja haasteita.

### 2.1 Liiketoimintaprosessien automaatio

Ohjelmistorobotiikan voidaan katsoa olevan osa liiketoimintaprosessien automatisointia (business process automation, BPA), mutta liiketoimintaprosessien automatisointia on tehty jo pidemmän aikaa ennen ohjelmistorobotiikan kehittymistä, mutta sen menetelmät ovat olleet jossain määrin alkeellisempia. BPA on kuitenkin tuonut organisaatioille mahdollisuuden nykyisten prosessiensa ainakin osittaiseen automatisointiin ilman että varsinaiseen liiketoimintaan ja sen prosesseihin on täytynyt tehdä sen suurempia muutoksia (Dumas, La Rosa, Mendling & Reijers, 2013). Voidaankin nähdä, että BPA on avannut tietä ja mahdollistanut ohjelmistorobotiikan kehittymisen automatisointimenetelmäksi.

Ohjelmistorobotiikka onkin nykyään yksi liiketoimintaprosessien automatisointiin käytetyistä tekniikoista. Ohjelmistorobotiikka ei siis ole ensimmäinen tai ainoa tekniikka liiketoimintaprosessien automatisoinnissa, mutta sen avulla automatisoinnista on tullut helpompaa ja monet automaatiokohteet ovat nykyisin taloudellisesti järkeviä toteuttaa.

## 2.2 Ohjelmistorobotiikan määritelmä

Ohjelmistorobotiikka (englanniksi robotic process automation, RPA) on ylätasoinen termi käytetyille menetelmille, kun automatisoidaan tietojärjestelmiä samaan tapaan kuin ihminenkin niitä käyttäisi (Aalst, Bichler & Heinzl, 2018). Tietojärjestelmiä pyritään ohjelmistorobotiikan avulla automatisoidessa hyödyntämään sillä tavoin, ettei järjestelmiin itsessään tarvitse tehdä muutoksia, vaan pystytettiin hyödyntämään jo olemassa olevia käyttöliittymiä ja muita rajapintoja. Tällaisella menettelyllä pyritään saamaan automatisointi tehtyä pienemmällä panostuksella, kuin tehtäessä muutoksia olemassa oleviin järjestelmiin.

Usein ohjelmistorobotiikan voidaan ajatella muodostuvan käyttöliittymien hyödyntämisestä prosessien automatisoinnissa. Tämä ei kuitenkaan ole koko totuus, vaan ohjelmistorobotiikassa voidaan hyödyntää myös muitakin rajapintoja. Määritelmää voidaan täydentää esimerkiksi seuraavalla tavalla:

Robotic process automation (RPA) tools perform "if, then, else" statements on structured data, typically using a combination of user interface (UI) interactions, or by connecting to APIs to drive client servers, mainframes or HTML code. An RPA tool operates by mapping a process in the RPA tool language for the software "robot" to follow, with runtime allocated to execute the script by a control dashboard. (Tornbohm & Durnie, 2017, s. 2)

Tässä tuodaan esiin API-rajapintojen hyödyntämisen sisältyminen ohjelmistorobotiikan määritelmään. Käsittelen myöhemmin enemmän API-rajapintojen hyödyntämistä ja sen tuomia mahdollisuuksia ohjelmistorobotiikalla liiketoimintaprosesseja automatisoidessa.

IEEE ei ole määritelmässään niin tarkka käytettävien teknologioiden suhteen vaan määrittelee ohjelmistorobotiikan yleisemmällä tasolla seuraavasti:

**robotic process automation (RPA):** A preconfigured software instance that uses business rules and predefined activity choreography to complete the autonomous execution of a combination of processes, activities, transactions, and tasks in one or more unrelated software systems to deliver a result or service with human exception management. (IEEE Corporate Advisory Group, 2017, s. 11)

Tämä yleisemmällä tasolla tehty määritelmä jättää enemmän mahdollisuuksia käytettävien teknologioiden suhteen. Kuitenkin se kuvaa hyvin ohjelmistorobotiikkaa ja sitä kuinka sen avulla automaatioita toteutetaan.

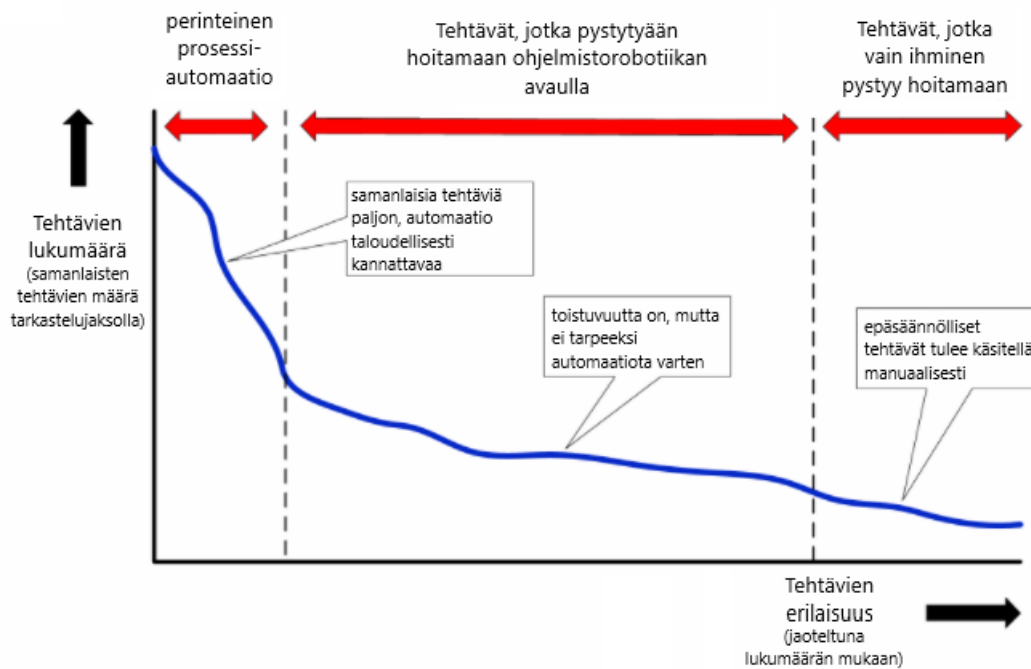
Kattavassa kirjallisuuskatsauksessaan Syed ym. (2020) tutkivat 125 ohjelmistorobotiikkaa käsittelevää artikkelia. Artikkeleista ainoastaan 24 yritti määrittellä ohjelmistorobotiikkaa. Kuitenkin tutkimuksessa havaittiin yhteisiä teemoja ja useimmat tutkituista artikkeleista olivat määritelleet tavalla tai toisella ohjelmistorobotiikan avulla korvattavan ihmisten tekemää manuaalisyötä ohjelmistoroboteilla. Tutkimuksessa kävi myös ilmi artikkelien määrittelevän ohjelmistorobotiikan käyttävän käyttöliittymiä samaan tapaan kuin ihmisetkin niitä käyttäisi

(Syed ym., 2020) Näin ollen voidaan ajatella ohjelmistorobotiikan jäljittelevän ihmisen toimintaa tietojärjestelmiä käyttäessään.

Aguirre ja Rodriguez (2017) esittelevät tutkimuksessaan myös prosessin valintaan vaikuttavia tekijöitä. Näitä tekijöitä ovat vähäinen kognitiivisten päätösten teon tarve, suuri volyyymi, useiden kohdejärjestelmien käyttö, rajallinen poikkeuskäsittely ja manuaalisen työn virheherkkyys. Näiden tekijöiden ohjaamana ohjelmistorobotiikan avulla automatisoitavaksi voidaan valita sellaisia prosesseja, joiden automatisointi onnistuu paremmin ja tehokkaammin kuin pienemmän automatisointipotentialin omaavat prosessit.

Tutkimuksessaan Asatiani ja Penttinen (2016) kuvaavat ohjelmistorobotiikassa imitoitavan ihmisen tekemistä automatisoidessa strukturoituja prosesseja. Tutkimuksessa kerrotaan järjestelmiä hyödynnettävän tietokoneen ruudulta käsin API-rajapintojen sijaan (Asatiani ja Penttinen, 2016). Koska tutkimukseni käsittelee API-rajapintojen hyödyntämistä, ei ohjelmistorobotit kuitenkaan ole täysin erillään perinteisemmistä rajapinnoista. Voidaan kuitenkin nähdä, että varsinkin ohjelmistorobotiikan alkuaikoina käyttöliittymien kautta tapahtuva automatisointi nähtiin helposti jonkinlaisena vastakohtana API-rajapintojen hyödyntämiselle. Kuitenkin nykyään ollaan tulossa pisteeseen, jossa halutaan päästää kattavasti hyödyntämään eri tekniikoita, jotta päästäisiin mahdollisimman kattavasti automatisoimaan eri prosesseja.

Aalst ym. (2018) esittelevät tutkimuksessaan työn pitkän hännän (kuvio 1). Tämä tarkoittaa sitä, että mitä suurempi määrä prosessissa on samanlaisia tapauksia ja mitä pienempi määrä erilaisia tapausten vaihtoehtoja, sitä helpompi prosessi on automatisoida. Helpoimmat automatisoitavat kohteet ovat luultavasti jo automatisoitu aiemmin perinteisillä menetelmillä, mutta ohjelmistorobotiikka pyrkii asettumaan kaikkein helpoimpien ja kaikkein vaikeimpien tapausten välimaastoon. Kaikkea ei ole järkevä automatisoida ja ihmisiä tarvitaan tulevaisuudessakin tuottavassa työssä, mutta ohjelmistorobotiikan avulla ihmiset pystyvät siirtymään rutiininomaisista tehtävistä yhä tuottavampiin tehtäviin.



KUVIO 1 Ohjelmistorobotiikan mahdollisuudet (Aalst ym., 2018)

### 2.3 Ohjelmistorobotiikan hyödyt

Ohjelmistorobotiikka on tullut täydentämään tarvetta automatisoida yhä suurempi määrä liiketoimintaprosesseja. Ohjelmistorobotiikan avulla pyritäänkin saavuttamaan hyötyjä nopeammin ja tehokkaammin kuin perinteisiä menetelmiä käyttämällä.

Penttisen, Kasslinin ja Asatianin (2018) mielestä ohjelmistorobotiikka on suhteellisen kustannustehokas automatisointitapa, jolla hyötyjä pystytään saavuttamaan lyhyessäkin aikataulussa. Lisäksi he kertovat ohjelmistorobotiikan avulla pystyttävän automatisoimaan prosesseja, joiden automatisointi olisi aiemmin ollut liian kallista. Myöskin he tuovat esiin sen, että ohjelmistorobotteja hyödyntämällä pystytään automatisoimaan myös sellaisia järjestelmiä, joihin muunlaisten rajapintojen toteuttaminen olisi muuten hankalaa tai mahdotonta (Penttinen ym., 2018).

Ohjelmistorobotiikassa ei välttämättä tarvitse tehdä muutoksia olemassa oleviin järjestelmiin, jolloin vältetään raskaammilta taustajärjestelmien integraatioilta (Asatiani & Penttinen, 2016). Tämän ansiosta ohjelmistorobotiikan avulla pystyttäisiin tekemään nopeitakin kokeiluja ja saavuttamaan sitä kautta hyötyjä, joiden saavuttamiseen muilla menetelmillä voitaisiin joutua tekemään suuriakin investointeja.

Tutkimuksessaan Anagnoste (2017) tuo myös esiin erilaisia ohjelmistorobotiikalla saavutettavia hyötyjä. Tärkeimpinä hyötyintä tutkimuksessa nostetaan

esiin tuottavuuden kasvattaminen pienillä muutoksilla, hyvä sijoitetun pääoman tuottoaste ja ohjelmistorobotiikan halpuus perinteisempiin menetelmiin verrattuna (Anagoste, 2017). Ohjelmistorobotiikan avulla voidaan siis saavuttaa organisaatiossa merkittäviäkin muutoksia suhteellisen pienellä panostuksella.

Šimek ja Šperka (2019) nostavat tutkimuksessaan esiin ohjelmistorobotiikan suurimpina hyötyinä sen ketteryyden ja joustavuuden. Ohjelmistorobotiikan avulla pystytään lyhyessäkin läpimenoajassa saavuttamaan organisaation kannalta merkittäviäkin hyötyjä. Tutkimuksessaan he tuovat esiin myös tarpeen ohjelmistorobottien orkestroinnille, kun ohjelmistorobottien määrä kasvaa tarpeeksi suureksi. Tutkimuksessa tuodaan myös esiin prosessien selkeä kuvaus ja määrittely tärkeänä alkutehtävänä ohjelmistorobotiikkaa hyödynnettäessä (Šimek ja Šperka, 2019).

Jotkin lähtöasetelmat tulisi olla kunnossa, jotta ohjelmistorobotiikan avulla voitaisiin saavuttaa onnistuneesti sille asetettuja tavoitteita. Willcocks, Lacity ja Craig (2015) esittelevät tutkimuksessaan kolme lähtökohtaa, joista ensimmäisenä mainitaan se, että tulisi alusta alkaen mahdollistaa ratkaisun skaalautuminen mahdollisimman kattavaksi. Toisena lähtökohtana tutkimuksessa esitetään se, että kaikki tarpeelliset sidosryhmät tulisi pitää mukana mahdollisimman aikaisesta vaiheesta lähtien. Kolmantena lähtökohtana he nostavat esiin sen, ettei ohjelmistorobotiikan avulla kannata haaveilla pelkästään nopeista voitoista, vaikka nekin olisivat mahdollisia (Willcocks ym., 2015). Kun lähtökohdat ovat kunnossa niin ohjelmistorobotiikalla on paremmat mahdollisuudet onnistua eikä onnistuminen ole pelkästään tuurin varassa.

## 2.4 Ohjelmistorobotiikan haasteet

Vaikka ohjelmistorobotiikan avulla pystytään saavuttamaan monia hyötyjä, liittyy siihen kuitenkin omat haasteensa. Eri automatisointitapoja vertaillessa ja sopivaa tapaa valittaessa tuleekin koittaa selvittää eri menetelmiin liittyvät hyödyt ja haasteet ja valita niiden perusteella sopivin toimintatapa.

Asatiani ja Penttinen (2016) nostavat ohjelmistorobotiikan haasteena esiin sen, että ohjelmistorobotiikka sopii vain tiettyntyyppisten selkeiden prosessien automatisointiin. He mainitsevat myös perinteisen järjestelmien välisen integraation olevan ohjelmistorobotiikkaa parempi automatisointitapa ja näkevät ohjelmistorobotiikan vain tietynlaisena välivaiheena (Asatiani & Penttinen, 2016). Sittemmin ohjelmistorobotiikka on kasvanut suurempaan rooliin ja se on nähtävissä vähintäänkin keskipitkän aikavälin ratkaisuna eikä välttämättä pelkkänä välivaiheena.

Tutkimuksessaan Šimek ja Šperka (2019) nostavat esiin sen, että käyttöliittymien päälle tehtävä automatisointi ei ole yhtä luotettavaa kuin API-rajapintojen hyödyntäminen. Oman tutkimukseni tavoitteena on pyrkiä tuomaan esiin ohjelmistorobotiikassa perinteisen käyttöliittymäautomatisoinnin ja API-rajapintojen hyödyntämisen erilaisuuksia. Tavoitteena olisi pyrkiä tuomaan esiin

erilaisten vaihtoehtojen hyvät ja huonot puolet ja vähän kertoa eri vaihtoehtojen soveltuvuudesta erilaisiin käyttökohteisiin.

Willcocks ym. (2015) esittelevät tutkimuksessaan erilaisia ohjelmistorobotiikan haasteita. Heidän tutkimuksessaan haasteena nousee esiin se, voidaanko ohjelmistorobotiikalla oikeasti saavuttaa parempia tuloksia nopeammin ja halvemmalla. He myös esittelevät haasteena sen, että ohjelmistorobotiikan tulisi mahdollisesti olla kuitenkin IT-projekti vaikkei ohjelmistorobotiikassa välttämättä tarvitakaan kovin merkittävää teknistä osaamista (Willcocks ym., 2015).

Ohjelmistorobotiikassa lokien taso voi joskus olla heikkoa ja se tuo omat ongelmansa virheenselvitykseen (Agostinelli, Marrella ja Mecella, 2020). Tämän takia lokituksen laatuun olisi hyvä panostaa ohjelmistorobotiikan prosesseja kehitettäessä. Ohjelmistorobotiikkaan liittyy kuitenkin hyvin kiinteänä osana manuaalisesti tehtävä virheenselvitystyö, joka on paljon helpompaa, kun käytettävissä on hyvää lokidataa.

### 3 API-RAJAPINNAT JA NIIDEN HYÖDYNTÄMINEN OHJELMISTOROBOTIIKASSA

API-rajapintojen hyödyntämistä ohjelmistorobotiikassa käsitellään hyvin vähän akateemisessa kirjallisuudessa. Niiden hyödyntämistä voidaan sivuta ohimennen, mutta varsinaista aiempaa tutkimusta aiheesta oli vaikea löytää. Osittain siitä syystä asiaa onkin hyvä tutkia. Koitan alkuun vähän avata API-rajapintoja ja niiden määritelmää, jonka jälkeen katsotaan vähän niiden hyödyntämistä ohjelmistorobotiikan parissa. Lopuksi vähän perehdytään myös hyperautomaatioon.

#### 3.1 API-rajapinnat

Erilaiset palvelut tarjoavat erilaisia rajapintoja käytettäväksi yhä enenevässä määrin. Rajapinnoista aiemmin suosittu SOAP-palvelut ovat jäämässä taka-alalle REST-palvelujen noustessa pintaan. (Neumann, Laranjeiro ja Bernardino, 2018). Erilaisten API-rajapintojen avulla sovellukset ja palvelut voivat vaihtaa ja välittää tietoa keskenään.

Kirjallisuudesta löytyy monia API-rajapintojen määritelmiä ja niistä yksi kuuluukin seuraavasti:

API, Application Programming Interface. Ohjelmointirajapinta, jonka avulla eri sovellukset voivat jakaa tietoa, toiminnallisuutta ja muita resursseja toisten kanssa. API voi olla REST API eli sitä voidaan käyttää internetissä ohjelmointikielystä riippumatta tai se voi olla tietyn ohjelmointikielen tai -kirjaston rajapinta. (Moilanen, Niinioja, Seppänen ja Honkanen, 2018, s. 211)

Tämän määritelmän mukaan API-rajapintojen avulla pystytään hyödyntämään eri sovellusten välille erilaisten resurssien vaihtoa. Ohjelmistorobotiikka on nähty vaihtoehtona API-rajapintojen hyödyntämiselle, mutta kuitenkin on mielenkiintoista koittaa selvittää sitä, voisiko ohjelmistorobotiikkaa ja API-

rajapintojen käyttöä yhdistää. Tämän takia tutkielmassani pyrinkin selvittämään API-rajapintojen käyttämisen mahdollisuuksia ohjelmistorobottikan parissa.

Toinen API-rajapintojen määritelmä kuuluu seuraavan laisesti:

APPLICATION PROGRAMMING INTERFACES (APIs) are, in the simplest term, specifications that govern interoperability between applications and services. Over the years, the API paradigm has evolved from beginnings as purpose-built initiatives to spanning entire application domains. (Vukovic ym., 2016, s. 1)

Tämän näkemyksen mukaan API-rajapinnat voidaan nähdä jopa omana tarkoituksenaan eikä pelkästään vain osana laajempaa kokonaisuutta. Ohjelmistorobotiikan tapauksessa on kuitenkin ehkä merkityksellisempää paneutua sellaisiin rajapintoihin, jotka ovat osa laajempaa kokonaisuutta.

### 3.2 API-rajapinnat ohjelmistorobotiikassa

Ohjelmistorobotiikassa monesti ajatellaan automatisointi totutettavan ihmiskäyttäjän toimintoja matkimalla. Tällöin pääasiassa prosessit toimisivat erilaisilla käyttöliittymillä samaan tapaan kuin ihmiskäyttäjätkin järjestelmää käyttäisivät. Tutkimukseni kannalta on kuitenkin mielenkiintoista selvittää sitä, millä tavoin API-rajapintoja voitaisiin hyödyntää ohjelmistorobotiikassa käyttöliittymäautomatisoinnin rinnalla sen sijaan että tehtäisiin automaatioita pelkästään käyttöliittymien päälle.

Aiheesta ei ole kovin paljo aiempaa tutkimustietoa, mutta jotain viittauksia aiheeseen kuitenkin löytyy. Osittain tämän vuoksi tutkimuksen kohde on mielenkiintoinen ja ajankohtainenkin. API-rajapintojen hyödyntämiseen on myös käytännön tasolla tarvetta ja olisikin myös hyvä saada jonkinlaista tutkimustietoa myös käytännön kokeilujen rinnalle.

Šimek ja Šperka (2019) tutkimuksessaan toteavat perinteisten API-rajapintojen käytön olevan toimintavarmempaa kuin ohjelmistorobottien. Tutkimuksessa todetaan ohjelmallisesti käyttöliittymien käyttöön liittyvän aina kuitenkin jonkinlaista epävarmuutta. (Šimek ja Šperka, 2019). Sen sijaan että jouduttaisiin tekemään valinta näiden kahden välillä, voisi vaihtoehtona olla kuitenkin molempien hyödyntäminen. Ne asiat, jotka kannattaisi tehdä API-rajapintoja hyödyntämällä voisi edelleen tehdä API-rajapintojen kautta, kun taas ne asiat, jotka kannattaisi tehdä ohjelmistorobotteja hyödyntäen voitaisiin tehdä ohjelmistorobotteja hyödyntäen.

Tutkimuksessaan Penttinen ym. (2018) koittavat löytää kriteereitä, joiden perusteella voitaisiin valita, toteutetaanko automatisointi perinteisellä järjestelmien yhdistämisellä käyttäen rajapintoja hyväksi vai hyödynnetäänkö ohjelmistorobotiikkaa. Tutkimuksessa löydettiinkin hyvä nippu kriteereitä, joiden perusteella valintaa voi ainakin koittaa ohjata oikeaan suuntaan (Penttinen ym. 2018). Vaikkakin tutkimus asettaakin ohjelmistorobotiikan ja perinteisen järjestelmäintegraation API-rajapintoja hyödyntämällä vastakkain, on siinä kuitenkin hyviä



pointteja, joiden perusteella ohjelmistorobotiikan sisälläkin voidaan tehdä mahdollinen valinta käytetystä automatisointimenetelmästä.

### 3.3 Hyperautomaatio

Vaikkakin tutkimukseni ei varsinaisesti käsittele hyperautomaatiota, on sitä kuitenkin hyvä sivuta jonkin verran. Hyperautomaatio on ohjelmistorobotiikankin piirissä melko uusi asia, eikä sitä välttämättä ole kovin tarkasti tai yhteneväisesti määritelty aiemmissa tutkimuksissa. Hyperautomaatiossa ohjelmistorobotiikkaan yhdistetään tekoälyä ja muita älykkäitä ratkaisuja (Lasso-Rodriguez ja Winkler, 2020). Tekoälyn hyödyntäminen on tutkimuksessani tarkoituksella rajattu ulkopuolelle, mutta tämä on kuitenkin hyvä ottaa huomioon.

Gartner määrittelee hyperautomaation seuraavalla tavalla:

Business-driven hyperautomation is a disciplined approach that organizations use to rapidly identify, vet and automate as many business and IT processes as possible. Hyperautomation involves the orchestrated use of multiple technologies, tools or platforms to achieve business results. These include, but are not limited to, AI, machine learning, event-driven software architecture, robotic process automation (RPA), iPaaS, packaged software, and process/task automation tools. (Welsh de Grimaldo, Cana & Nandan, 2022, s. 81)

Hyperautomaatiossa pyritään siis mahdollisimman kattavasti automatisoimaan kaikki mahdolliset automatisoitavissa olevat prosessit hyödyntäen ja yhdistellen erilaisia automatisoinnin työkaluja ja menetelmiä. Ohjelmistorobotiikka on merkittävässä roolissa hyperautomaatiota ja organisaatiossa voidaankin hyödyntää ohjelmistorobotiikkaa yksinkertaisempien rutiininomaisten tehtävien, kuten tietojen syötön automatisointiin. Puolestaan edistyneempiä ja älykkäämpiä toimintoja voidaan samassa organisaatiossa hyödyntää monimutkaisempia prosesseja, joita ei välttämättä voitaisi hoitaa pelkästään perinteisen ohjelmistorobotiikan avulla.

Hyperautomaatiossa on tavoitteena vähentää ihmisen manuaalisen työn tarve liiketoimintaprosesseissa minimiin. Optimitilanteessa automatisoitu prosessi pyörii mahdollisimman paljon riippumattomana ihmisen toiminnasta. Tähän tavoitteeseen pääsemiseksi hyperautomaatiossa käytetäänkin edistyneempiä teknologioita monimutkaisen tiedon analysointiin ja ymmärtämiseen. Tämän perusteella automaatiossa pystytään tekemään päätöksiä monimutkaisemmissakin tilanteissa.

Hyperautomaation avulla pystytään saavuttamaan tehokkaampaa toimintaa organisaatiossa. Liiketoimintaprosesseista saadaan tehokkaampia ja virheiden riskiä saadaan pienennettyä, kun tarve ihmisen manuaaliselle työlle saadaan minimoitua tai peräti poistettua kokonaan. Prosessien suorittamiseen kuluva aika lyhenee ja kustannukset pienenevät samalla kun henkilöstöä voidaan vapauttaa mielekkäämpiin ja enemmän lisäarvoa tuottaviin työtehtäviin. Hyperautomaation avulla organisaatiot pysyvät muuttuvassa toimintaympäristössä

myös paremmin kilpailukykyisinä, kun muuttuviin markkinatilanteisiin ja asiakastarpeisiin pystytään reagoimaan nopeammin.

## 4 TUTKIMUSMENETELMÄ

Valitsin käyttäväni työssäni kvalitatiivista tutkimusmenetelmää. Tähän tulokseen tulin pääasiassa sen vuoksi, kun käsittelemästäni aiheesta löytyi vain hyvin vähän aiempaa tutkimusta ja aineiston kerääminen onnistui järkevämmiin kvalitatiiviseen tutkimukseen kuin kvantitatiiviseen tutkimukseen. Työssäni on käytetty kvantitatiivisen tutkimuksen induktiivista lähestymistapaa (Yin, 2016). Koska aiheestani oli vain vähän aiempaa tutkimusta, ei ollut valmiita konsepteja, joita olisin voinut hyödyntää aineistoni keräämisessä. Tämän vuoksi induktiivinen lähestymistapa oli looginen valinta.

Tutkimusmenetelmäksi työhöni valikoitui sisältöanalyysi. Sisältöanalyysin avulla lähdeaineistosta pystytään löytämään työn kannalta merkittävät seikat helposti. Sisältöanalyysin eri haaroista käyttöön valikoitui konventionaalinen sisältöanalyysi. Konventionaalisisessa sisältöanalyysissä käytettävät koodit määritellään lähdeaineiston analysoinnin aikana. (Hsieh ja Shannon, 2005). Työlleni ei ollut löydettävissä sopivaa valmista teoriaa, jonka johdosta konventionaalinen sisältöanalyysi sopi erinomaisesti tähän tapaukseen.

Työni kohteena oli yksi ohjelmistorobotiikan toteutusprosessi, jossa asiakkaalle toteutettiin automaatio, joka toimi sekä kolmannen osapuolen, että automaation toteuttajan oman järjestelmässä. Kolmannen osapuolen järjestelmään oli valmiiksi olemassa API-rajapinta, jota automaatiossa oli tarkoitus lähteä hyödyntämään ja samalla selvittää mitä hyötyjä API-rajapintojen käytöstä ohjelmistorobotiikan toteutuksessa voidaan saavuttaa. Aineistoa oli tarkoitus kerätä järjestämällä haastatteluja.

Aineistoa kerätäkseni järjestin kehitysprosessin jälkeen ryhmähaastattelun, johon valikoin mukaan sopivaksi katsomani joukon ihmisiä. Automaatio oli haastattelun aikaan kerennyt olemaan tuotantokäytössä kuukauden ajan. Haastatteluryhmäni muodostui yhdestä business analystista, yhdestä ratkaisuarkkitehdista ja kahdesta ohjelmistokehittäjästä. Heistä jokainen on työskennellyt ohjelmistorobotiikan parissa jo useamman vuoden ajan haastattelun aikaan. Sain aineistooni laajempaa näkemystä ottamalla mukaan erilaisissa työtehtävissä työskenteleviä ihmisiä. Eri tahoilta olisi mahdollista saada työn kannalta merkityksellisiin asioihin hyvin erilaisia vastauksia ja tutkittavaa aihetta pystyttäisiin

tarkastelemaan laajemmalla skaalalla. Haastattelu nauhoitettiin, jotta aineiston käsittely jälkikäteen olisi helpompaa.

Haastattelun jälkeen nauhoitus purettiin ja sitä alettiin analysoimaan. Sisältöä tarkasteltiin useampaan kertaan, jotta saataisiin käsitys siitä, millaisia koodia sitä varten olisi hyvä alkaa rakentamaan. Kun koodit (Liite 1) oli saatu tehtyä niin haastattelussa esiin nousseet erilaiset hyödyt lajiteltiin koodien alle. Tämän avulla pystyttiin hahmottamaan, millaisia hyötyjä API-rajapintoja hyödyntämällä pystytään saavuttamaan.

## 5 ANALYYSI

Toteutetussa RPA-automaatiossa hyödynnettiin sekä API-rajapintaa, että käyttöliittymän päälle rakennettua automaatiota. API-rajapintaa käytettiin kolmannen osapuolen tarjoamaan järjestelmään ja käyttöliittymää käytettiin automaation toteuttaneen tahon omassa järjestelmässä, johon oli tehty jo useita automatisointeja aiemminkin hyödyntäen käyttöliittymää. Seuraavaksi käsittelen eri hyötyjä, joita API-rajapinnan käytöstä nähtiin syntyvän.

### 5.1 Muutosten hallinnan helppous

Haastattelussa nousi esiin API-rajapintojen hyödyntämisen eduista eniten muutosten hallinnan helppous verrattuna käyttöliittymien automatisointiin. Käyttöliittymiin tulevien muutosten on nähty aiheuttavan ongelmia automaatioihin, joissa hyödynnetään puhtaasti käyttöliittymiä. Käyttöliittymiä muutettaessa ei välttämättä tulla miettineeksi sitä, aiheuttaako joku muutos ongelmia ohjelmistorobotin toiminnalle. Tämän vuoksi API-rajapintoja hyödyntäessä muutosten hallinta nähdään helpompana, kun käytettävää järjestelmää käytetään siten kuin sitä on tarkoitettukin käytettävän.

Projektissa tutkimuksen kohteena olleeseen automaatioon oli jouduttu tekemään yksi muutos. Tämän muutoksen toteuttaminen oli kuitenkin sujunut hyvin helposti ja suoraviivaisemmin kuin tilanteessa, jossa olisi jouduttu hakemaan jokin uusi tieto käyttöliittymän kautta. Käyttöliittymän kautta automatisoidessa olisi tarvittu enemmän työtä vastaavanlaisen muutoksen toteuttamiseksi. Nähtiin että API-rajapintojen käytön yhteydessä ylläpito on huomattavasti edullisempaa ja tehokkaampaa.

Nähtiin myös, että uuden henkilön on huomattavasti helpompi saada käsitys automaation toiminnasta silloin, kun käytetään API-rajapintoja käyttöliittymien sijaan:

”Ylläpito on huomattavasti edullisempaa ja tehokkaampaa jos automaation toiminta perustuu API-rajapintoihin käyttöliittymän sijasta.

Ylläpitovaiheessa, koska automaation toiminta on yksinkertaisempaa ja API:n toimintaa on helpompi ymmärtää kuin käyttöliittymää, niin automaation ylläpito helpottuu ja sen omaksuminen on helpompaa.” (Ratkaisuarkkitehti)

Haastattelussa selvisi, että API-rajapintoihin tulevilla muutoksilla on helpompi tietää, onko jollain muutoksella vaikutusta automaation toimintaan. Käyttöliittymiä automatisoidessa voidaan hyödyntää myös joitain ruudulla näkymättömiä tietoja, jolloin voi olla hankala tunnistaa onko jollain muutoksella vaikutusta automaation toimintaan vai ei. Nähtiin myös, että rajapinnat ovat usein taaksepäin yhteensopivia, joten muutokset rajapintoihin eivät yleensä riko automaation toimintaa. Muutokset käyttöliittymiin voivat puolestaan olla hyvinkin arvaamattomia automaation kannalta ja pienikin muutos voi rikkoa koko ohjelmistorobotin toiminnan. Vaikka käyttöliittymään tulisikin jokin muutos, niin vastaavassa tilanteessa API-rajapintaan ei välttämättä tarvitse tehdä minkäänlaista muutosta.

## 5.2 Toteutuksen helppous

Toteutuksen nähtiin olevan helpompaa ratkaisuisissa, joissa hyödynnetään API-rajapintoja käyttöliittymien sijaan. API:n käyttämisen nähtiin olevan paljon modernimpaa ja suoraviivaisempaa. API-rajapintoja hyödyntäessä yhdellä kutsulla voidaan saada aikaan sama asia, jonka toteuttamiseksi käyttöliittymällä pitäisi käydä useassa eri paikassa. Myös varsinainen tekninen toteutus nähtiin helpompana API:n tapauksessa.

”API on suoraviivaisempi ja oikaisee toteutusta. API:n kautta voidaan saada yhdellä kutsulla sama asia, jonka saamiseksi käyttöliittymältä pitäisi siirtyä paikasta ja näkymästä toiseen.” (Ratkaisuarkkitehti)

Varsinaisen toteutuksen lisäksi myös määrittelytyö nähtiin helpompana API-rajapintoja hyödyntäessä kuin käyttöliittymiä automatisoidessa. Käyttöliittymiä automatisoidessa joutuu määrittelemään hyvin tarkasti sen mitä tietoa saadaan mistäkin, mutta API:n tapauksessa määrittely on yksinkertaisempaa, kun järjestelmät on lähtökohtaisesti suunniteltu koneen käyttöön sen sijaan että niitä olisi suunniteltu ihmisen käytettäväksi. Myös määrittelyjen laatu oli API-rajapintojen tapauksessa parempi eikä kehittäjien tarvinnut kysellä niin paljoa tarkentavia kysymyksiä enää kehitystyön aikana.

Määrittelyn pohjana oli helppoa käyttää API-rajapinnan rajapintakuvausta. Tässä tapauksessa käytettävissä oli valmiiksi olemassa oleva hyvä rajapintakuvaus ja se koettiin tärkeäksi osaksi onnistunutta toteutusta. Käyttöliittymiä hyödyntäessä joutuisi kuvaamaan tarkasti mistä mikäkin tieto löytyy, mutta hyvästä rajapintakuvauksesta kaikki automatisoitavan kohteen kannalta tarpeelliset tiedot on helposti löydettävissä.

### 5.3 Luotettavuus

Haastattelussa nousi esiin ohjelmistorobotiikan toteutusten luotettavuus silloin kun hyödynnetään API-rajapintoja käyttöliittymien sijaan. Nähtiin että automaatioita pystytään saamaan toimimaan luotettavammin APIen kanssa kuin käyttöliittymiä hyödyntäen. Haastattelussa kävi myös ilmi, että määrittelyistä saa luotettavampia APIen tapauksessa eikä määrittelyihin jää niin paljoa tulkinnanvaraakaan.

”Kolmannen osapuolen kanssa kaksi vaihtoehtoa: käyttää käyttöliittymää tai integroitua rajapintaan. Kyseisessä prosessissa hyvä API-rajapinta. Järjestelmää käytetään tiedon hakuun. API-rajapinta on luotettavuuden ja tehokkuuden ja toteutuksen helppouden kannalta parempi kuin käyttöliittymän käyttäminen.” (Ratkaisuarkkitehti)

Myöskin nähtiin, että inhimillisiä virheitä ja vastoin prosesseja toimimista pystytään vähentämään, kun käytetään lähinnä koneille suunniteltuja rajapintoja sen sijaan että käytettäisiin ihmisille suunniteltuja käyttöliittymiä. Haastattelun aikaan varsinaisia hyötyjä oli saavutettu vasta vähän, mutta APIen käyttö nähtiin kuitenkin selkeästi luotettavampana vaihtoehtona.

Myös aiemmassa tutkimuksessa Šimek ja Šperka (2019) olivat havainneet, että APIen hyödyntäminen oli luotettavampaa kuin käyttöliittymien käyttäminen. Käyttöliittymiä hyödyntävien automaatioiden luotettavuus on heidän mukaansa pitkälti riippuvainen automatisoitavan järjestelmän luotettavuudesta ja siitä, kuinka varmasti se toimii.

### 5.4 Tehokkuus

API-rajapintojen hyödyntäminen nähtiin tehokkaampana keinona automaatioiden toteuttamiseen kuin käyttöliittymien kautta tapahtuva automaatio. Tehokkuudesta nähtiin olevan kahdenlaisia hyötyjä: kapasiteettia on helpompi hallita ja käyttäjät saavat tarvittaessa tiedot nopeammin. API-rajapintoja käytettäessä tapahtumat saadaan hoitumaan nopeammin kuin tekemällä vastaava toteutus hyödyntäen käyttöliittymiä. Tämä johtaa siihen, ettei APIen tapauksessa tarvitse miettiä ja suunnitella ohjelmistorobotiikan toteutusten osalta tarvittavan kapasiteetin käyttö niin paljoa vaan kapasiteetin hallinta on helpompaa.

”Prosesseissa, joissa käsitellään suuria volyymejä: API:n kautta pystytään paljon nopeammin hakemaan tietoja kuin mennä yksittäin katsomaan jokainen tietue ja siihen liittyvät tiedot. Nopeuttaa huomattavasti, viitaten vähän kapasiteettiin, mutta käyttäjät saavat myös tarvitsemansa tiedot nopeasti.” (Business analyst)

Toteutuksen nopeudella päästään myös siihen, että käyttäjät saavat automaation tuotoksen käyttöönsä nopeammin. Tämä on tärkeää varsinkin sellaisissa tapauksissa, joissa automaation syöte tulee käyttäjiltä itseltään, ja he tavalla tai toisella odottavat automaation valmistumista. Työn kohteena olevassa automaatiossa tällaisesta ei varsinaisesti ollut kyse, mutta haastattelussa nähtiin tämän olevan kuitenkin tärkeää sellaisissa tapauksissa, joissa käyttäjä odottelee automaation valmistumista.

## 5.5 Testauksen helppous

Viimeisenä hyötynä API-rajapintojen käytöstä ohjelmistorobotiikassa nousi esiin testauksen helpottuminen. Jossain tapauksissa käyttöliittymiä hyödyntävissä ohjelmistorobotiikan toteutuksissa testaus voi olla hankalaa. Käyttöliittymille pitää pystyä luomaan monenlaisia erilaisia testitapauksia ja niiden tekeminen voi olla hyvin haastavaa.

”Automaation toiminta on lähes aina APIa käyttäessä yksinkertaisempi: kompleksisuus vähenee ja testaus on sitä myötä helpompaa” (Ratkaisuarkkitehti)

API-rajapintoja hyödyntäessä testauksesta tulee helpompaa kuin käyttöliittymiä käyttäessä. Toteutusten ollessa suoraviivaisempia ja vähemmän kompleksisia tulee testauksestaakin yksinkertaisempaa ja helpompaa.



## 6 POHDINTA JA YHTEENVETO

Aiemmassa kirjallisuudessa ei suoraan annettu vastauksia tutkimuskysymykseeni, mutta siitä saa kuitenkin hyvät lähtökohdat omalle empiiriselle tutkimukselleni. Aiemmassa tutkimuksessa paneuduttiin enemmän ohjelmistorobotiikkaan itsessään tai API-rajapintojen ja ohjelmistorobotiikan asettamiseen vastakkain. Kuitenkin aiheesta pääsi hyvin jatkamaan eteenpäin ja yhdistelemään jo olemassa olevaa tutkimusta omia tarpeitani vastaavaksi. Oli mielenkiintoista päästä miettimään API-rajapintojen hyödyntämistä enemmän osana ohjelmistorobotiikan kokonaisuutta kuin sen vastakohtana. Tällä näkemyksellä päästään paremmin automatisoimaan erilaisia liiketoimintaprosesseja, kun päästään valitsemaan kulloinkin paras mahdollinen teknologia automaation toteuttamiseksi. Mahdollisesti muuten hankalasti automatisoitavissa olevia kohteita päästään automatisoimaan tehokkaammin.

Empiirisen tutkimuksen aion alun perin tehdä haastattelututkimuksena ja olisin haastatellut työkavereitani API-rajapintojen hyödyntämisestä ja niiden mahdollisuuksista, mutta matkan varrella päädyinkin tekemään tapaustutkimuksen aiheesta. Tutkimuksen kohteena on yksinkertainen automaatio, joka toteutettiin osin API-rajapintoja hyödyntämällä ja osin käyttöliittymiä automatisoiden. Tällaiseen toimintamalliin päädyttiin sen takia, koska automaation kohteena oli kaksi järjestelmää, joista toiseen oli hyvä API-rajapinta olemassa, mutta toisen järjestelmän automatisointi jouduttiin toteuttamaan käyttöliittymiä hyödyntäen. Tämä menettelytapa havaittiin toimivaksi tavaksi ja varmasti jatkossa vastaavissa tilanteissa päädytään myös käyttämään helpointa ja tehokkainta käytettävissä olevaa tapaa automaation toteuttamiseksi.

Empiirisessä osuudessa sain kerättyä kokemuksia API-rajapintojen ja käyttöliittymien hyödyntämisestä ohjelmistorobotiikassa ja niiden eri piirteistä. Pääsin vertailemaan menetelmien hyviä ja huonoja puolia ja sain vertailtua näitä kahta keskenään. Ennen tutkimukseni valmistumista olin kiinnostunut säästeystä ajasta ja automaation toimintavarmuudesta tuotantokäytössä. Kuitenkaan tutkimuksen kohteena ollut automaatio ei kerennyt olemaan tuotantokäytössä niin pitkään että näihin asioihin olisi saatu mitattua minkäänlaisia tuloksia. Mielestäni tutkimuksessa nousi esiin kuitenkin muita mielenkiintoisia seikkoja API-

rajapintojen hyödyntämisestä ja varsinkin havaitsemani viisi hyötyä API-rajapintojen hyödyntämisestä oli merkityksellisiä.

Aiemmissa tutkimuksissa oli hyvin vähän vertailtu API-rajapintojen ja käyttöliittymiä hyödyntävän automaation eroja ja erilaisia hyötyjä, joten oma työni toi esiin uutta tietoa, jota ei aiemmin ollut tutkittu. Šimek ja Šperka (2019) olivat tutkimuksessaan havainneet API-rajapintojen hyödyntämisen olevan luotettavampaa kuin vastaavan toteutuksen tekemisen käyttöliittymiä hyödyntäen. Tämä oli oikeastaan ainoa aiemman tutkimuksen esiin nostama asia API-rajapintojen ja käyttöliittymien hyödyntämisen vertailussa. Missään aiemmissa tutkimuksissa ei ollut tuotu esiin neljää muuta havaitsemaani hyötyä: muutoksen hallinnan helppous, toteutuksen helppous, tehokkuus ja testauksen helppous. Ehkäpä API-rajapintojen hyödyntäminen on nähty ohjelmistorobotiikan vastakkaisena toimintana eikä niinkään osana ohjelmistorobotiikkaa, joten näihin asioihin ei ole aiemmassa tutkimuksessa paneuduttu.

Tutkimukseni tuloksia pystyy hyödyntämään käytännössä valitessaan käytettäviä teknologioita liiketoimintaprosessien automatisoinnissa. Varsinaisesti mitään työkalua käytettävän teknologian valintaan en kehittänyt, eikä se ollut tarkoituskaan. Tuloksiani pystyy kuitenkin hyödyntämään pohtiessaan, minkälaisella tavalla jokin prosessi on järkevää automatisoida. Tutkimuksen kentälle tuli uutena tietona erilaiset hyödyt API-rajapintojen hyödyntämisestä, mutta ne hyötyisivät jatkotutkimuksesta, jolloin niistä saataisiin kattavampi ja parempi käsitys.

Tutkimukseni ei ollut täydellinen vaan sillä oli omat rajoitteensa. Graduni rajoittui yhteen yksittäiseen tapaukseen, joten se ei välttämättä kerro koko totuutta aiheesta. Kattavammalla aineistolla olisi luultavasti saanut paremmin yleistettävän tuloksen ja olisi mahdollisesti päässyt havaitsemaan jonkinlaista hajontaa tuloksissa. Keskityin pääasiassa API-rajapintojen hyödyntämisen hyötyihin, enkä koittanut lainkaan selvittää onko API-rajapintojen hyödyntämisellä jonkinlaisia haittoja käyttöliittymien automatisointiin verrattuna. En varsinaisesti koittanut selvittää miten pystyttäisiin tekemään paras mahdollinen valinta käytettävästä automatisointitekniikasta, jos valittavina on sekä API-rajapintojen hyödyntäminen että käyttöliittymien käyttäminen.

Jatkotutkimuksessa olisi mielenkiintoista nähdä kattavamman aineiston käyttöä ja selvittää että päädytäänkö kattavammalla aineistolla samanlaiseen lopputulokseen. Myöskin olisi mielenkiintoista selvittää, onko API-rajapintojen hyödyntämisellä jonkinlaisia haittoja. Olisi myös hyvä kehittää jonkinlainen viitekehys sille, miten voitaisiin parhaalla mahdollisella tavalla valita kulloinkin käytettävä teknologia liiketoimintaprosessin automatisoinnissa. Sopivasta viitekehuksesta voisi olla hyötyä varsinkin käytännön kannalta. Hyperautomaatio on yksi tulevaisuuden tärkeimmistä aiheista liiketoimintaprosessien automatisoinnin saralla ja tutkimuksessa olisikin mielenkiintoista nähdä, mitkä erilaiset automatisointikohteet sopivat eri tekniikoilla toteutettavaksi. Hyperautomaation osalta olisi myös hyvä selvittää miten hankalasti automatisoitavia prosesseja pystyttäisiin automatisoimaan helpommin. Mahdollisimman kattavasti eri

prosessien automatisoinnissa ohjelmistorobotiikalla on kuitenkin suuri merkitys niin käyttöliittymiä kuin API-rajapintoja hyödyntäen.

## LÄHTEET

- Aalst, W., Bichler, M., & Heinzl, A. (2018). Robotic process automation. *Business & Information Systems Engineering*, 60(4), 269-272. doi:10.1007/s12599-018-0542-4
- Agostinelli, S., Marrella, A., & Mecella, M. (2020). Towards intelligent robotic process automation for BPMers. *arXiv preprint arXiv:2001.00804*.
- Aguirre, S., & Rodriguez, A. (2017, September). Automation of a business process using robotic process automation (RPA): A case study. In *Workshop on engineering applications* (pp. 65-71). Springer, Cham.
- Anagnoste, S. (2017, July). Robotic Automation Process-The next major revolution in terms of back office operations improvement. In *Proceedings of the International Conference on Business Excellence* (Vol. 11, No. 1, pp. 676-686). Sciendo.
- Asatiani, A., & Penttinen, E. (2016). Turning robotic process automation into commercial success—Case OpusCapita. *Journal of Information Technology Teaching Cases*, 6(2), 67-74.
- Dumas, M. k., La Rosa, M. k., Mendling, J. k., & Reijers, H. A. k. (2013). *Fundamentals of Business Process Management*. Springer Berlin Heidelberg.
- Hsieh, H. & Shannon, S. E. (2005). Three Approaches to Qualitative Content Analysis. *Qualitative health research*, 15(9), 1277-1288. <https://doi.org/10.1177/1049732305276687>
- IEEE Corporate Advisory Group. (2017). IEEE Guide for Terms and Concepts in Intelligent Process Automation.
- Lasso-Rodriguez, G., & Winkler, K. (2020). Hyperautomation to fulfil jobs rather than executing tasks: the BPM manager robot vs human case. *Romanian Journal of Information Technology and Automatic Control*, 30(3), 7-22.
- Moilanen, J., Niinioja, M., Seppänen, M., & Honkanen, M. (2018). API-talous 101.
- Neumann, A., Laranjeiro, N., & Bernardino, J. (2018). An analysis of public REST web service APIs. *IEEE Transactions on Services Computing*.
- Penttinen, E., Kasslin, H., & Asatiani, A. (2018). *How to choose between robotic process automation and back-end system automation?* Retrieved from [https://explore.openaire.eu/search/other?orpid=&#61;core\\_ac\\_uk\\_:d7dd23770f64ceef6441d6a05f1e909e](https://explore.openaire.eu/search/other?orpid=&#61;core_ac_uk_:d7dd23770f64ceef6441d6a05f1e909e)
- Ray, S., Villa, A., Alexander, M., Guttridge, K., Wang, A. & Vincent, P. (2022) Magic Quadrant for Robotic Process Automation. *Report G00756568*. Gartner

- Ray, S., Vincent, P., Guttridge, K. & Karamouzis, F. (2021) Beyond RPA: Build Your Hyperautomation Technology Portfolio. *Report G00734894. Gartner*
- Šimek, D., & Šperka, R. (2019). How robot/human orchestration can help in an HR department: a case study from a pilot implementation. *Organizacija*, 52(3), 204-217.
- Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J., Ouyang, C., ... & Reijers, H. A. (2020). Robotic process automation: contemporary themes and challenges. *Computers in Industry*, 115, 103162.
- Tornbohm, C., & Dunie, R. (2017). Gartner market guide for robotic process automation software. *Report G00319864. Gartner*
- Vukovic, M., Laredo, J., Muthusamy, V., Slominski, A., Vaculin, R., Tan, W., ... & Branch, J. W. (2016). Riding and thriving on the API hype cycle. *Communications of the ACM*, 59(3), 35-37.
- Willcocks, L. \*, Lacity, M. \*, Andrew, \*, , C., Rights, A. \*, , R. !, . . . Craig, A. (2015). *The outsourcing unit working research paper series the IT function and robotic process automation*
- Welsh de Grimaldo, S., Cana, M. & Nandan, A. (2022) Hype Cycle for Communications Service Provider Operations. *Report G00765745. Gartner*
- Yin, R. K. (2016). *Qualitative research from start to finish* (Second edition.). Guilford Press.

**LIITE 1 SISÄLTÖANALYYSIN KOODIT**

<b>Koodi</b>	<b>Esiintymät</b>
Muutoksen hallinnan helppous	10
Toteutuksen helppous	8
Luotettavuus	6
Tehokkuus	3
Testauksen helppous	1