

# Ohjelmistorobotiikkaselvitys Hämeen ammattikorkeakoululle



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinnan korkeakoulukeskus, Tietojenkäsittelyn koulutusohjelma

Syksy 2019

Asser Heinänen

Tietojenkäsittelyn koulutusohjelma  
Hämeenlinnan korkeakoulukeskus

---

<b>Tekijä</b>	Asser Heinänen	<b>Vuosi</b> 2019
<b>Työn nimi</b>	Ohjelmistorobotiikkaselvitys Hämeen ammattikorkeakoululle	
<b>Työn ohjaaja/t</b>	Lasse Seppänen	

---

## TIIVISTELMÄ

RPA:lla, eli ohjelmistorobotiikalla, haetaan yleensä ratkaisuja toistuviin työtilanteisiin, kuten esimerkiksi sähköposteihin, datan siirtämiseen ja sen ylläpitämiseen. Ohjelmistorobotiikka on ollut jo pitkään olemassa oleva konsepti, mutta se on viime vuosina noussut taas trendikkääksi lisääntyvän työtaakan ja datan vuoksi.

Tämän dokumentin tarkoitus on toimia perustason ohjeena ohjelmistorobotiikkaan, pääsääntöisesti organisaatio-, mutta myös yksikötasolla. Tätä opinnäytetyötä varten selvitettiin myös asioita, joita ohjelmistorobottia rakennettaessa on hyvä ottaa huomioon niin robottia rakentavan tiimin, kuin asiakkaankin. Näitä asioita ovat muun muassa prosessi- ja projektinhallinta, testaus ja yleiset ohjelmistorobotiikan riskit.

Testatut alustat esitellään lyhyesti ja vertaillaan käytännön työn yhteydessä. Vertailussa keskityttiin lähinnä alustojen helppokäyttöisyyteen. Käytännön osuus toteutettiin automatisoimalla yksi Hämeen ammattikorkeakoulun henkilöstöpalveluiden prosessi. Alustojen hinnat eivät ole julkista tietoa, joten niitä ei tähän työhön sisällytetty.

Avainsanat: ohjelmistorobotiikka, automatisointi

**Sivut** 28 sivua

Degree Programme in Business Information Technology  
Hämeenlinna University Centre

---

<b>Author</b>	Asser Heinänen	<b>Year</b> 2019
<b>Subject</b>	A software robotics study for Häme University of Applied Sciences	
<b>Supervisors</b>	Lasse Seppänen	

---

ABSTRACT

RPA, or robotic process automation, generally seeks solutions to repetitive work situations, such as emails, data transfer and maintenance. Robotic process automation has long been a concept but has become trendy again in recent years due to increasing workload and data.

The purpose of this thesis is to serve as a basic guide to software robotics, mainly at the organization, but also unit level. For the purpose of this thesis, matters to consider during the building of a software robot were explored, for both the software construction team and the customer. These include process and project management, testing and general software robotics risks.

The tested platforms are briefly presented and later compared in the practical part of the thesis. The comparison between the platforms mainly focused on the ease of use. The practical part was accomplished by automating one of human resources process at Häme University of Applied Sciences. Platform prices are not public information and were not included in this work.

Keywords: Robotic Process Automation, automatization

**Pages** 28 pages

# Termistö

## Aktiviteetti

Toiminto UiPathissa (ohjelmistorobotiikka-alusta). Aktiviteetit vastaavat Automation Anywheressä (myös ohjelmistorobotiikka-alusta) komentoja (command).

## Automatisaatio

Työtehtävän muuttaminen automaattiseksi. Tämän opinnäytetyön yhteydessä työtehtävän siirtäminen ohjelmistorobotin hoidettavaksi.

## CRPA

Cognitive Robotic Automation. Tekoälyllinen ohjelmistorobotiikka.

## Elementti

Esimerkiksi verkkosivuilla ja ohjelmissa nappi tai tekstikenttä.

## ERP-järjestelmä (Enterprise Resource Planning)

Toiminnanohjausjärjestelmä, jolla integroidaan yrityksen toimintoja. Tällaisia toimintoja ovat muun muassa tuotanto, kirjanpito ja henkilöstöhallinto.

## Iteraatio

Kehityssykli ketterässä ohjelmistokehityksessä.

## Muokkausavain

Muun muassa: vaihto-, control-, alt- ja windows- näppäimet.

## Pikanäppäin

Näppäin tai näppäinyhdistelmä, toteuttaa toiminnon ilman valikon avaamista.

## Rajapinta

Mahdollistaa ohjelmien ja laitteiden välisen tiedonsiirron.

## RPA

Robotic Process Automation (ohjelmistorobotiikka).

## Scrum

Ketterän ohjelmistokehityksen viitekehys.

## Tuoteomistaja

Asiakas tai asiakkaan edustaja.

## Valitsin (selector)

Sisältää tiedon elementin sijainnista ruudulla.

# SISÄLLYS

1	JOHDANTO.....	1
2	OHJELMISTOROBOTIIKKA .....	2
2.1	Automatisoitavat prosessit .....	4
2.2	Projektinhallinta .....	5
2.3	Ohjelmistorobotin testaus .....	6
2.4	Ohjelmistorobotin käyttö.....	7
2.5	Automatisoinnin riskit .....	8
2.6	Ohjelmistorobotin ylläpito .....	9
3	OHJELMISTOROBOTTIEN ESIMERKKEJÄ .....	10
3.1	Telefonica O2 .....	10
3.2	Xchanging .....	11
3.3	Royal DSM .....	12
4	RPA-OHJELMISTOT.....	15
4.1	UiPath.....	15
4.1.1	UiPathin perusteet ja käyttöliittymä .....	16
4.1.2	UiPathin tallennusmenetelmät .....	17
4.2	Automation Anywhere .....	19
4.2.1	Automation Anywheren perusteet ja käyttöliittymä .....	19
4.2.2	Automation Anywheren tallennusmenetelmät ja metatallennukset .....	20
5	ROBOTTIEN VERTAILU .....	21
5.1	Testattava prosessi.....	21
5.2	Eroavaisuudet, heikkoudet ja vahvuudet .....	22
6	YHTEENVETO .....	25
7	LÄHTEET.....	26

## 1 JOHDANTO

Hämeen ammattikorkeakoulussa toteutettiin keväällä 2019 ICT-projektissa opiskelijaryhmällä RPA-pilottiprojekti. Projektilla oli tarkoitus kartoittaa automatisaation mahdollisuuksia koulun henkilöstöpalveluiden prosesseissa. Projektissa käytettiin UiPath-automatisaatio-ohjelmaa. Tämän opinnäytetyöhön otettiin vertailukohteeksi Automation Anywhere käytännön projektiksi.

Tämän opinnäytetyön tarkoitus on käsitellä ohjelmistorobotiikan seikoja, joita Hämeen ammattikorkeakoulussa on hyvä ottaa huomioon RPA projektien toteutuksessa ennen kuin automatisointi pääsee täyteen vauhtiin ja sen aikana. Työn on myös tarkoitus selventää ohjelmistorobotiikan potentiaalia ja realistisia rajoja.

Opinnäytetyön tekijä on ollut kevään 2019 ICT-projektissa kehittäjänä ja myöhemmin loppuvuonna työharjoittelijana ohjelmistokehityksessä. Käytännön tutkimus tässä opinnäytetyössä on suoritettu projektina, jossa automatisoitiin yksi henkilöstöpalveluiden prosessi. Projekti selostetaan läpi ja lopussa käydään läpi tulokset ohjelmien suoriutumista. Tässä opinnäytetyössä ei oteta tarkemmin kantaa ohjelmistosuositukseen, vaan kerrotaan lähinnä kohdatut erot alustojen käytettävyydessä. Ohjelmien hintoja ei myöskään vertailla keskenään, johtuen niiden salassa pidettävyydestä. Työn lopussa vastataan tiivistetysti tutkimuskysymyksiin. Tutkimustyössä molemmilla automatisaatio-ohjelmilla käytettiin ilmaisia community-versioita.

### Tutkimuskysymykset

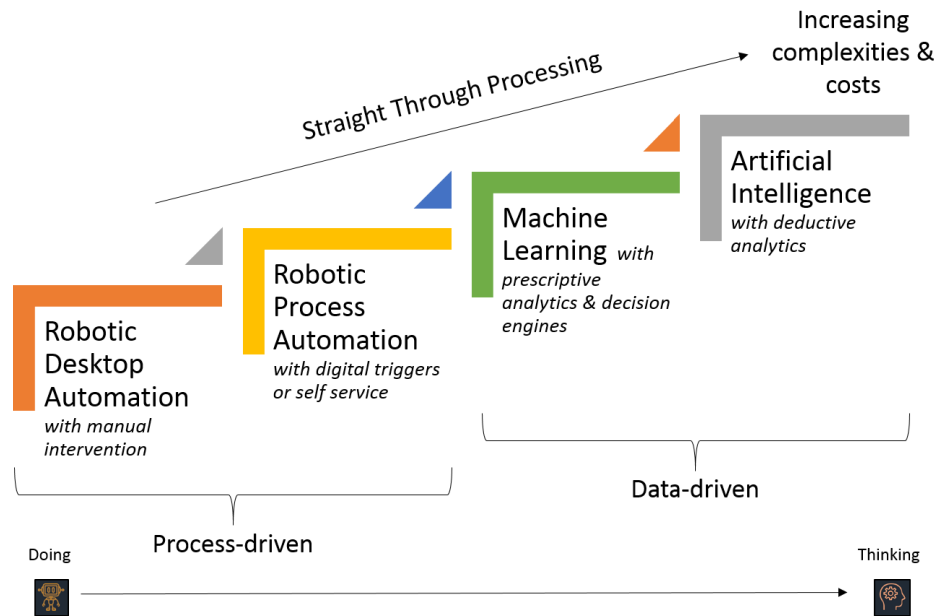
- Minkälaisiin prosesseihin RPA soveltuu yleisesti ja Hämeen ammattikorkeakoulussa?
- Mitä ohjelmistorobotiikka vaatii toteutuakseen?
- Kuinka käyttäjäystävällisiä testattavat alustat ovat?

## 2 OHJELMISTOROBOTIIKKA

Ohjelmistorobotiikka, eli RPA, on ollut olemassa terminä jo viimeiset 20 vuotta. 2000-luvun ensimmäisen vuosikymmenen aikana yritykset alkoivat huomata RPA:n edut toistettavien tehtävien automatisoinnissa. (Roboticprocess365.com, 2018) Nyt ohjelmistorobotiikka on noussut uudestaan puheenaiheeksi tehokkuuden ja kasvavien datamäärien käsittelyn tavoittelussa. IT-alan oma ”teollinen vallankumous” onkin ollut kiihtymässä jo jonkin aikaa. Tässä luvussa esitellään mitä ohjelmistorobotiikka on, sekä mihin se yleisesti sopii.

RPA on käsitteenä melko yksinkertainen. Robotic-osa on kokonaisuus, joka voidaan tietokoneen avulla ohjelmoida tekemään erinäisiä tehtäviä ja jonka tarkoituksena on matkia ihmisen tekemistä. Process-osa koostuu vaiheista, joilla koko tapahtuma saadaan aikaiseksi. Tästä esimerkkinä kirjautuminen sähköpostiin: käyttäjän syöttäminen, salasanan syöttäminen ja kirjautuminen. Näin on määritelty kolmiosainen prosessi. Automation-osa toteutuu, kun jokin tehtävä tapahtuu ilman ihmisen osallistumista prosessiin. Näistä kolmesta osasta koostuu ohjelmistorobotiikka. (Edureka, 2019)

Ohjelmistorobotteja on joko normaalisti koodin mukaan toimivia tai tekoälyllä avustettuja. Perinteisempää ohjelmistorobottia käytetään samalla tavalla toistuviin prosesseihin. Tekoälyllä avustetulla ohjelmalla puolestaan on enemmän joustoa tässä suhteessa. Tekoälyllä varustettu robotti voi oppia tekemistään prosesseista toiston avulla, tätä kutsutaan kone- tai syväoppimiseksi. Yleensä tekoälystä puhuttaessa tarkoitetaan ohjelmia, jotka pystyvät myös ylikirjoittamaan itseään ja täten mukautumaan paremmin prosessiin. Nykyään termi Kognitiivinen RPA (Cognitive RPA/CRPA) on yleistynyt IT-maailmassa tekoälyllisille ohjelmille. (SkyMind, n.d.) Kuvassa 1 on tekoälyn kirjo, jossa tekevällä puolella (vasemmalla) ovat perinteisemmät RPA-ratkaisut ja ajattelevalla (oikealla) puolella koneoppiminen ja tekoäly sellaisena kuin ne yleisemmin mielletään.



Kuva 1. Tekoäly kirjo (CFB Bots, 2018)

Robottia suunnitellessa on hyvä huomioida vaikeutuva ylläpito ja päivittäminen, mitä joustavammin robotti tehdään. Myös järjestelmän muuttaminen, jossa robotti toimii, vaikuttaa päätökseen. Nopeasti muuttuviin järjestelmiin ei välttämättä ole järkevää kehittää tekoälyllä varustettua robottia, koska koko robotti tai osia siitä voidaan joutua tekemään uudestaan, jos muutos on tarpeeksi suuri.

Ohjelmistorobotiikan hyödyntäminen sopii parhaiten prosesseihin, joita tapahtuu paljon ja joissa on korkean tason standardointi. Ohjelmistorobotiikka ei sovellu hyvin tilanteisiin, joissa prosessien syy-seuraus-suhde on heikosti havaittavissa. Monimutkaisiakin prosesseja voidaan automatisoida, jos eri prosessin vaiheet on tarkoin määriteltä ja niiden muuttujista on kontrolli. Ohjelmistoroboteilla on etuna myös toiminta lähes millä tahansa alustalla, kunhan robotti pääsee käsiksi näkymään, jonka käyttäjä näkee päätelaitteellaan. (Lacity, Willcocks, & Craig, 2015, s.9-10)

Aiemmin 2019 keväällä Hämeen ammattikorkeakoululla toteutetusta pilotiprojektista selvisi myöskin, että alustat, joissa on paljon dynaamisesti päivittyvää sisältöä, ovat suhteellisen vaikeita automatisoinnin kohteita. Projektissa kohdatut latausanimaatiot esimerkiksi aiheuttivat usein tilanteita, joissa automatisaatio-ohjelma ymmärtää verkkosivun olevan käyttövalmis, mutta käytännössä ruudulla pyörii latausanimaatio, joka estää elementtien klikkaukset. Tämä johti robotin prosessin kaatumiseen.

Ohjelmistorobotiikan paras puoli, käyttöliittymän hyödyntäminen, on helposti haavoittuva. Jos käyttöliittymä muuttuu, se voi rikkoa robotin toiminnan kokonaan. Pienikin muutos voi johtaa hajonneeseen prosessiin. (Bloomberg, 2018) Robotin sietoa muutoksille pystytään kuitenkin lisäämään jonkin verran hyödyntämällä automatisoitavan kohteen rajapintaa



mahdollisimman paljon. Robotin on mahdollista käyttää dynaamisesti päivitettäviäkin sivustoja, kuten aiemmin mainitussa projektissakin, jos niistä saadaan yksilöivää dataa, jolla robotti esimerkiksi löytää tietyn elementin verkkosivulta.

Esimerkiksi palkanmaksu- tai työajanseurantajärjestelmät toimivat hyvin ohjelmistorobotiikan kanssa yhteen. Näiden järjestelmien on oltava hyvin testattuja ja stabiileja, jottei asiakkaille aiheudu potentiaalisia kuluja, jos järjestelmä ei toimi. Järjestelmää ei yleensä voi muuttaa jälkikäteen paljoo, koska sen seurauksena pitäisi suorittaa uudestaan kattava testaaminen. Mitä vähemmän nämä robotin työympäristöt muuttuvat, sitä kannattavampaa se on robotin toimintavarmuuden ja ylläpidon kannalta.

Työntekijöille puolestaan ohjelmistorobotiikka tarkoittaa tylsien ja toistuvien työtehtävien vähenemistä, jättäen enemmän aikaa työntekijöille tehdä merkityksellisimpiä tehtäviä, joissa robottia ei voi hyödyntää. Samaan aikaan robotti kykenee tekemään toistuvia töitä useankin ihmistyöntekijän edestä. Robotin konfigurointi ei välttämättä vaadi syvällistä osaamista koodaamisesta, sillä monessa ohjelmistorobotteja tarjoavissa työkaluissa on käytössä helppokäyttöisiä graafisia käyttöliittymiä. Vaadittava koodaamisen ymmärtäminen riippuu myös prosessin monimutkaisuudesta. Automatisaatio-ohjelmistojen opetteluun tietenkin kuluu aina oma aikansa.

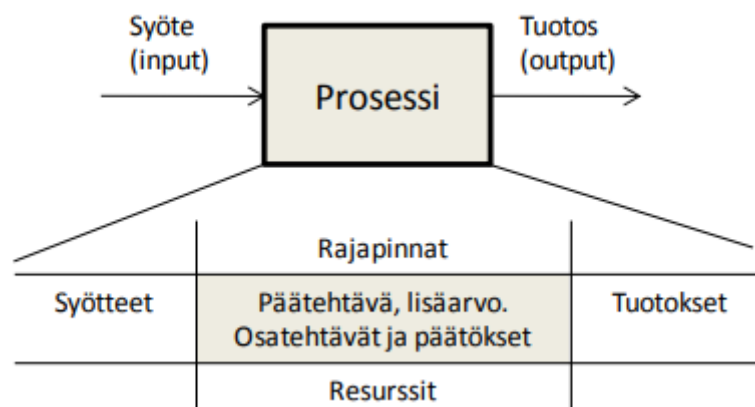
## 2.1 Automatisoitavat prosessit

Prosessin tarkka määrittäminen on yksi onnistuneen projektin kulmakiviä. Jotta ohjelmistorobotiikkaa voidaan hyödyntää järkevästi ja tehokkaasti, on automatisoitavan prosessin vaiheet tunnettava tarkasti.

”Prosessit ovat asiakkaalle lisäarvoa luovia tapahtumaketjuja, joihin yritys käyttää resursseja” (Martinsuo & Blomqvist, 2010, s.4). Esimerkiksi Kuvan 2 tapahtumaketju voisi kuvastaa rekrytointia: koulu lisää rekrytointi-ilmoituksen (syöte), saa ehdokkaita, karsii haastatteluilla sopivat kandidaatit (prosessi) ja rekrytoi tai ei rekrytoi työntekijää (tuotos). Sama asetelma toimii myös itse robotin kohdalla: työntekijä syöttää robotille vaadittavan datan, robotti suorittaa prosessin ja tuloksena on joko onnistunut tai epäonnistunut tuotos.

Prosessien mallintamisessa on tärkeää aloittaa ydinprosesseista. Nämä prosessit määrittelevät, mikä on kokonaiskuvan kannalta arvokkaimpia ja kriittisimpiä. Näistä prosesseista rakennetaan ensin karkea kuvaus. Karkeaan kuvaukseen sisällytetään prosessin vaiheet, tärkeimmät päätökset, vaadittavat syötteet, tuotokset ja käytettävät järjestelmät, kuten kuvan 2 alaosassa on esitetty. (Martinsuo & Blomqvist, 2010, s.9) Ohjelmistorobotin kannalta olisi myös oltava selvillä erinäisistä järjestelmä- ja

tietoliikenneoikeuksista ja -rajoituksista, jos mahdollista, jotta niihin osataan varautua prosessin toteutuksessa.



Kuva 2. Ydinprosessi (Martinsuo & Blomqvist, 2010, s.10)

Jos jokin ydinprosessi sisältää monimutkaisia vaiheita tai on laaja, on suositeltavaa tehdä yksityiskohtainen kuvaus karkean lisäksi. Näitä tilanteita varten käytetään monenlaisia vuokaavioita. (Martinsuo & Blomqvist, 2010, s.10-11) Yksityiskohtaisella kuvauksella voi olla suurikin vaikutus siihen, miten ohjelmistoa lähdetään rakentamaan tai miten hyvin projekti-tiimi ymmärtää prosessin. Kuvauksiin on hyvä sisällyttää itse prosessin vaiheen lisäksi myös järjestelmä, missä se suoritetaan, jos järjestelmiä on prosessissa useita.

## 2.2 Projektinhallinta

Nykyään ohjelmistokehityksen projekteissa suositaan ketteriä menetelmiä. Ketterä ohjelmistokehitys rakentuu neljän arvon ympärille: yksilöt ja kanssakäyminen, toimivan ohjelman priorisointi, asiakasyhteistyö, sekä muutokseen reagointi. Nämä neljä arvoa ovat kuitenkin vain viitekehys ja niitä on suotavaa soveltaa tarpeen mukaan.

Yksilökohtaisessa kehityksessä yhteydenpidon ja muutosten ajaminen on sujuvaa, sillä se ei vaadi tarkkaa aikataulutusta ja tapahtuu tarpeen mukaan. Tarkoituksena on, että käytettävät menetelmät ja järjestelmät tukevat yksilöitä, ei toisin päin. Yksilökeskeinen kehittäminen vaatii kuitenkin työntekijöiden kohdalla enemmän vastuunkantoa ja omatoimisempaa suunnittelua työtehtävistään. (Lehtonen et al., 2014; Smartsheet, n.d.; Software Testing Fundamentals, n.d.)

Toimivan ohjelmiston aikaansaaminen on pääprioriteetti ja dokumentaation tarkoitus on lähinnä selventää ja ylläpitää kokonaiskuvaa ohjelmiston kehityksestä. Liiallinen dokumentointi on yleensä pois tuotteen kehittämisajasta. Dokumentointia ei kuitenkaan voi kokonaan eliminoida. Tämä korostuu tilanteissa, joissa projektin jäsen lähtee tai vaihtuu ryhmässä. (Lehtonen et al., 2014; Smartsheet, n.d.; Software Testing Fundamentals, n.d.) Myös poissaolojen kannalta on perustason dokumentaation hyvä

olla kunnossa, jotta ryhmän jäsenten on helpompi pysyä kehityksen mukana.

Vaikka osallistuminen kehitystyöhön rasittaaakin asiakkaan puolta, on se erittäin tärkeää, jotta lopputuotoksesta saadaan tehtyä sellainen kuin siitä pitäisi tulla. Asiakkaan ensisijainen tarkoitus on antaa palautetta toimivista esittelyversioista, joita ketterässä kehityksessä toimitetaan sovituin väliajoin. Myös asiakkaan osallistuminen tiettyihin ryhmän kokouksiin ja palautteen antaminen niissä on mahdollista. Näin kehitystä saadaan muutettua palautteen mukaan silloin kuin sitä tarvitaan. Asiakkaan ei ole kuitenkaan lähtökohtaisesti tarkoitus osallistua jokaiseen ryhmän kokoukseen. (Lehtonen et al., 2014; Smartsheet, n.d.; Software Testing Fundamentals, n.d.) Asiakkaan puolelta on myös saatava projekteihin mukaan prosessin hyvin tunteva henkilö. Tämä korostuu RPA-projekteissa, kun kehittäjät tarvitsevat selvityksen prosessista, jota he eivät tunne. Kehittäjät pystyvät tällöin kommunikoimaan kyseisen henkilön kanssa automatisoitavan prosessin eri vaiheista, sekä mahdollisista ongelmista automatisoinnin kannalta. Prosessin tuntevilla henkilöillä saattaa olla myös tietoa prosessin poikkeuksista tai käytännöistä, joita ei välttämättä löydy dokumentaatiosta tai tule esille prosessia kartoittaessa.

Muutoksiin on myös kyettävä reagoimaan nopeasti. Tämän vuoksi suunnitelmien tekemisen pitäisi rajoittua vain siihen, mitä tarvitaan seuraavaksi tuotteen kehittämisessä. Tämä kattaisi ketterässä kehityksessä yleensä nykyisen tai seuraavan iteraation. Yksityiskohtainen suunnittelu pitkille aikaväleille ei ketterässä kehittämisessä ole suotavaa mahdollisten muutosten takia, jolloin suunnittelu menee hukkaan. (Lehtonen et al., 2014; Smartsheet, n.d.; Software Testing Fundamentals, n.d.)

### 2.3 Ohjelmistorobotin testaus

RPA-projekteja voi olla vaikea testata, sillä RPA itsessään usein rajoittaa testaamistapoja ja -käytäntöjä. Tähän ongelmaan vastataan seuraavilla esiteltävillä tavoilla.

Ketterässä ohjelmistokehityksessä käytetään käyttäjätarinoita. Käyttäjätarina vastaa kysymyksiin mitä ja miksi jokin asia tehdään ohjelmistolla. Näiden tarinoiden ylläpidosta projektissa vastaa yleensä tuoteomistaja. Tuoteomistajana toimii asiakas tai henkilö, joka edustaa asiakasta. Tuoteomistajan ei tarvitse käyttäjätarinoiden kannalta olla selvillä ohjelmiston teknisestä toteutuksesta, vaan vain sen halutuista toiminnoista. (Rose, 2017) Esimerkiksi ”Käyttäjänä haluan pystyä syöttämään tunnukseni kenttiin x ja y, jotta voin kirjautua sivulle z” on käyttäjätarina.

Tarkoituksena olisi myös pitää tarinat tarpeeksi pieninä, jotta niitä on helppo käyttää testauksen yhteydessä vaatimuksina. Testatessa katsotaan lähinnä sitä, tekeekö ohjelma sitä mitä sen pitäisi, kun syöte on

oikein tai väärin. Jos syöte ja ohjelman logiikka on oikein, pitäisi prosessin silloin edetä. Puolestaan virheilmoitukset on testattava esimerkiksi väärällä syötteellä. Myös tilanteisiin, joissa ohjelma ei löydä kohde-elementtiä tai ei pysty tekemään toimintoa siihen, olisi hyvä olla tarkka ilmoitus elementin nimestä ja mahdollisesti sijainnista automatisaatio-ohjelmassa.

Automaatioprosessille pitäisi olla myös mielellään kolme eri testiympäristöä: kehitysympäristö, tuotantoympäristö ja kopio tuotantoympäristöstä. Jokaisen testikäytössä olevan tietokoneen olisi myös hyvä olla samanlainen kuin tuotantokäytössä oleva kone. (Odess-Gillett, 2018) Näistä kolmesta tuotantoympäristön kopio on merkitykseltään pienin, riippuen projektin koosta. Suurissa projekteissa tuotantoympäristön kopiolle löytyy paljon käyttöä muun muassa erinäisten elementtien tunnistamisessa automatisaatio-ohjelmassa, sillä ohjelmien ja verkkosivujen elementtien nimet muuttuvat yleensä ympäristön mukaan. Jos tuotantoympäristön kopio on käytettävissä, säästää se päänvaivaa ohjelman kehittämisessä käyttövalmiiksi, kun elementtien nimet eivät muutu enää tuotantokäyttöön. Tuotantoympäristössä saattaa olla kuitenkin joitain elementtejä tai toimintoja, jotka eroavat tai puuttuvat tuotantoympäristön kopiosta. Tämän vuoksi automatisaatio-ohjelma on testattava myös erikseen tuotantoympäristössä, vaikka siitä olisikin kopioympäristö.

## 2.4 Ohjelmistorobotin käyttö

Valmiin ohjelmistorobotin käyttö riippuu paljolti käytettävästä lisenssistä maksullisilla alustoilla. Tähän työhön valitut alustat, UiPath ja Automation Anywhere, esitellään tarkemmin luvussa 4. UiPathin Orchestrator- ja Automation Anywheren Control Room-pilvipalveluilla robotteja kyetään hallinnoimaan, ajamaan ja ajastamaan etänä, kun taas paikalliset lisenssit vaativat omat ratkaisunsa. Työkoneella paikallisesti sijaitseva ohjelmistorobotti laukaistaan joko suoraan automaatio-ohjelmasta tai esimerkiksi .exe-tiedoston kautta. Myös jonkin toisen ohjelman kautta käynnistys on mahdollista, jos käytettävä lisenssi sen sallii.

Ohjelmistorobotilla ajettaviin ohjelmiin tarvittavat tunnukset, salasanat ja muut tarvittavat tiedot robotin ajoa varten voidaan puolestaan kerätä esimerkiksi käyttäjän syötteenä ohjelman käynnistyessä tai alustaa ohjelmistorobotissa itsessään. Tiedot voidaan hakea myös tiedostosta, jonka ohjelmistorobotti lukee käynnistyessään. Prosesseja, joissa ohjelma kerää tiedot käyttäjältä, olisi kuitenkin hyvä mahdollisuuksien mukaan kehittää mahdollisimman vähän työntekijöiden syötteen varaan, jotta automatisoinnista hyödytään enemmän.

Kummallakin alustalla on omat ratkaisunsa turvalliseen ohjelmien sisäiseen salasanojen säilyttämiseen. UiPathissa on erillinen aktiviteetti salasanojen alustamiseen ja Automation Anywherellä on oma muuttujatyyppi

kirjautumistiedoille. Näin vältetään salasanojen säilyttäminen luettavassa muodossa. Tiedostosta tunnuksia lukiessa on myös otettava tietoturva huomioon. Tällöin on suunniteltava tiedoston suojaus ja automatisaatioalustan työnkulku tiedoston käsittelyssä, jotta tunnukset eivät ole luettavissa tiedostosta ulkopuolisille.

## 2.5 Automatisoinnin riskit

Ohjelmistorobotiikkaan liittyy monia riskejä. Aiemmissä luvuissa on muutamia jo mainittukin, mutta tässä luvussa käsitellään niitä kootusti.

Automatisoinnin alussa vastaan tulevia ongelmia ovat muun muassa: lopputavoitteiden puuttuminen, osaamaton henkilöstö ja sen puute, sekä väärin tai tarkoitukseen soveltumattomien alustojen ja ratkaisujen käyttö. Automatisaatioalustoja on tarjolla monia, mutta organisaation tarpeisiin sopivaa alustaa voi olla vaikea arvioida. Organisaation ulkoista neuvontaa on suositeltavaa hyödyntää tarvittaessa tässä vaiheessa. (Liao, 2019)

Tarkoitus ei ole hankkia ominaisuuksiltaan liian kapeaa tai liian laajaa alustaa, ellei kyseessä ole vapaan lähdekoodin ratkaisu. Tällöin laajasta työkaluvalikoimasta ei aiheudu taloudellista ongelmaa. Tämä kuitenkin riippuu alustan itsensä käyttövaikeudesta, josta usein seuraa kustannuksia palkattavassa työvoimassa. Maksullisissa alustoissa robotin rakentaminen on usein helpompaa ja avustetumpaa, mutta kulut voivat nousta äkkiä lisenssitarpeen mukaan. Ennen ostopäätöstä on siis hyvä tarkkaan selvittää tuotteen käyttörajat ja ominaisuudet, sekä niiden tarpeellisuus. Osaamaton henkilöstö puolestaan ei pysty hyödyntämään alustaa tarkoituksenmukaisesti, saati suunnittelemaan sillä tarpeeksi kattavaa ohjelmaa. Myös henkilöstön riittävydestä kehitys- ja ylläpitotyöhön on hyvä olla selvillä. Uusia automatisaation kohteita on mahdotonta hyödyntää, jos käytössä oleva henkilöstö on jatkuvasti ylläpitämässä edellisiä robotteja.

Automatisoinnin alettua tulosten näkyminen, testauksen ja dokumentaation puutteet, sekä ongelmat automatisaation tehokkaassa hyödyntämisessä ovat seuraavia riskejä. Selkeitä tuloksia pystytään laskemaan esimerkiksi siitä, kuinka paljon aikaa ja rahaa jonkin prosessin automatisointi säästää suhteessa ohjelman lisenssiin, mutta epäsuoria vaikutuksia on paljon vaikeampi arvioida. Kuinka paljon automatisaation vapauttama työntekijä pystyy hyödyntämään prosessissa säästettyä aikaa tai kuinka paljon se henkisesti helpottaa työpäivää? Näitä vaikutuksia ei pystytä suoranaisesti arvioimaan. Testauksen ja dokumentaation puutteet puolestaan voivat esimerkiksi viivästyttää ohjelmiston käyttöönottoa tai muodostua sen jälkeen ongelmaksi. Muun muassa ohjelmassa käytettävät muuttujat ja niiden käyttöyhteydet on huomattavasti helpompi ymmärtää, jos ne on dokumentoitu huolellisesti.

Organisaation yksiköiden ”siiloittuminen” on myös ongelma. Jos automatisaatiota ajetaan yksikkökohtaisesti, ei tällöin synny organisaatiotason ratkaisua. Näin jokainen ratkaisu alkaa elää omassa kuplassaan, ilman keskitettyä työryhmää, joka hoitaisi kehitys- ja ylläpitotyötä. (Fersht, Gupta, & Christopher, 2019) Pahimmillaan tästä seuraa turhia lisäkuluja, kun kehitetään samanaikaisesti eri alustoilla, sekä ylläpidetään erillisiä ylläpitäjiä.

## 2.6 Ohjelmistorobotin ylläpito

Jos automatisoitava järjestelmä muuttuu, on ohjelmistorobottiakin päivitettävä. Mikäli muutos on pieni ja se esimerkiksi muuttaa kenttien sijaintia ohjelmassa, ei ohjelmistorobottia välttämättä tarvitse muuttaa millään tavalla. Jos puolestaan kenttien nimet muuttuvat tai niiden muoto muuttuu, on tarvetta usein päivitykselle. Muodon muuttumisesta tässä yhteydessä tarkoitetaan esimerkiksi tekstisyötekentän muuttumista pudentusvalikoksi. Tässä tapauksessa robottia on pakko muuttaa, koska tietoa ei saada enää samoilla komennoilla kentistä. Kenttien nimien muuttuessa myös niiden valitsimet täytyy päivittää, jotta ohjelmistorobotti löytää kohde-elementin.

Kenttien muuttuminen välttämättä aiheuta kuitenkin suuria muutoksia robottiin. Kyse on silloin lähinnä komentojen muuttamisesta prosessissa, mikä ei itsessään vaadi paljoa työtä. Tällaisissa tilanteissa muutoksella ei ole kokonaiskuvan kannalta suurta vaikutusta, kun prosessin vaiheiden järjestys pysyy kutakuinkin samana. Kuitenkin tapauksissa, joissa kentät ovat lähinnä ihmisille käyttäjäystävällisiä, voi muodostua suurempia ongelmia. Esimerkiksi ihmiskäyttäjälle mukavalta tuntuva kalenteri tai liukuvalikko syötekentässä saattaa vaatia paljonkin työtä robotilla, jos kenttään ei ole mahdollista laittaa syötettä tekstinä. Tällöin joudutaan mahdollisesti käyttämään tekoälyä apuna tai kehittämään muita tapoja kiertää ongelma.

### 3 OHJELMISTOROBOTTIEN ESIMERKKEJÄ

Tässä luvussa esitellään kahta esimerkkitapausta RPA:n implementoinnista ja niiden tuloksia. Myös syitä automatisaation harkitsemiseen käydään läpi.

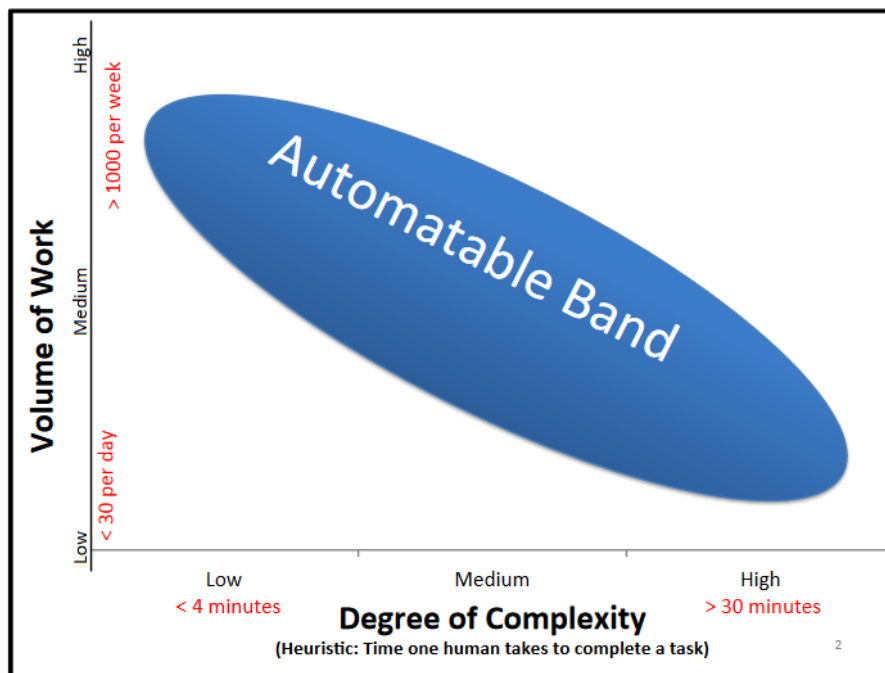
#### 3.1 Telefonica O2

Telefonica O2 on Iso-Britannialainen telekommunikaatioalan yritys. Se työllisti vuonna 2013 yli 20 000 ihmistä ja vuonna 2015 Telefonicalla oli 24 miljoonaa asiakasta globaalisti. (M. Lacity et al., 2015)

Telefonicalla oli kokonaisuudessaan pääprosesseja 60 ja sivuprosesseja noin 400. Näistä automatisoitiin 15 pääprosessia, joiden kuukausittaiset toistomäärät olivat noin 400 000:sta 500 000 kappaleeseen. Valittuihin prosesseihin lukeutui muun muassa tilausten ja puheluiden käsittelyä. Yritys implementoi yli 160 ohjelmistorobottia, joiden arvioitu takaisinmaksuaika oli vuosi ja investoinnin tuotto kolmessa vuodessa noin 650 prosentista 800 prosenttiin. (M. Lacity et al., 2015)

Automatisaatiota lähdettiin yrityksessä harkitsemaan, kun Intiaan siirretyn työn hinta alkoi kasvaa työntekijöiden palkkatason noustessa. Työmäärän jatkuva lisääntyminen ja nousevat palkkakustannukset aiheuttivat yritykselle kestävämmän tilanteen pitkällä aikavälillä. Työn siirtovaiheessa Intiaan osastosta oli pudotettu britannialaisten määrää 98:sta 50 työntekijään. Samalla intialaisten työntekijöiden määrä kasvoi 200:sta 375 työntekijään. Automatisaation jälkeen osastolla työskenteli edelleen noin 250 kokoaikaista intialaista. (M. Lacity et al., 2015)

Kuvassa 3 näkyy, millaisille määrille ja laajuuksille RPA:n implementointi oli kannattavaa Telefonica O2:ssa. Telefonican toteuttama RPA-ratkaisu tuotti hyviä tuloksia. Alan pioneerina yritys joutui kuitenkin toteamaan, että tämän tapauksen prosesseissa, joissa RPA:lla olisi pystytty korvaamaan kolme tai vähemmän työntekijää, ei ollut tarpeeksi kannattavaa. Ohjelmistorobotin implementointi ja ylläpito vaatii aina tietyn tasoisin investoinnin, eikä sen implementointi ole näin kannattavaa, jos robotti ei ehdi maksaa itseään takaisin ennen kuin järjestelmä muuttuu tai vaihtuu. (M. Lacity et al., 2015)



Kuva 3. RPA:n hyödyntämisen rajoja Telefonica O2:ssa (Lacity et al., 2015, s.11)

### 3.2 Xchanging

Xchanging on myös Iso-Britannialainen globaalisti toimiva yritys. Yritys tarjoaa IT- ja teknologiapalveluita. Nämä palvelut kattavat myös bisnesprosessien teknologisoinnin useilla eri aloilla. Vuonna 2014 yrityksessä työskenteli yli 7400 työntekijää. (Willcocks, Lacity, & Craig, 2015)

Automatisoituja prosesseja Xchanging:lla oli 14 ja ohjelmistorobotteja näihin prosesseihin implementoitiin 27. Kohdeprosessit keskittyivät vaakuutusdokumenttien käsittelyyn ja erinäisten raporttien luontiin. Automatisoinnilla saatiin karsittua manuaalisen tiedonsiirron määrää järjestelmien ja dokumenttien välillä. Prosessien suorituskertoja oli kuukausittain noin 120 000. Automatisoinnilla saatiin keskimäärin 30 prosentin säästö per prosessi, eikä tämä vaatinut työntekijöiden irtisanomisia, vaan automatisoinnilla lisättiin nykyisten työntekijöiden tuottavuutta. Lopputuloksena projekteja saatiin nopeammin toteutettua, sekä toimintatarkkuus parani. Automatisoitujen prosessien tarkkuus kohosi lopulta 93 prosenttiin, vaikka tavoitteena olikin vain 80 prosenttia. (Willcocks et al., 2015)

Xchanging:n automatisointia kohden johti osittain 2011 vuonna koittanut tulosvaroitus, jolloin yrityksen tuloksen ennustettiin jäävän heikolle tasolle. Samana vuonna yrityksen toimitusjohtaja vaihtui. Säästöihin oli tarvetta ja yrityksen tarkoitus oli pyrkiä parantamaan toimintaansa nopeamaksi ja kustannustehokkaammaksi. Näihin ongelmiin sopi ohjelmistorobotiikka. Vuoden 2013 lopussa yritys aloitti sopivien prosessien kartoittamisen. Xchanging:lla oli aluksi vaikeuksia tunnistaa automatisointiin



sopivia prosesseja, mutta lopulta päädyttiin aiemmin mainittuihin yksinkertaisiin, mutta suuriin työmääriin keskittyviin prosesseihin, kuten tiedonsiirtoon. Xcanging toteutti vuoden 2014 aikana testitapaukset, koulutuksen ja ensimmäisten 10 ohjelmistorobotin implementoinnin. Vuoden 2015 alusta eteenpäin yritys oli tehnyt jatkokehitystä ja toteuttanut loput 17 ohjelmistorobottia saman vuoden kuluessa. (Willcocks et al., 2015)

Mutkitta automatisointi ei kuitenkaan edennyt. IT-osaston kanssa alkoi muodostua ongelmia: "There were a lot of skeptics in the technology space; it took a lot of convincing to allow business based operations to take some form of control over what is a decent sized IT change initiative, and a different way of operating for us as an organization" (Willcocks et al., 2015, s. 9). Ongelma ratkaistiin muodostamalla RPA-ryhmä, johon kuului 20 työntekijää. Tästä ryhmästä puolet toimi vakuutusasioiden parissa ja toinen puolikas lähinnä IT-puolen tehtävissä. Tämä muodostettu ryhmä toteutti ensimmäiset testitapaukset. (Willcocks et al., 2015)

Tärkeimmiksi syiksi Xchanging:n onnistumiselle automatisoinnissa nousi kolme syytä. Yleinen automatisaation hyväksyntä työyhteisössä on näistä ensimmäinen ja laajentamisen kannalta tärkein. Toisena on IT-osaston mukana olo, joka on tärkeä osa laajentamisen mahdollistamista ja sen hallinnointia. Viimeisenä syynä on yrityksen sisäisen henkilökunnan RPA-osaamisen taso. (Willcocks et al., 2015)

### 3.3 Royal DSM

Royal DSM on Alankomaissa toimiva monikansallinen yritys. Vuonna 2015 se työllisti noin 20 000 henkeä. Yrityksen tuotteisiin lukeutuu muun muassa vitamiineja, mineraaleja, biomuoveja ja -polttoaineita. (M. C. Lacity, Willcocks, & Craig, 2016)

Automatisoitavia prosesseja Royal DSM:lle kertyi 44 kappaletta, jotka implementoitiin kahdessa eri syklissä. Molemmat syklit kestivät noin puoli vuotta. Ensimmäisessä syklissä implementoitiin 19 ohjelmistorobottia ja loput 25 seuraavassa syklissä. Prosessien vaiheista pystyttiin automatisoimaan 89 prosenttia. Lopputuloksena prosessit pystyttiin käymään läpi kolmessa päivässä aiemman kahden viikon sijasta. Myös toimintatarkkuus parani. Automatisaation kulut saatiin katettua yhdeksässä kuukaudessa. (M. C. Lacity et al., 2016)

Yrityksen ohjelmistorobottien äärelle ajoi pitkälti sama ongelma kuin kahdella aiemmalla esimerkkiyrityksellä: kulut nousivat työmäärän kasvaessa, mutta kustannustehokkuutta oli kyettävä parantamaan. Royal DSM oli kyennyt vastaamaan näihin ongelmiin jo muilla keinoilla tietotekniikan ja henkilöstöpalveluiden yksiköissään. Seuraavana listalla oli rahoituspalveluiden yksikkö. (M. C. Lacity et al., 2016)

Royal DSM perusti vuonna 2012 jaettujen talouspalveluiden yksikön (Financial Shared Services). Yksikön oli tarkoitus yhdistää yrityksen eri bisnesyksiköt itseensä. Aikatavoite tälle asetettiin neljän ja viiden vuoden välille, johtuen yrityksen käyttämistä monista eri ERP-järjestelmistä. Yhtä yksittäistä ERP-alustaa ei kuitenkaan haluttu valjastaa kaikkien yksiköiden käytettäväksi, vaan ratkaisussa haluttiin hyödyntää jo käytössä olevia järjestelmiä. (M. C. Lacity et al., 2016) Tämä on ymmärrettävää esimerkiksi maakohtaisten rajoitteiden ja lakien kannalta. Yhden ison alustan olisi otettava huomioon usean eri maan käytännöt huomioon samaan aikaan.

Jaettujen talouspalveluiden johto alkoi selvittää syitä muiden kilpailijoiden toiminnan tehokkuuteen. Osaston johto osallistui selvitystä varten seminaareihin. Yksi seminaareissa usein esille tulleista syistä oli ohjelmistorobotiikka. Johdon suurimmat huolet ohjelmistorobotiikan hyödyntämisessä olivat sen turvallisuus, implementoinnin kesto ja robottien toimivuus yrityksen ERP-järjestelmien kanssa. Näiden huolien kartoittamiseksi aloitettiin testiprojekti. (M. C. Lacity et al., 2016)

Testiprosessiksi valikoitui yrityksen tilien tarkistus- ja laskentaprosessi. Prosessien suorittamiseen vaadittiin yhteensä 485 000 manuaalista vaihetta kuukausittain. Tarkistamisien ja tilisiirtojen lisäksi prosesseihin kuului myös asianmukainen dokumentointi, joka sekin tehtiin edelleen manuaalisesti. Projektitiimi arvioi 60 prosentin automatisoinnin riittävän kattamaan investoinnin kulut. Prosessi suoritettiin ohjelmistoroboteilla ja tuloksia vertailtiin. Prosessin suoriutumisenopeus oli paljon nopeampaa robottien avulla. Myös robotteja käyttäneet työntekijät olivat tyytyväisiä sen käyttämiseen. Projektiryhmä sai näin todennettua ohjelmistorobotien kannattavuuden ylemmälle johdolle. (M. C. Lacity et al., 2016)

Vuoden 2015 kesällä aloitettiin ensimmäinen implementointisykli. Tavoitteena oli automatisoida kolmen bisnesryhmän tiliprosessit. Tähän sykliin sisältyi yhteensä 19 prosessia. Projektiryhmään kuului nyt talouspalveluiden kolmen henkilön lisäksi myös neljä asiantuntijaa automaatio-ohjelmiston tarjoajalta. Näiden asiantuntijoiden tarkoitus projektissa oli pääosin varmistaa rakennettavan ohjelmistorobotin toimiminen tehokkaasti ja hyvien käytäntöjen mukaisesti. Prosessia suunnitellessa varmistettiin asiantuntijoiden avulla myös se, että vaihe oli mahdollista suorittaa ohjelmistorobotilla, ennen kuin sitä alettiin työstää. Prosessia muutettiin roboteille sopivammaksi ja joitakin osia prosessista poistettiin kokonaan automatisoinnin yhteydessä, jotta se saatiin mahdollisimman tehokkaaksi. Syklin lopussa saavutettiin 89 prosentin automatisaatio prosessin vaiheista arvioidun 60 prosentin sijaan. (M. C. Lacity et al., 2016)

Toinen sykli aloitettiin talvella 2016. Mukaan automatisaation otettiin tällä kertaa kuusi uutta bisnesryhmää. Tähän sykliin mennessä yrityksen oma henkilöstö oli jo ehtinyt oppia tarpeeksi käytettävästä ohjelmistosta, joten vastuu päätöksistä siirtyi nyt enemmän heidän harteilleen.

Implementointi hoitui saman kaavan mukaan kuin ensimmäisessä syklissä. Syksyllä suoritettiin hyväksyntätestaukset ja seuraavan kuukauden aikana toisen syklin prosessit saatiin toimintaan lopullisessa ympäristönsään. Projektitiimin taidot olivat myös kehittyneet jo pisteeseen, jossa he kykenivät parhaimmillaan automatisoimaan yksittäisen prosessin viikossa. (M. C. Lacity et al., 2016)

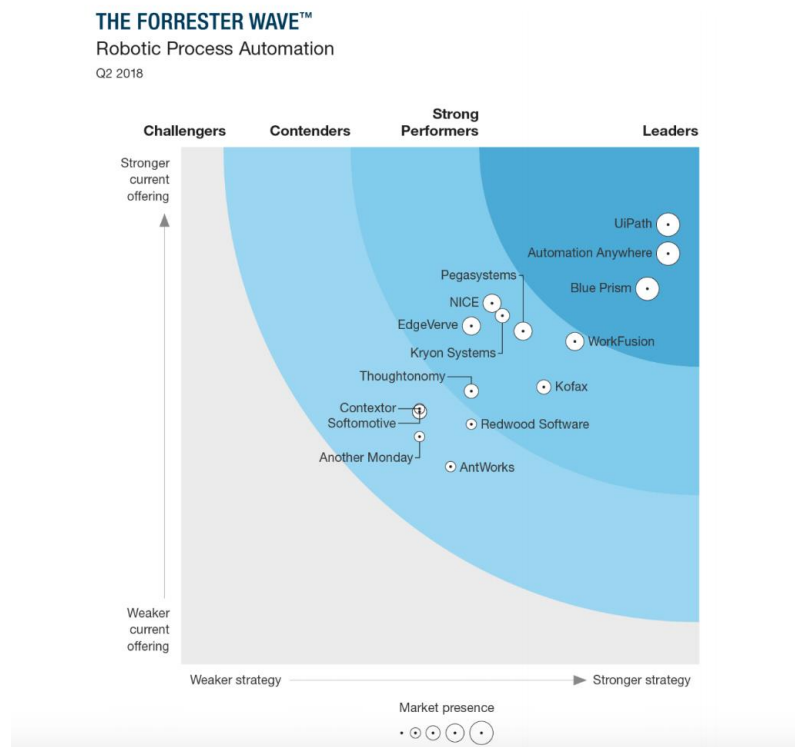
Yrityksen automatisaatiossa seuraava askel on laajentaa ja optimoida nykyisiä automatisaatioprosesseja. Optimisointi aloitetaan ylimääräisten prosessin vaiheiden poistoista. Nämä prosessin osat rakennettiin ohjelmistorobottiin lähinnä tarkistuspisteiksi, jotta saatiin vakuudet prosessin toimivuudesta. Nyt kun prosessin toimivuus on varmistettu, ei näille vaiheilla ole prosessissa tarvetta. Automatisaation laajentamisessa puolestaan yrityksen suunnitelmana oli laajentaa automatisaatiotaan jäljelle jääneiden yksiköiden puolelle. (M. C. Lacity et al., 2016)

## 4 RPA-OHJELMISTOT

RPA-ohjelmistoja löytyy monenlaiseen käyttötarkoitukseen ja -ympäristöön. Tässä luvussa esitellään kahta alan vahvinta kilpailijaa, UiPathia ja Automation Anywhereä. Tarkoituksena oli löytää alusta, joka on helppokäyttöinen myös muillekin kuin automatisaatioon erikoistuneelle henkilöstölle.

### 4.1 UiPath

Hämeen ammattikorkeakoulun henkilöstöpalveluihin aiemmin toteutetussa ICT-projektissa käytetty UiPath tarjoaa visuaalista työkalua automatisointiin. UiPath on yksi alan johtavia työkaluja Blue Prism ja Automation Anywhere-ohjelmistojen ohella (UiPath, n.d.-b). UiPath on maksullinen alusta, mutta antaa mahdollisuuden ilmaiseen Community-julkaisuun kehittäjille ja pienille tiimeille, sekä 60 päivän testiversion yrityksille. UiPath tarvitsee toimiakseen käytettävään selaimen lisäosan, jos halutaan automatisoida selainympäristöä.



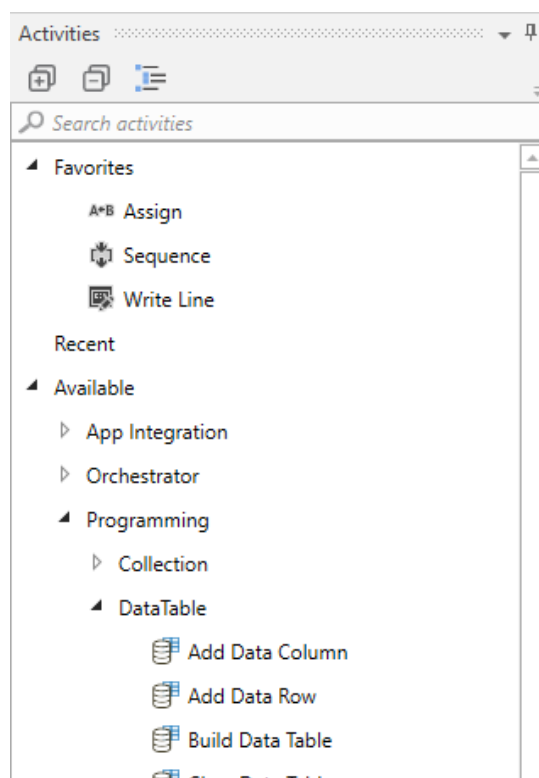
Kuva 4. UiPath verrattuna muihin RPA-kilpailijoihin, Forrester Wave-arviointi (UiPath, 2018, -a)

UiPath on nopeasti noussut johtoasemaan ohjelmistorobotiikka-alustana. Yrityksen Intian toimitusjohtajan Raghunath Subramanianin mukaan UiPath nelinkertaisti työntekijöiden määrän, kuusinkertaista liikevoiton ja sai yli 100 uutta yritysasiakasta vuoden 2016 aikana. Vuonna 2017 UiPath oli saavuttanut jo Automation Anywhere'n ja Blue Prismin

ohjelmistotarjonnan tasolla. (Raghunath, 2017) Kuvassa 4 näkyy vuoden 2018 tilanne, jossa UiPath on siirtynyt kilpailussa johtoasemaan.

#### 4.1.1 UiPathin perusteet ja käyttöliittymä

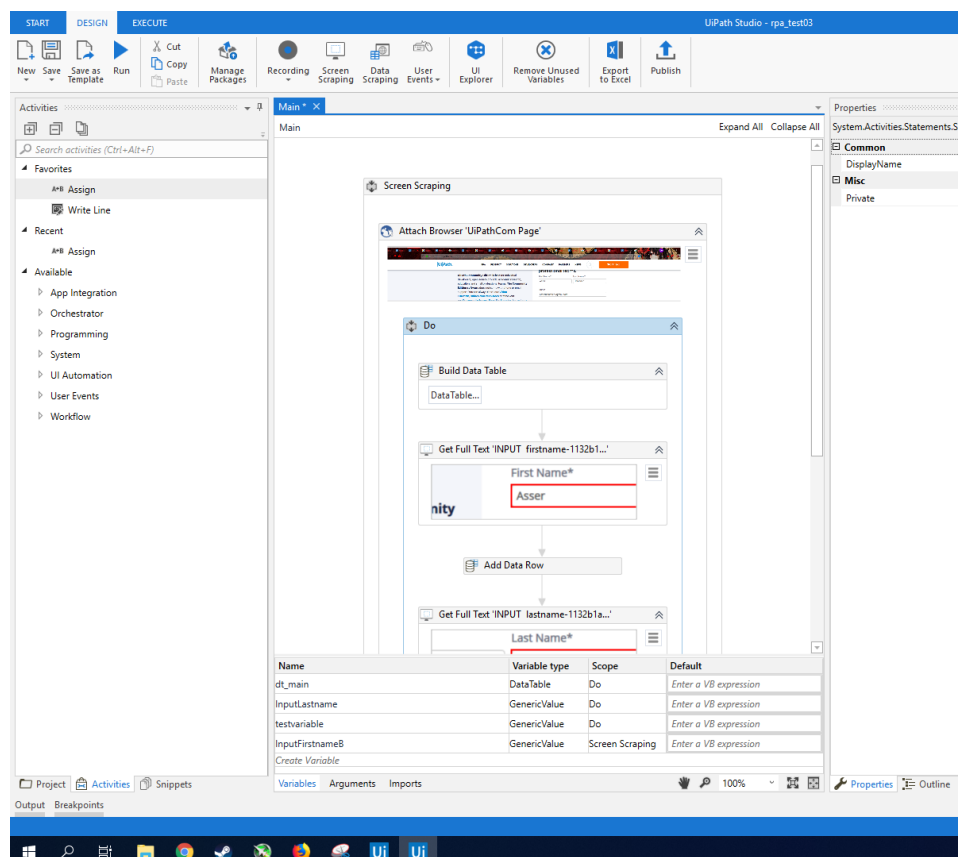
UiPath antaa mahdollisuudet käyttäjälleen siirrä-ja-pudota rakennusmalliin ohjelmaa tehtäessä. Ohjelmassa on useita valmiiksi rakennettuja aktiviteetteja, joista käyttäjä voi valita tarvitsemansa, siirtää sen hiirellä raahtaten rakennettavaan ohjelmaan ja pudottaa sen haluamaansa kohtaan. Kuvassa 5 näkyy muutamia valmiita aktiviteetteja. UiPath tukee oletuksena muun muassa tietokantoja, Microsoft Exceliä ja Microsoft Outlookia. Aktiviteetteja voi luoda myös oman tarpeen mukaan.



Kuva 5. UiPath aktiviteettilista

UiPathilla on mahdollisuus hyödyntää seuraavia ohjelmointikieliä: C++, C#, VB.net ja VB6. Myös Javaa ja Pythonia tuetaan laajennuksilla. UiPathilla on kolme perusaktiviteettiä ohjelmien rakennukseen: vuokaavio (flowchart), jakso (sequence), ja nauhoittaminen. Vuokaaviot ovat hyödyllisiä ohjelman logiikan hahmottamiseen. Vuokaavion sisään laitetaan jaksot, joiden sisälle puolestaan asetetaan ohjelman osat. Jaksot sisältävät yleensä pienempiä aktiviteetteja kuten tietokoneen hiiren klikkaus, tekstin syöttö, tietokannan luonti tai tiedon kerääminen verkkosivulta. Jaksot voivat myös sisältää toisia jaksotai vuokaavioita. Tarvittaessa käyttäjä voi käyttää tallennustoimintoa helpottamaan ohjelman rakentamista.

Kuvassa 6 näkyy UiPath-studion käyttöliittymä kokonaisuudessaan. Studioissa aktiviteetit näkyvät vasemmalla, itse ohjelma keskellä ja aktiviteettien ominaisuudet oikealla. Muuttujat löytyvät keskiosan vasemmasta alalaidasta (Variables).



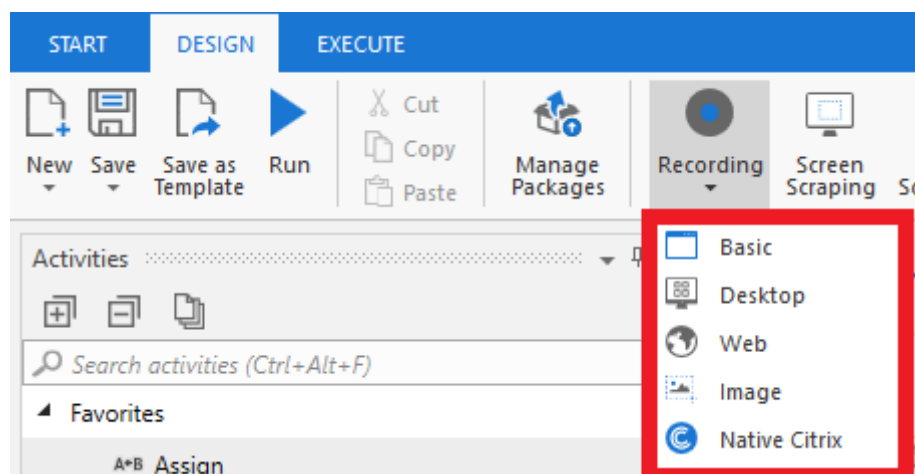
Kuva 6. UiPath-käyttöliittymä

#### 4.1.2 UiPathin tallennusmetodit

UiPathilla on mahdollista laittaa ohjelma tallentamaan automatisoitavan prosessin vaiheet käyttäjän toiminnan mukaan tai manuaalisesti. Automaattinen tallentaminen voi tallentaa muun muassa hiiren vasemman klikkauksen nappeihin, check-boxeihin, pudotusvalikoihin ja samantyyppiisiin elementteihin. Tekstin syöttö on myös mahdollista tallentaa automaattisella tallentimella. Manuaaliseen tallennukseen sisältyvät: pikanäppäimet, muokkausavaimet, hiiren oikea klikkaus, hiiren leijunta elementin päällä ja muutamia muita toimintoja. (UiPath, n.d.-c)

Automaattinen tallennus on jaettu viiteen eri tyyppiin: perus (basic), työpöytä (desktop), verkko (web), kuva (image) ja Citrix. Nämä näkyvät kuvassa 9. Perustallennus luo valitsimen (selector) nauhoitetulle aktiviteetille, jotta robotti löytää kohde-elementin. Valitsin on pätkä tekstiä, joka sisältää tiedon elementin sijainnista. Perustallennus ei kuitenkaan luo säiliötä (container), minkä takia tämä tallennustyyppi on hyödyllisin yksittäisten aktiviteettien nauhoitukseen, ei prosessien tai sen osien kokonaisuuden nauhoitukseen. ”A container gives you a little more context for the

button or field you want to use, so that you can tell windows apart or different areas of the same app” (UiPath, n.d.-d).



Kuva 7. UiPath automaattitallennustyytit

Työpöytä tallennus rakentaa säiliön tallennetuille aktiviteeteille ja on nopeampi rakennetun ohjelman suorittamisessa tämän takia. Verkkotallennus on lähes sama kuin työpöytä tallennus, mutta siinä on muutamia eri oletusasetuksia, johtuen sen räätälöinnistä selaimille ja verkkosovelluksille. (UiPath, n.d.-c)

Neljäs ja viides tallennustyyppi ovat kuva- ja Citrix-metodit. Kuvatallennusta ja Citrixiä käytetään virtuaaliympäristöjen kanssa ja molemmat käyttävät kuvahakua. Kuvatallennus ja Citrix ovat virtuaaliympäristöissä ainoat toimivat tallennustavat UiPathissa, sillä virtuaalikoneet lähettävät käyttäjälle vain kuvaa palvelimelta. Kuvahakua käytettäessä ohjelma tarvitsee tarkat ohjeet toimiakseen, kuten uniikin sanan tai kuvan, jolla kohde-elementtiä haetaan. Uniikit kuvat voivat toimia myös ankkureina, joilla ohjelmalle kerrotaan tehdä toiminto suhteessa ankkurin sijaintiin.

Kuvatallennuksen ja Citrixin suurin ero on käyttölaajuudessa ja -ympäristössä. Kuvatallennus voi automatisoida kuvia, tekstin- ja pikanäppäinsyötöä. Citrix puolestaan pystyy edellisiin toimintoihin ja niiden lisäksi klikkaukseen, tekstin keräämiseen, tiedon keräämiseen ja muutamisiin muihin toimintoihin. Vaikka Citrix-tallennus on toiminnoiltaan monikäyttöisempi, käytetään sitä vain siihen tuetuissa ympäristöissä. Citrix-tallennus vaatii toimiakseen lisäosan UiPath studioon, sekä etäkäyttöön tarkoitetun ohjelmiston käytettävälle virtuaalikoneelle. Nämä lisäosat mahdollistavat valitsimien generoinnin aktiviteeteille myös virtuaalikoneessa, jolloin automatisointi on luotettavampaa verrattuna kuvatallennukseen. (UiPath, n.d.-a)

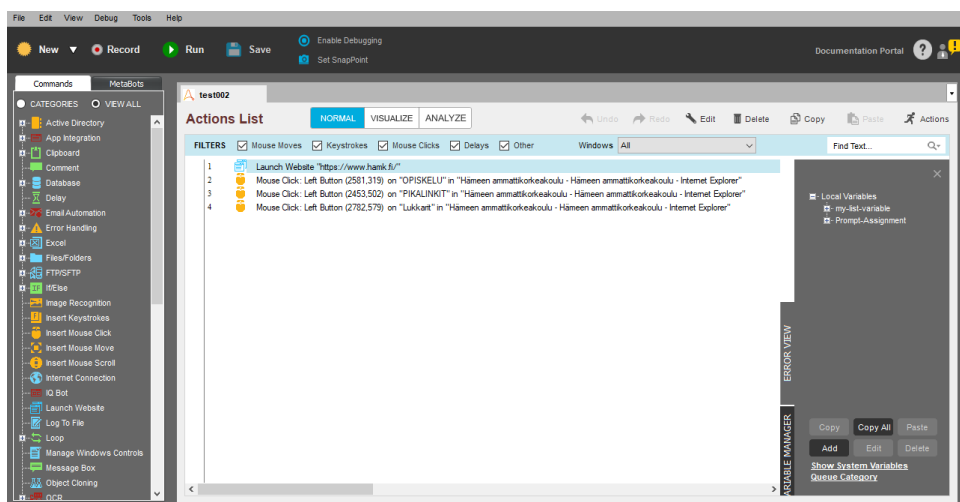
## 4.2 Automation Anywhere

Automation Anywhere on UiPathin ja Blue Prismin suurin kilpailija (UiPath, n.d.-b). Ohjelmisto on saatavilla ilmaisena Community-versiona (karsituilla ominaisuuksilla), sekä maksullisena Enterprise-versiona. Community-versio on saatavilla ainoastaan pienille yrityksille, kehittäjille ja opiskelijoille (Automation Anywhere, n.d.-c).

### 4.2.1 Automation Anywheren perusteet ja käyttöliittymä

Ohjelmistorobottia hallinnoidaan joko paikallisesti koneelta, jolla projekti on tai se voidaan tallentaa pilvipalveluun. Pilvitalennettua robottia voidaan hallita Control Roomin avulla. Control Roomia on mahdollista käyttää etänä myös puhelinsovelluksella tai selaimen kautta muilla laitteilla kuten esimerkiksi tabletilla. Automation Anywhere jonottaa prosesseja, jos niitä ajetaan useampia samaan aikaan. (Automation Anywhere, n.d.-b)

Hyödynnettäviä ohjelmointikieliä Automation Anywherellä ovat lähinnä Java, VB.net, sekä C#. Kuvassa 8 on studion käyttöliittymä. Näkymä on pohjimmiltaan samankaltainen kuin aiemmin esitellyn UiPathin: komennot löytyvät vasemmalta, rakennettava ohjelma keskeltä ja ominaisuudet oikealta. Automation Anywheren komennot vastaavat UiPathissa aktiviteetteja. Samaan tapaan myös Automation Anywherellä on mahdollista käyttää prosessin automaattitallennusta, jolloin käyttäjän ei tarvitse itse kasata prosessin osia ohjelmaansa. Tämän ominaisuuden käyttäminen vaatii lisäosan asentamisen selaimen, jos selainympäristöjä halutaan automatisoida.



Kuva 8. Automation Anywhere: studion käyttöliittymä



#### 4.2.2 Automation Anywheren tallennusmetodit ja metabotit

Automation Anywheren automaattitallennuksessa on kolme eri tilaa standardin lisäksi. Näistä ensimmäinen on Smart record. Smart recordin avulla on mahdollista tallentaa toimintaa esimerkiksi Javalla toimivissa ohjelmissa. Tallennukseen otetaan mukaan näppäinkomennot ja hiiren klikkaukset. (Automation Anywhere, n.d.-d)

Web record puolestaan on nimensä mukaisestikin rajattu selaimessa tapahtuvaan tallentamiseen. Mahdollisia tallennustapahtumia ovat muun muassa: datan syöttäminen, sen kerääminen, kerätyn datan tallennus paikalliseen tiedostoon, sekä selaimissa toimivien ohjelmien käytön tallennus. Automation Anywhere pystyy myös paikantamaan kohde-elementit, vaikka niiden sijainti vaihtuisi ruudulla. (Automation Anywhere, n.d.-d)

Secure recordia käytetään, kun halutaan välttää arkaluonteisten tietojen tallennusta. Esimerkiksi kuvia tai muuttujia ei tallenneta, riippuen käytetävästä toiminnosta. Ruutukaappausten esto robotin toimiessa on myös mahdollista. Automation Anywhere mahdollistaa myös Citrix-ympäristöjen hyödyntämisen kuin vastaavasti UiPathissa. (Automation Anywhere, n.d.-d)

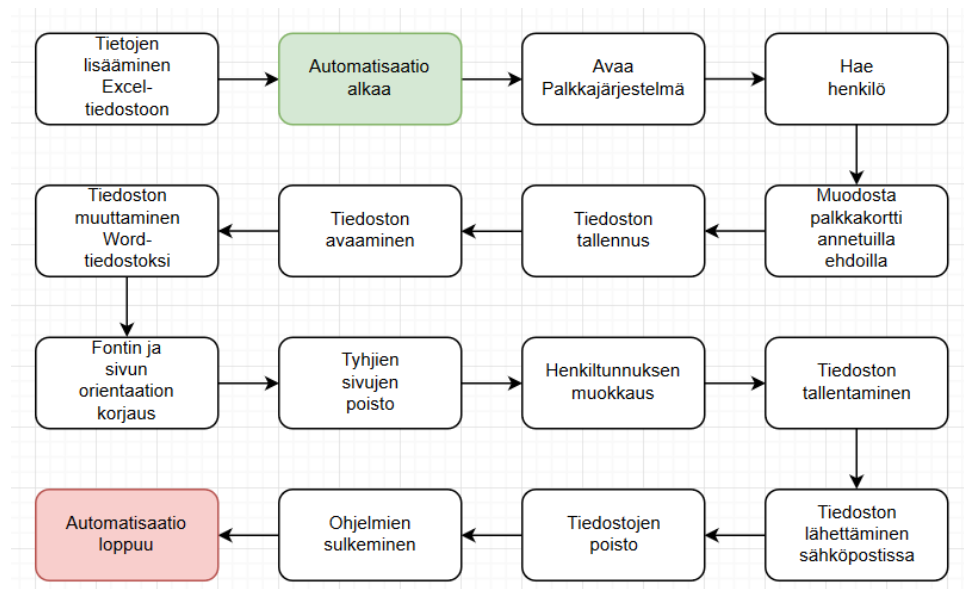
Metabotit mahdollistavat ohjelmien tai sen osien uudelleen käytön, jolloin niitä voi hyödyntää muissakin ohjelmistoroboteissa. Suurin metabotien etu on kuitenkin niiden lisäämät mahdollisuudet prosessien automatisoinnissa. Metabotin avulla kyetään luomaan paikallinen kopio automatisoitavasta ohjelmasta, jolloin prosessia pystyy automatisoimaan ilman suoraa yhteyttä ohjelmaan. Metabotteja pystyy myös jakamaan tai hankkimaan Bot Storessa. Bot Storessa olevat metabotit ovat ammattilaisten tarkistamia. (Automation Anywhere, n.d.-a)

## 5 ROBOTTIEN VERTAILU

Käytännön projektin päämääränä oli testata kahta eri automatisaatio-ohjelmistoa, UiPathia ja Automation Anywherea, ja selvittää niiden eroavaisuuksia, heikkouksia ja vahvuuksia.

### 5.1 Testattava prosessi

Testiprojektiksi ohjelmistoille valittiin palkkakortin muodostus ja lähettäminen sähköpostilla Hämeen ammattikorkeakoulussa käytettävästä palkkaohjelmasta (Personec F). Kuvassa 9 on kuvattuna koko automatisoidun prosessin oleelliset kohdat.



Kuva 9. Palkkakortin prosessikaavio

Ennen automatisointia on tiedot lisättävä Excel-tiedostoon, josta saadaan henkilötiedot myöhempiä vaiheita varten. Tämä vaihe automatisoidaan ohjelman jatkekehityksessä, mutta testin vuoksi tietojen lisääminen tehdään toistaiseksi käsin. Kerättäviin tietoihin tarvitaan henkilötunnus, palkanlaskentayksikkö, sekä haettavan aikavälin alku- ja loppupäivä.

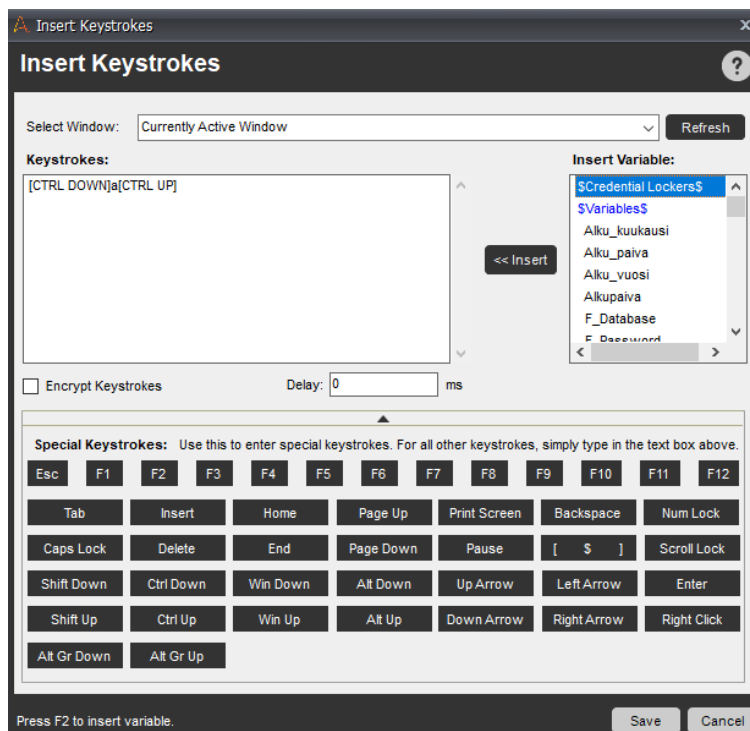
Seuraava vaihe on käynnistää ja kirjautua palkkaohjelmaan. Tämä on aina toteutettava Internet Explorer-selaimella, johtuen palkkaohjelmasta. Ohjelma aukeaa Java-ikkunaan, jossa seuraavaksi etsitään henkilö annetun henkilötunnuksen perusteella. Palkkakortin muodostusvaiheessa ohjelmistorobotti syöttää annettujen tietojen mukaan palkanlaskentayksikön ja poiminnan alku- ja loppupäivät. Tämän jälkeen käynnistetään palkkaohjelman tietojen haku.

Kun tiedot on haettu, tallennetaan tiedosto haluttuun tiedostopolkuun. Palkkaohjelman antama tiedostomuoto on tässä tapauksessa oletuksena .rtf, (rich text file) joten se on muutettava seuraavaksi Microsoft Wordille sopivaan muotoon. Tiedosto avataan Wordilla ja muunnetaan se ruudulle ilmestyvän oletuksen mukaan. Seuraavaksi tiedoston tekstin fontti muutetaan kokoon 8 ja orientaatio horisontaaliksi, jotta tiedot mahtuvat paremmin näkyville. Samalla poistetaan myös tyhjä sivu dokumentin alusta, jonka palkkaohjelma tiedostoon jättää. Lopuksi poistetaan tekstissä näkyvän henkilötunnuksen viimeiset viisi merkkiä tietosuojan vuoksi. Muutosten jälkeen tiedosto tallennetaan .docx -formaattiin. Prosessin lopuksi tiedosto lähetetään sähköpostiliitteenä eteenpäin, jonka jälkeen molemmat .rtf ja .docx -tiedostot poistetaan ja käytetyt ohjelmat suljetaan.

## 5.2 Eroavaisuudet, heikkoudet ja vahvuudet

UiPath ja Automation Anywhere suoriutuivat molemmat prosessin automatisaatiosta, mutta ohjelmien käytettävyydessä oli ajoittain suuriakin eroja. Uipath on käyttöliittymältään modernimpi, mikä on myös ajoittain haitaksi. UiPathilla ohjelma kasataan studiossa sisäkkäisiin säiliöihin (container), mikä vaikuttaa aluksi kätevältä, mutta ohjelman monimutkaistuksessa elementtien ryhmittymisistä on vaikea ottaa selvää. Tämä voi johtaa myös muuttujien kohdalla vaikeuksiin pysyä selvillä kunkin muuttujan suojaustasoista. UiPathissa ei ole rivejä, kuten monessa ohjelmointiohjelmassa, joten aktiviteetit on muistettava nimetä aina selvästi tai virheiden etsimisestä tulee vaikeaa. Automation Anywheressä puolestaan ohjelmaa kasataan riveittäin, joten virheilmoituksissa ilmoitetaan automaattisesti virheen rivinumero. Muuttujat ovat Automation Anywheressä globaaleja, joten niitä ei ole luokiteltu käyttölaajuuden mukaan.

Näppäinsyötteiden kohdalla UiPath häviää käyttöhelppoudessa. Näppäinsyötteitä oli pääosin järkevämpää käyttää tässä prosessissa kuin hiiren klikkauksia, johtuen niiden helpommasta implementoinnista ja vakaammasta toimivuudesta. Molemmissa automatisointityökaluissa on annettu listaukset käytettävistä näppäimistä, mutta Automation Anywheren yksinkertainen esitystapa, klikattavat nappulat, vievät voiton. UiPathilla on käytössä pudotusvalikko, josta erikoisnäppäimiä voi valita, mutta ne on itse korjattava tarpeen mukaan, mikäli näppäintä mieli painaa pohjassa tai vapauttaa kyseisen näppäimen.



Kuva 10. Automation Anywhere näppäinkomennot

Näppäinkomentojen helppokäyttöisyyden eron huomaa helposti komennon pituuden kasvaessa. Kuvassa 10 esimerkki lyhyestä Automation Anywheren näppäinkomennosta ([CTRL DOWN]a[CTRL UP]), joka kirjoitetaan vastaavasti UiPathissa:

”[d(ctrl)]a[u(ctrl)]”

Puolestaan hiiren klikkauksissa UiPath tuntui olevan luotettavampi, johon valitsimen helposta muokattavuudesta. Varsinkin palkkaohjelman Java-ikkunassa Automation Anywherellä oli ongelmia saada ohjelma klikkaamaan pudotusvalikkoja.

Yhteisinä ongelmina ohjelmilla oli muun muassa; usean näytön käyttäminen, päivämäärien datamuokkaus, sekä Microsoft Wordin automatisointi. Usean näytön käyttäminen automatisoidessa ei ole suositeltavaa, sillä se saattaa tehdä ikkunoiden löytämisen mahdottomaksi automatisaatio-ohjelmalle, varsinkin jos ohjelma kehitetään usealla näytöllä ja käytetään vain yhdellä näytöllä. Päivämäärien datamuokkaus puolestaan vaatii ohjelmiston laajennuksia, koodaamista ja/tai muuttujilla päivämäärän leikelyä päiviin, kuukausiin ja vuosiin. Muussa tapauksessa Excelistä tulevien päivämäärien muoto on väärä, jos päivä tai kuukausi on pienempi kuin 10. Näissä tapauksissa yksi ratkaisumalli on tarkistaa muuttujan arvo ja tarpeen mukaan lisätä nolla päivän tai kuukauden eteen, jotta syöte pysyy oikean muotoisena palkkaohjelmalle. Wordin automatisointi oli molemmilla alustoilla tehtävä näppäinkomennoilla. Word-tiedoston valitsimia ei klikkauselementeille löytynyt, eikä UiPathin Word-laajennuskaan tarjonnut työkaluja ongelmaan.

Kumpikin ohjelmisto soveltuu Hämeen ammattikorkeakoulun tarpeisiin, vaikkakin Automation Anywhere vaikuttaa soveltuvan paremmin suurien data- tai tiedostomäärien käsittelyyn paremmin ja UiPath puolestaan enemmän selaimessa tapahtuvaan automatisointiin.

## 6 YHTEENVETO

RPA soveltuu niin yleisesti, kuin Hämeen ammattikorkeakoulussakin, prosesseihin, jotka ovat yksinkertaisia, mutta vaativat paljon työtä. Näitä prosesseja on myös tapahduttava usein, jotta automatisaatiosta on hyötyä. Hämeen ammattikorkeakoulussa tällaisia prosesseja ovat esimerkiksi monet henkilöstö- ja palkkatietojen käsittelyä koskevat työtilanteet.

Robottiikka vaatii järkevästi toteutuakseen monta asiaa. Näitä ovat prosessin tuntemus, projektihallinta, testaus, riskien tunteminen, sekä käytettävän ohjelmiston kustannukset ja rajat. Myös organisaation ja sen työntekijöiden ymmärrys ja hyväksyntä automatiikkaa kohtaan vaikuttaa paljon robotisoinnin toteutumiseen ja sen potentiaalın hyödyntämiseen.

Tässä työssä esitellyt automatisaatio-ohjelmat ovat hyvin käyttäjäystävällisiä ongelmistaan huolimatta. Ohjelmien käytön aloittaminen ei vaadi liiojia ja niitä on mahdollista käyttää muidenkin kuin pelkästään automatisaatioon erikoistuneen henkilöstön. Perusteiden ymmärtämiseen riittää muutama viikko alustan käyttöä ja kouluttautumista. Käyttäjäystävällisyys kuitenkin maksaa, joten se on huomioitava lisenssien sopivuudessa suhteessa ohjelman käytettävyyteen.

## 7 LÄHTEET

- Automation Anywhere. (n.d.-a). Automation Anywhere Metabots. Haettu 5.12.2019 osoitteesta <https://resources.automationanywhere.com/home/metabots>
- Automation Anywhere. (n.d.-b). Best Robotic Business Process Automation Software and Digital Workforce Platform | Automation Anywhere. Haettu 27.5.2019 osoitteesta <https://www.automationanywhere.com/products/enterprise>
- Automation Anywhere. (n.d.-c). Community Edition. Haettu 27.5.2019 osoitteesta <https://www.automationanywhere.com/lp/community-edition>
- Automation Anywhere. (n.d.-d). Record a task. Haettu 5.12.2019 osoitteesta <https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/aae-client/bot-creator/getting-started-with-aa/recording-simple-tasks.html>
- Bloomberg, J. (2018). Why You Should Think Twice About Robotic Process Automation. Haettu 26.3.2019 osoitteesta <https://www.forbes.com/sites/jasonbloomberg/2018/11/06/why-you-should-think-twice-about-robotic-process-automation/#5178a0d15fe1>
- CFB Bots. (2018). The Difference between Robotic Process Automation and Artificial Intelligence | Robotic Process Automation | CFB Bots. Haettu 27.3.2019 osoitteesta <https://www.cfb-bots.com/single-post/2018/04/09/The-Difference-between-Robotic-Process-Automation-and-Artificial-Intelligence>
- Edureka. (2019). Blue Prism vs UiPath vs Automation Anywhere | RPA Tools Comparison | Edureka - YouTube. Haettu 27.5.2019 osoitteesta <https://www.youtube.com/watch?v=EnpfCl9iAng>
- Fersht, P., Gupta, S., & Christopher, E. (2019). RPA is dead. Long live Integrated Automation Platforms - Horses for Sources. Haettu 6.6.2019 osoitteesta [https://www.horsesforsources.com/rpa-dead-integrated-automation-platforms\\_041519](https://www.horsesforsources.com/rpa-dead-integrated-automation-platforms_041519)
- HISTORY of Robotic Process Automation | Roboticprocess365.com. (2018). Haettu 13.8.2019 osoitteesta <https://roboticprocess365.com/history-of-robotic-process-automation/>
- Lacity, M. C., Willcocks, L. P., & Craig, A. (2016). Robotizing Global Financial Shared Services at Royal DSM. *The Outsourcing Unit Working Research Paper Series*, 16/02(November), 1–26. Haettu 18.12.2019 osoitteesta <http://www.umsl.edu/~lacitym/OUWP022016Post.pdf>
- Lacity, M., Willcocks, L., & Craig, A. (2015). *The Outsourcing Unit Working Research Paper Series Paper 15/02 Robotic Process Automation at Telefónica O2 Research on Business Services Automation Research Objective*. Haettu 6.6.2019 osoitteesta <https://www.umsl.edu/~lacitym/TelefonicaOUWP022015FINAL.pdf>

- Lehtonen, T., Tuomivaara, S., Rantala, V., Käsälä, M., Mäkilä, T., Jokela, T., ... Isomäki, M. (2014). Sulautettujen järjestelmien ketterä käsikirja. Haettu 2.6.2019 osoitteesta [https://tech.utu.fi/embedded\\_kasikirja/index.html#sisallys](https://tech.utu.fi/embedded_kasikirja/index.html#sisallys)
- Liao, X. (2019). Top 8 risks associated with RPA and how to mitigate them. Haettu 16.10.2019 osoitteesta <https://www.joltag.com/blog/top-8-risks-associated-with-rpa-and-how-to-mitigate-them>
- Martinsuo, M., & Blomqvist, M. (2010). Prosessien mallintaminen osana toiminnan kehittämistä. In *Tampere University of Technology. Faculty of Business and Technology Management. Lecture Notes 2*. Haettu 2.6.2019 osoitteesta [https://tutcris.tut.fi/portal/files/2098668/prosessien\\_mallintaminen.pdf](https://tutcris.tut.fi/portal/files/2098668/prosessien_mallintaminen.pdf)
- Odess-Gillett. (2018). Testing Your RPA Project - Mimics.io. Haettu 25.11.2016 osoitteesta <https://mimics.io/blog/testing-your-rpa-project/>
- Raghunath, S. (2017). UiPath recognised as “The RPA Industry Leader” in The Forrester Wave™ RPA Q1 2017. Haettu 6.6.2019 osoitteesta <https://www.linkedin.com/pulse/ui-path-recognised-rpa-industry-leader-forrester-q1-2017-subramanian>
- Rose, D. (2017). Agile at Work: Planning with Agile User Stories: Become an Agile certified ... - Douglas Rose - Google-kirjat. Haettu 2.6.2019 osoitteesta <https://books.google.fi/books?id=0qV6DwAAQBAJ&pg=PT14&dq=user+stories&hl=fi&sa=X&ved=0ahUKewixoc24IMviAhUbwsQBHXoxDIYQ6AEILzAB#v=onepage&q&f=false>
- SkyMind. (n.d.). AI Applications in Robotic Process Automation (RPA) | SkyMind. Haettu 26.3.2019 osoitteesta <https://skymind.ai/wiki/robotic-process-automation-rpa>
- Smartsheet. (n.d.). The 4 Values and 12 Principles of the Agile Manifesto. Haettu 2.6.2019 osoitteesta <https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>
- Software Testing Fundamentals. (n.d.). Agile Testing - Software Testing Fundamentals. Haettu 2.6.2019 osoitteesta <http://softwaretestingfundamentals.com/agile-testing/>
- UiPath. (n.d.-a). About Native Citrix Automation. Haettu 9.8.2016 osoitteesta <https://studio.uipath.com/docs/about-native-citrix-automation>
- UiPath. (n.d.-b). Forrester Wave Robotic Process Automation, Q2 2018. Haettu 6.6.2019 osoitteesta <https://www.uipath.com/company/rpa-analyst-reports/forrester-wave-2018-robotic-process-automation>
- UiPath. (n.d.-c). Recording Types. Haettu 9.8.2019 osoitteesta <https://studio.uipath.com/docs/about-recording-types>
- UiPath. (n.d.-d). UI Activities Properties. Haettu 2.6.2019 osoitteesta <https://studio.uipath.com/v2017.1/docs/ui-activities-properties>



Willcocks, L., Lacity, M. C., & Craig, A. (2015). Robotic Process Automation at Xchanging. *The Outsourcing Unit Working Research Paper Series*, (June 2015), 1–26. Haettu  
9.6.2019 osoitteesta  
[http://eprints.lse.ac.uk/64518/1/OUWRPS\\_15\\_03\\_published.pdf](http://eprints.lse.ac.uk/64518/1/OUWRPS_15_03_published.pdf)