

# **Laskuarkiston työprosessin automatisointi ohjelmistorobotiikalla**

LAB-ammattikorkeakoulu

Insinööri (AMK)

2023

Juho Ahonen

## Tiivistelmä

Tekijä(t) Juho Ahonen	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika 2023
	Sivumäärä 33	
Työn nimi <b>Laskuarkiston työprosessin automatisointi ohjelmistorobottiikalla</b>		
Tutkinto ja koulutusala Insinööri (AMK), Tieto- ja viestintätekniikan koulutus		
Toimeksiantajaorganisaatio (jos opinnäytetyöllä on toimeksiantaja) Telia Finland Oyj		
<p>Tiivistelmä</p> <p>Tässä opinnäytetyössä käydään läpi ohjelmistorobotiikan keskeisiä asioita kuten, mitä ohjelmistorobotiikka tarkoittaa ja mitä ja millaisia ovat ohjelmistorobotiikan kehitykseen tarjotut työkalut. Tavoitteena on automatisoida Telia Finland Oyj:n laskuarkistosta tulevan virheilmoituksen lähettäminen asiakkaalle, ohjelmistorobotiikan työkaluja hyödyntäen.</p> <p>Ohjelmistorobotiikka on työpöytäautomaatioon keskittyvä teknologian ala, jolla pyritään automatisoimaan päätetyössä tehtyjä rutiiniprosesseja. Ohjelmistorobotti käyttää kohdejärjestelmiä samalla tavalla kuin tavallinen käyttäjäkin.</p> <p>Yrityksellä on työprosessi, joka noudattaa aina samaa rutiininomaista kaavaa. Prosessin automatisointi on todettu kannattavaksi, joten prosessille luodaan ohjelmistorobotti Blue Prism ohjelmalla. Blue Prism on low-code käyttöliittymän tarjoava ohjelmistorobotiikan kehitystyökalu.</p> <p>Ohjelmistorobotti saatiin kehitettyä halutulla tavalla ja työprosessiin kuluva aika lyheni huomattavasti. Prosessiin sidotut ihmisresurssit voitiin siirtää vaativimpiin töihin.</p>		
Asiasanat Ohjelmistorobotiikka, Prosessiautomaatio, RPA, Blue Prism		

## Abstract

Author(s)	Type of Publication	Published
Juho Ahonen	Thesis, UAS	2023
	Number of Pages	
	33	
Title of Publication		
<b>Robotic process automation solution to automate manual work process done in invoice archive</b>		
Degree, Field of Study		
Engineer (UAS), Information and communication technology		
Organisation of the client (if the thesis work is commissioned by another party)		
Telia Finland Oyj		
Abstract		
<p>This thesis goes through the key issues of software robotics, such as what is software robotics and what are the tools offered for the development of software robots. The goal is to automate the sending of an error message from the company's invoice archive to the customer, using the tools for software robotics.</p> <p>Software robotics is a field of technology that focuses on desktop automation. It aims to automate routine processes performed when working with computers. The software robot uses the target systems in the same way as a normal user.</p> <p>Telia Finland Oyj has a work process that always follows the same routine pattern. Automation of the process has been found to be profitable, so a software robot is created for the process using the Blue Prism program. Blue Prism is a software robotics development tool offering a low-code user interface.</p> <p>The software robot was able to be developed in the desired way and the time spent on the work process was significantly reduced. Human resources tied to the process could be transferred to more demanding jobs.</p>		
Keywords		
Robotic process automation, RPA, Blue Prism		

## Sisällys

1	Johdanto.....	1
2	Ohjelmistorobotiikka .....	2
2.1	Prosessien automaatio .....	2
2.2	Automatisoitavan prosessin elinkaari .....	2
3	Blue Prism .....	5
3.1	Blue Prism historia.....	5
3.2	Kehitys Blue Prismillä .....	5
3.3	Work Template .....	7
3.4	Blue Prism -prosessi.....	9
3.5	Blue Prism -objekti.....	10
3.5.1	Application Modeller .....	11
3.5.2	Objektin viimeistely .....	13
3.6	Virheen käsittely .....	15
3.7	Virhetiedot .....	16
3.8	Työjono.....	17
4	Automatisoitavan prosessin tunnistus .....	19
4.1	Prosessin arviointi.....	19
4.2	Prosessin analyysi .....	19
5	Laskuarkiston virheilmoituksen automatisointi .....	21
5.1	Alustus.....	21
5.1.1	Manuaalisen prosessin kuvaus .....	21
5.1.2	Automatisoitavan prosessin kuvaus .....	21
5.2	Järjestelmien kartoitus .....	24
5.3	Prosessin luonti .....	24
5.4	Objektin luonti.....	25
5.5	Mallinnus- ja objektisivujen rakentaminen .....	26
5.6	Laskuarkiston käyttösiivu .....	27
5.7	Lopullinen prosessi .....	29
5.8	Lopullisen prosessin virheen käsittely .....	30
6	Yhteenveto .....	32
	Lähteet .....	34

## 1 Johdanto

Teknologian kehittyessä työelämä mullistuu. Melkeinpä jokaiselta työpaikalta löytyy manuaalisia, rutiininomaisia työtehtäviä ja prosesseja, jotka eivät vaadi sen suurempaa päätelykykyä, mutta silti niitä joutuu joku tekemään. Tämä manuaalinen työ luonnollisesti luo yritykselle kuluja. Koska yritykset pyrkivät jatkuvasti luomaan enemmän tuottoa joko säästämällä vanhasta tai investoimalla uuteen, on moni yritys jo alkanut selvittämään, millä keinoin nykyisiä työprosesseja saataisiin automatisoitua. Automatisoimalla yksinkertaisemmat työtehtävät saadaan myös allokoitua ihmisresurssit järkevämpiin työtehtäviin, jotka oikeasti vaativat ajatustyötä. (Also 2023.)

Tämän opinnäytetyön tavoitteena on automatisoida Telia Finland Oyj:n laskuarkistossa tehty manuaalinen työ ohjelmistorobotiikan tarjoamilla työkaluilla. Ennen itse automaatiotratkaisun käsittelyä, tässä opinnäytetyössä tarkastellaan laajemmin, mitä ohjelmistorobotiikka käytännössä tarkoittaa.

Telia Finland Oyj on osa ruotsalaista Telia Company -konsernia, joka on pohjoismaiden ja Baltian johtava teleoperaattori. Telia Finland Oyj toimii Suomen maayhtiönä. Telia Finland Oyj tarjoaa laajan valikoiman palveluja, mukaan lukien matkapuhelin- ja laajakaistapalveluita, televisiopalveluita sekä yrityksille suunnattuja tietoliikenne- ja IT-palveluja. Yrityksen liikevaihto vuonna 2022 oli 1,2 miljardia euroa ja yritys työllisti 1660 henkilöä. (Asiakastieto.)

Ohjelmistorobotiikka automatiikan ratkaisuna tuo mukanaan tapoja säästää resursseja nykyisissä jo olemassa olevissa prosesseissa. Ohjelmistorobotiikka mahdollistaa myös sellaisten uusien prosessien kehittämisen, jotka olivat aiemmin liian aikaa vieviä tai monimutkaisia manuaalisesti hallittaviksi. Ohjelmistorobotiikan avulla pyritään tehostamaan prosesseja, vähentämään inhimillisiä virheitä ja säästämään arvokasta aikaa sekä resursseja. (Also 2023.)

Yritykset ovat yhä kiinnostuneempia ohjelmistorobotiikan tarjoamista mahdollisuuksista. Tulevaisuudessa prosessien automatisointi tulee koskemaan monia mahdollisia tehtäviä, jotka voidaan suorittaa toistuvasti ja säännönmukaisesti. (Also 2023.)

Tässä opinnäytetyössä keskitytään erityisesti yhteen IT-alan yrityksen työprosessiin, joka on tarkoitus automatisoida hyödyntäen ohjelmistorobotiikan työkaluja. Prosessin automaatiotratkaisuun tullaan käyttämään Blue Prism nimistä työkalua. Työkaluksi valikoitui Blue Prism, koska se on tämän yrityksen yleisesti käyttämä ohjelmistorobotiikan työkalu.

## 2 Ohjelmistorobotiikka

### 2.1 Prosessien automaatio

Ohjelmistorobotiikka on suhteellisen uusi teknologinen ratkaisu, jonka avulla automatisoidaan rutiiniprosesseja IT-alan töissä. Ohjelmistorobotiikan tarkoituksena on automatisoida liiketoimintaprosesseja, vähentää kustannuksia ja parantaa tehokkuutta. Myös laadun varmistus, lyhentynyt prosessien läpimenoaika ja nopeampi asiakaspalvelu ovat ohjelmistorobotiikan todennettuja hyötyjä. (Aslani 2020.)

Ohjelmistorobotti tekee käytännössä työtä samalla tavalla kuin ihminenkin, mutta paljon tehokkaammin. Ohjelmistorobotin tarkoituksena on suorittaa töiden toistuvia rutiineja, joiden tekemiseen ihmiseltä kuluu merkittävästi aikaa. Esimerkiksi tietojen kopiointi Excelistä yrityksen tietojärjestelmään on yleinen monessa yrityksessä hyvin läsnä oleva työ, jota edelleen tehdään manuaalisesti. Työhön kuluu paljon aikaa ja resursseja. Kun ohjelmistorobotti valjastetaan tekemään samat rutiinit, tippuu työn suoritus aika murto-osaan ja kustannukseksi tulee vain robotin investoinnin vaatimat kulut. Ohjelmistorobotiikasta käytetään usein lyhennettä RPA eli Robotic Process Automation. (Aslani 2020.)

Prosessiautomaatioon on tarjolla monia työkaluja ja niistä suosituimpiin kuuluu Blue Prism, UiPath ja Automation Anywhere. Koska työkaluja on niin monia ja ohjelmistorobotiikan käyttöönottoa aloittelevilla yrityksillä tietoa ko. teknologiasta vähän, vaatii oikean työkalun valitseminen aiheeseen perehtymistä ja tarpeiden kartoittamista. (Software Testing Help 2023.)

Kaikki työkalut tarjoavat samaa asiaa eli automaatiota, mutta kaikilla työkaluilla on myös täysin erilaiset käyttömahdollisuudet. Yrityksen on siis tärkeä kartoittaa eri työkalujen tarjoamia mahdollisuuksia ja ominaisuuksia, niin että ne sopivat parhaiten yrityksen liiketoimintaan. Tärkeimpinä vertailukohteina on yleensä työkalun käyttöönottoon liittyvät kustannukset, työkalun ylläpitoon liittyvät seikat, sekä työkalun skaalautuvuus yrityksen tarpeisiin. (Software Testing Help 2023.)

### 2.2 Automatisoitavan prosessin elinkaari

Prosessin automatisointi lähtee käyntiin ideasta tai tarpeesta automatisoida jokin tietty työprosessi. Tarpeen automatisoinnille voi luoda yrityksen halu säästää resursseja ja samalla parantaa prosessin tehokkuutta. Idea voi tulla halusta selvittää onko kyseessä olevan prosessin automatisointi mahdollista ja ennen kaikkea hyödyllistä. (Sahiti 2023.)

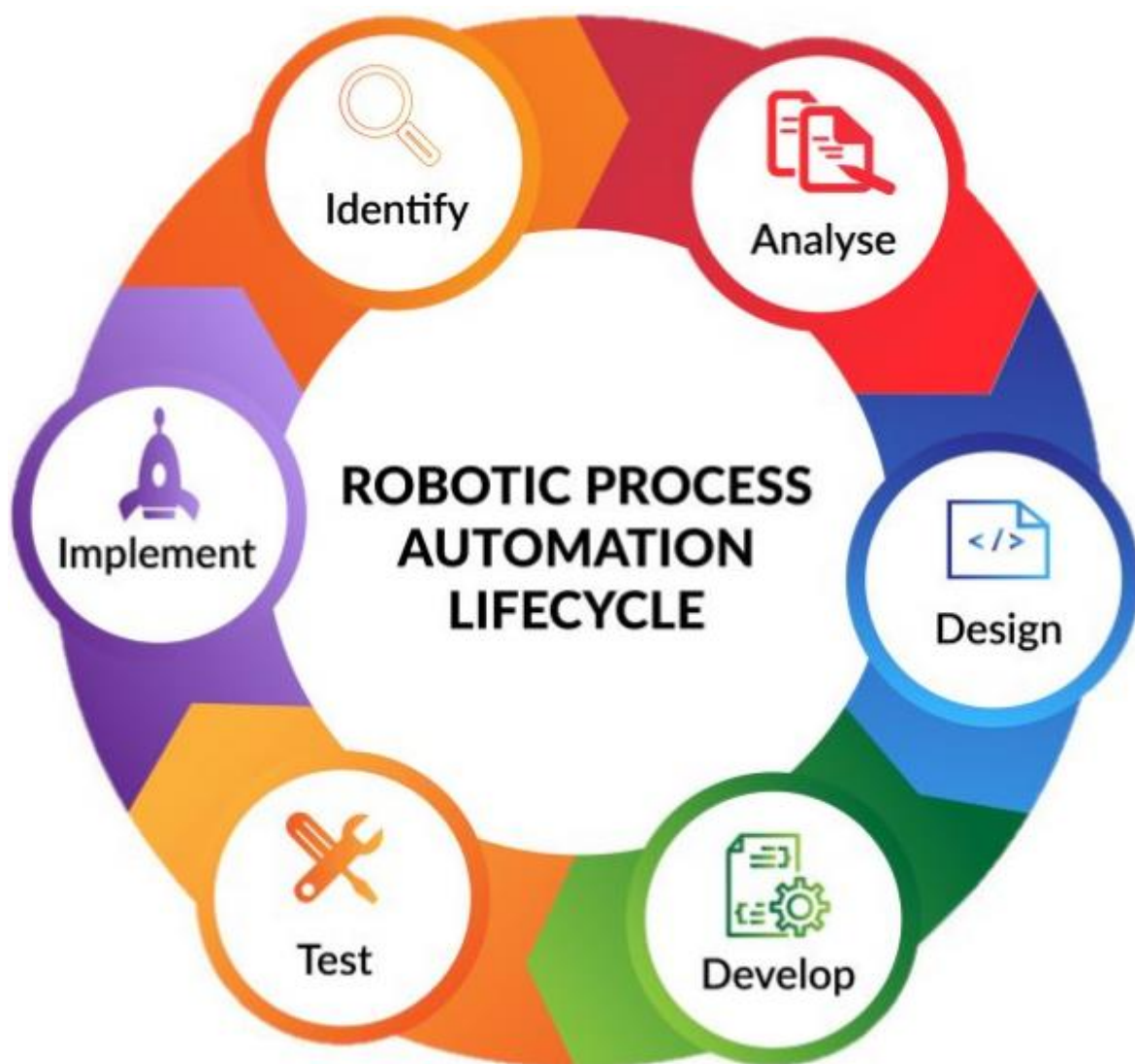
Kun automatisoinnin kannattavuutta lähdetään selvittämään, on kyseessä arviointivaihe. Tässä vaiheessa olennaista on arvioida automatisoinnin kannattavuutta. Tavoitteena on saada kokonaiskuva siitä, onko automatisointi mahdollista, miten automatisointi integroidaan nykyiseen toimintaympäristöön ja mitkä ovat automatisoinnista seuraavat hyödyt. (Motivitylabs.)

Kun prosessin automatisointi on todettu kannattavaksi, suoritetaan prosessin perusteellinen analyysi. Analyysissä dokumentoidaan yksityiskohtaisesti, mitä tehtäviä prosessiin kuuluu, miten ne suoritetaan, mitä rajapintoja prosessissa tulee käyttää ja millaisia resursseja tarvitaan. Samalla tunnistetaan mahdolliset poikkeustilanteet ja niiden käsittelyprosessit. Tämä vaihe on kriittinen, sillä se luo perustan automatisoinnin kehitykselle. (Motivitylabs.)

Kun analyysissä tehty prosessin dokumentaatio todetaan hyväksytyksi, alkaa kehitysvaihe. Kehitysvaiheessa tehtävänä on luoda RPA-työkaluja hyödyntäen, automaattioratkaisu eli robotti, joka suorittaa kaikki dokumentaatiossa määritetyt työvaiheet. (Motivitylabs.)

Kehittämisen jälkeen seuraa hyväksyntätestausvaihe eli UAT (User Acceptance Testing). Tässä vaiheessa kehitettyä robottia testataan esituotantoympäristössä ja tarkoituksena on tunnistaa mahdolliset virheet ja selvittää toimiiko robotti halutulla tavalla. Mikäli testauksessa ilmenee virheitä, robotti palautetaan kehitysvaiheeseen, missä testauksen aikana havaitut virheet korjataan ennen uutta testausta. Kun robotti todetaan hyväksytyksi, siirrytään robotin käyttöönottovaiheeseen. (Sahiti 2023.)

Käyttöönottovaiheessa käydään läpi robotin ylläpitoon liittyvät asiat kuten kuinka usein robotti käynnistetään, kuinka kauan robotti saa pysyä päällä ja keneen otetaan yhteyttä, jos robotissa ilmenee virheitä. Kun ylläpidolliset asiat on käyty läpi, robotti siirretään tuotantoympäristöön. Tuotantoon siirron jälkeen seuraa robotin ylläpito ja seuraaminen, jolloin automatisoitavan prosessin elinkaari on valmis. Tämä elinkaari on havainnollistettu kuvassa 1. (Sahiti 2023.)



Kuva 1. Automatisoitavan prosessin elinkaari (Motivitylabs)



### 3 Blue Prism

#### 3.1 Blue Prism historia

Blue Prism on jo vuonna 2001 alkunsa saanut yritys, jonka perustivat David Moss ja Alastair Bathgate. Moss ja Bathgate perustivat yrityksen huomattuaan, kuinka paljon yritysmaailmassa, on yksinkertaisia tehtäviä, joiden tekemiseen voisi ihmisen sijaan valjastaa robotin. (Ron 2019.)

Blue Prismin ensimmäinen tuote Automate julkaistiin vuonna 2003 ja vuonna 2005 sen otti käyttöönsä englantilainen pankkialan yritys Co-operative Financial Services. Käyttöön-otosta seurasi 50 työntekijän siirtyminen toimistojen tietokoneilta lähikontaktissa käytävään asiakaspalvelutyöhön. (Warwick 2007.)

Blue Prismin omistaa 16.3.2022 tarjouskilpailun voittanut amerikkalainen monikansallinen holding yhtiö SS&C (Cision PR Newswire 2022). Nykyään yritys julkaisee tuotteita, jotka pääasiassa kantavat nimeä Blue Prism.

Blue Prismin tarkoituksena on luoda yrityksille virtuaalinen työvoima, jota on helppo hallinnoida ja jonka avulla voidaan kustannustehokkaasti korvata fyysinen työvoima. Korvattu fyysinen työvoima voidaan siirtää työtehtäviin, jotka oikeasti vaativat ajattelua, päätöksentekoa ja asiakasrajapinnassa työskentelyä. Kun digitaalinen työvoima on otettu käyttöön, ainoa ihmisresurssi, jonka työ vielä mahdollisesti vaatii, on robottiesimies. Robottiesimiehen tehtävänä on valvoa, että robotti tekee työt oikein ja puuttua robotin toimintaan, jos se alkaa toimia virheellisesti. (Blue Prism Docs 2023c.)

#### 3.2 Kehitys Blue Prismillä

Blue Prism tarjoaa low-code tyylistä kehitystä, joka käytännössä tarkoittaa, että kehittäjän ei tarvitse osata koodaamista voidakseen käyttää Blue Prismiä. Kokemus koodauksesta auttaa tietysti huomattavasti kehittämistyötä, mutta pääasiassa Blue Prismin käyttämiseen tarvitaan vain loogisen päättelykyvyn osaamista. (Katya 2023.)

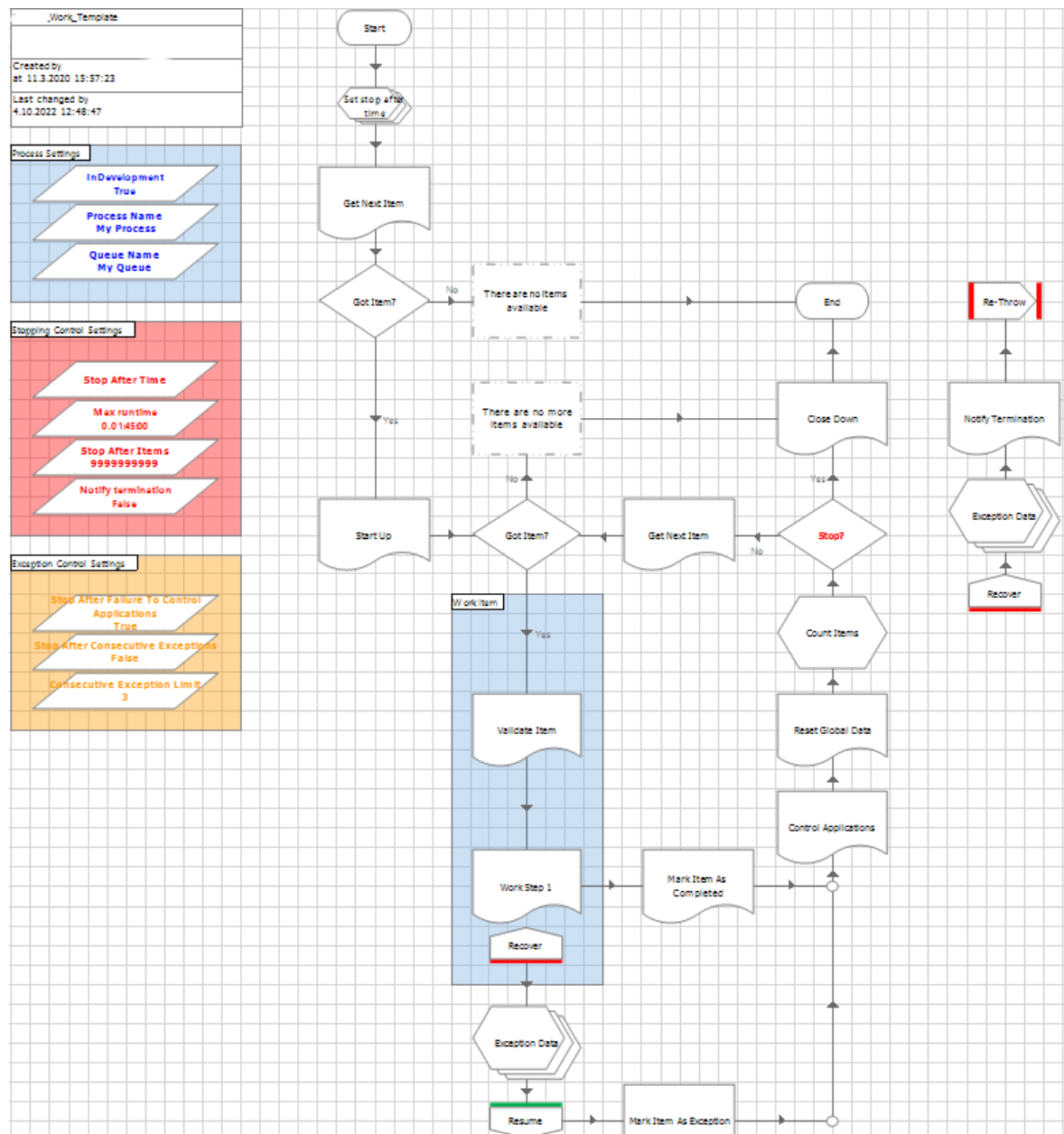
Blue Prismin käyttöliittymä on käyttäjäystävällinen ja intuitiivinen, mikä tekee siitä helposti lähestyttävän kehittäjille ja prosessien suunnittelijoille. Kehitykseen tarkoitettu käyttöliittymä on käytännössä vuokaaviomainen, mikä tarkoittaa, että prosessien rakentaminen ja automatisointi perustuu visuaaliseen suunnitteluun. Tämä tekee prosessien mallintamisesta ja määrittelystä ymmärrettävää ja saavutettavaa ilman syvällistä koodauksen osaamista. Visuaalinen lähestymistapa mahdollistaa prosessien hahmottamisen helposti ja

antaa kehittäjille mahdollisuuden rakentaa helposti ymmärrettäviä monimutkaisia automaatioprosesseja. (Blue Prism Docs 2023d.)

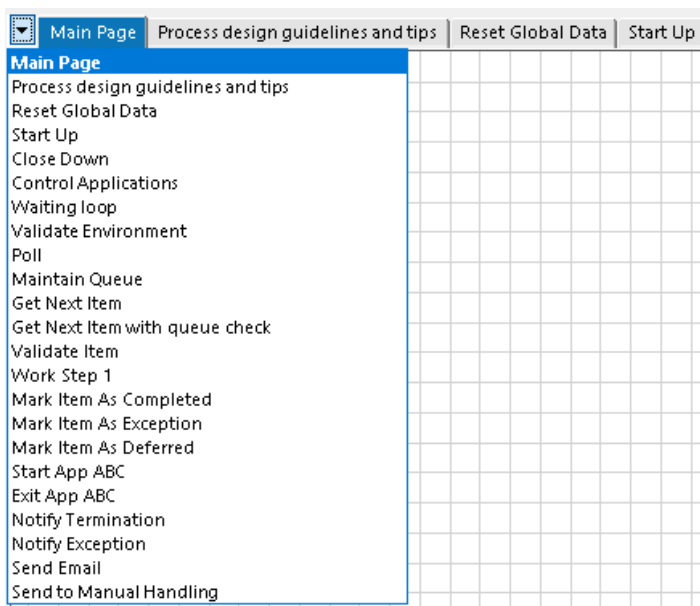
Tärkeää on myös, että Blue Prismin tarjoama low-code-kehitys ei rajoita automatisoitavan prosessin laajuutta tai monimutkaisuutta. Se soveltuu yksinkertaisista toistuvista tehtävistä aina laajojen liiketoimintaprosessien automatisointiin. Tämä tekee Blue Prismista monipuolisen työkalun organisaatioille, jotka haluavat hyödyntää RPA:ta liiketoimintaprosessien tehostamiseen ja virheiden vähentämiseen ilman syvää teknistä osaamista. (Simplilearn 2023.)

### 3.3 Work Template

Blue Prism tarjoaa oman Work Template -prosessin, jonka avulla on yksinkertaista lähteä kehittämään haluamaansa prosessia (Kuva 2; Blue Prism Portal 2017). Work Template -prosessi sisältää kaikki hallinnolliset sivut prosessin ylläpitoa ja työjonon käsittelyä varten (Blue Prism Portal 2017, 5). Hallinnollisiin sivuihin kuuluu sivut Main Page, Get Next Item, Control Applications, Start Up, Close Down, Mark item As Exception, Mark item As Completed, Reset Global Data (Kuva 3).



Kuva 2. Blue Prismin oman Work Template -prosessin Main Page -sivu



Kuva 3. Work Template -sivut

Main Page -sivu eli prosessin pääsivu ei itsessään tee töitä prosessissa, vaan sen sijaan kutsuu alisivuja, jotka tekevät kaiken tarvittavan työn. Tämä pitää pääsivun yksinkertaisena ja antaa korkean tason katsauksen siihen, mitä prosessi tekee. Pääsivulla on joitakin globaaleja ympäristöön, istuntoon ja jonoon liittyviä muuttujia. Pääsivulla tapahtuu myös prosessin ylimmän tason poikkeuskäsittely, joka käsittelee poikkeuksia, jotka nousevat esiin mistä tahansa alisivulta, kun työtä käsitellään. (Blue Prism Portal 2017, 5.)

Get Next Item -sivulla on toiminto, joka lukee pääsivun Queue Name -muuttujasta jonon nimen ja ottaa tästä jonosta ensimmäisen työn, lukitsee sen ja ottaa työn tiedot Item Data -tietueeseen. Item Data on globaali tietue ja sen näkyvyys ulottuu prosessin kaikille sivuille. (Blue Prism Docs 2023e.)

Control Applications -sivun tarkoituksena on käytettävien järjestelmien hallinnointi. Sivulla ei itsessään suoriteta mitään toimintoja, vaan kutsutaan järjestelmiä varten tehtyjä käynnistys- ja sulkemisalisivuja, sekä hoidetaan virheenkäsittely.

Alisivuja kutsutaan sen perusteella, millä järjestelmän hallinnointimuuttujan Control App arvolla Control Applications -sivulle on menty. Esimerkiksi prosessin alussa olevalla Start Up -sivulla määritetään käynnistettävien järjestelmien Control App -muuttujaan arvo START. Tämän jälkeen prosessi menee Control Applications -sivulle, jossa Choice-toiminto tunnistaa Control App -muuttujasta, että järjestelmä pitää käynnistää ja ohjaa prosessin halutun järjestelmän käynnistyspolkuun. Close Down -sivulla, taas Control App -muuttujiin laitetaan arvo EXIT jolloin, Control Applications -sivu ohjaa prosessin järjestelmien sulkemispolkuihin.

Mark item As Exception -sivun tarkoituksena on prosessin virheenkäsittely. Tämä sivu toimii pohjana, jolle kehittäjä voi rakentaa oman, virheen laadusta riippuvan, virheenkäsittelyn. Virheen käsittelyn jälkeen työ merkataan sivulla virheelliseksi. (Blue Prism Portal 2017, 7.)

Mark item As Completed -sivun tarkoituksena on merkata käsiteltävä jonotyö valmiiksi, kun prosessissa vaaditut työtehtävät on suoritettu. Kehittäjä voi tällä sivulla määrittää työn valmistumiseen liittyvät tiedot jonotyön Tag- ja Status-kenttiin. (Blue Prism Portal 2017, 7.)

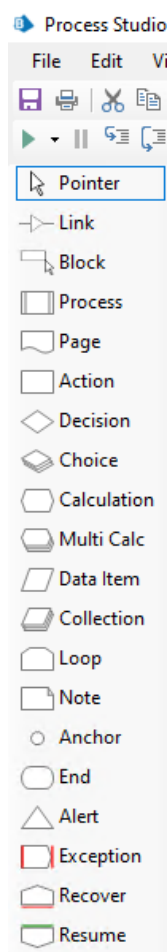
Reset Global Data -sivulle määritetään prosessia kehittäessä kaikki prosessissa käytettävät globaalit muuttujat, joita työn suorituksen aikana käytetään. Globaalit muuttujat ovat prosessin kaikille sivuille näkyviä muuttujia. Jos kehitettävässä prosessissa on muuttujia, joita tarvitaan usealla eri sivulla, on parempi määrittää muuttujat tälle sivulle, eikä kaikille sivuille erikseen. Tämä tehdään siksi, että muuttujien käyttäminen prosessissa pysyy selkeänä ja ymmärrettävänä. Kun prosessi menee Reset Global Data -sivun läpi, kaikki globaalit muuttujat nollaantuvat. Tarkoituksena on nollata muuttujat, jotta prosessi on niin sanotussa lähtötilanteessa ennen uuden työn ottamista. (Blue Prism Portal 2017, 8.)

### 3.4 Blue Prism -prosessi

Blue Prism -prosessi on rakenteeltaan jaettu sivuihin, jotka on jäsennelty pääsivulla eli Main Page -sivulla vuokaaviomaisesti suoritusjärjestykseen. Logiikka rakennetaan prosessin sivutasolla, ja kaikki kohdesovellukseen liittyvät toiminnallisuudet luodaan objektin tasolla. Prosessitason logiikan rakennukseen tarkoitetut toiminnallisuudet ovat kuvattuna kuvassa 4. (Blue Prism Docs 2023d.)

Prosessi koostuu alisivuista, jotka sisältävät toiminnallisen logiikan. Näillä alisivuilla on yleensä objektitason toimintoja, joilla suoritetaan haluttuja tehtäviä kohdesovelluksessa. Logiikka rakennetaan alisivulle näiden toimintojen ympärille. (Blue Prism Docs 2023g.)

Jokainen alisivu keskittyy tiettyyn tehtävään tai prosessin osaan ja yleensä nimetään suoritettavan tehtävän mukaisesti. Sivujen oikeanlainen nimeäminen mahdollistaa selkeän sekä tehokkaan prosessin hallinnan ja automatisoinnin. (Blue Prism Docs 2023h.)

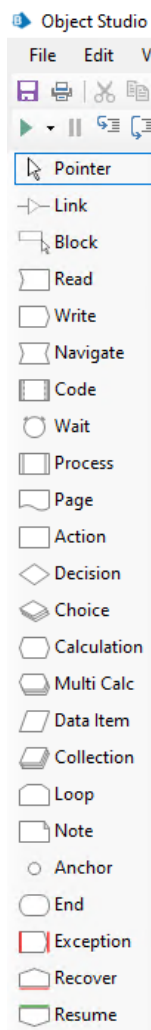


Kuva 4. Process Studio -toiminnot

### 3.5 Blue Prism -objekti

Objektit ovat olennainen osa Blue Prism -prosessin rakentamista. Objektien rakennus tapahtuu Object Studiossa, jossa on samankaltainen käyttöliittymä kuin prosessien rakentamiseen tarkoitettussa Process Studiossa. Erona on, että Object studiosta löytyy Application Modeller, jonka avulla halutut toiminnot mallinnetaan käytettävään ohjelmaan. Object Studiosta löytyy myös uusia toimintoja, joiden tarkoituksena on hallinnoida vuorovaikutusta kohdesovelluksen kanssa (Blue Prism Docs 2023i). Näihin toimintoihin kuuluu toiminnot Read, Write, Navigate, Code ja Wait. (Kuva 5).

Read- ja Write-toiminnoilla ohjataan tietojen hakemista sekä syöttöä. Navigate-toiminto mahdollista valikoiden avaamisen ja sivulta tai ohjelmistosta löytyvien nappien painamisen. Wait-toiminnolla voidaan varmistaa, että kohdesovellus on valmis seuraavaan toimintoon. Esimerkiksi ennen Navigate-toiminnon käyttämistä on hyvä varmistaa Wait-toiminnolla, että attribuutti johon navigointi kohdistuu, on latautunut kohdesovelluksessa. (Blue Prism Docs 2023i.)



Kuva 5. Object Studio -toiminnot

### 3.5.1 Application Modeller

Jotta prosessi tunnistaisi, mitä sen täytyy käytettävässä sovelluksessa tehdä, täytyy sovelluksessa työtehtäviin käytettävät elementit mallintaa Application Modellerilla. Application Modeller on Blue Prism -alustan työkalu, joka mahdollistaa käyttöliittymäelementtien, kuten painikkeiden, tekstikenttien ja valikoiden, mallintamisen ja määrittämisen automaatio-prosesseja varten. Tämän työkalun käyttäminen on keskeinen osa Blue Prism -prosessin kehittämistä, kun halutaan luoda tehokasta ja tarkkaa automaatiota eri sovellusten ja järjestelmien kanssa. (Blue Prism Docs 2023a.)

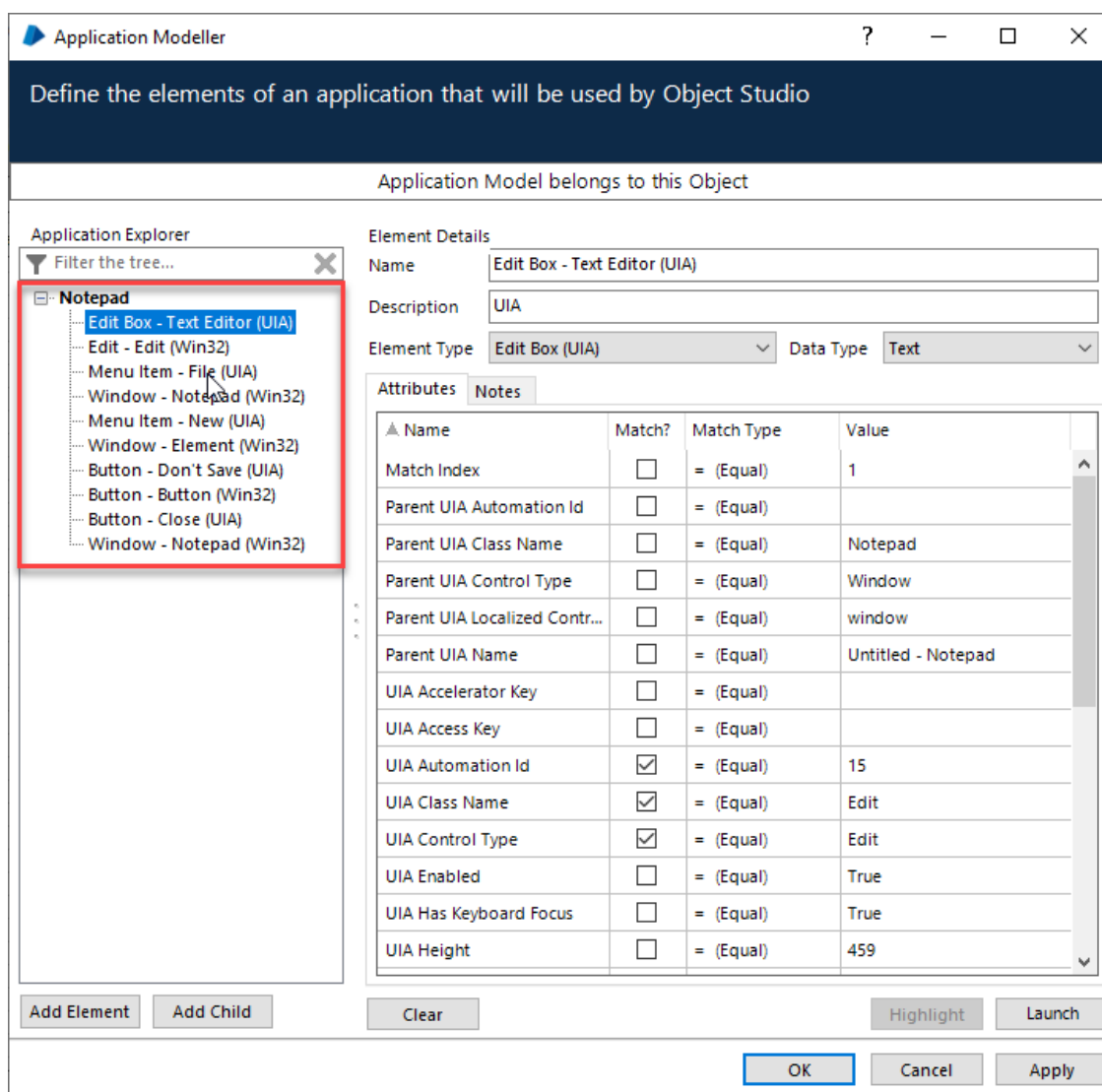
Tehtävän mallintaminen Application Modellerilla alkaa avattavan sovelluksen tiedostopolun määrittämisellä. Tämä voi olla esimerkiksi Windows-sovellus, verkkoselain tai mikä tahansa muu sovellus, johon automaatiota halutaan luoda. Jos tarkoituksena on mallintaa verkkosivun elementtejä, tulee avattavan verkkosivun URL-osoite määrittää Application Modelleriin sille tarkoitettulle riville, tiedostopolun alle. (Blue Prism Docs 2023b.)

Avaamisen jälkeen jokainen suoritettavaan työtehtävään haluttu sivu- tai sovelluselementti tulee mallintaa erikseen yhdeksi elementiksi. Mallintamisen jälkeen prosessi osaa löytää sovelluksesta kyseisen elementin, jonka jälkeen tälle elementille voidaan tehdä tarvittavia toimintoja. (Blue Prism Docs 2023b.)

Mallinnetut elementit tulisi nimetä niin, että nimestä on helppo tunnistaa, mikä elementti on kyseessä ja millä sivulla. Esimerkiksi jos mallinnetaan kohdesovelluksen lähetyssikkunassa oleva lähetä-nappi, olisi sopiva nimi tälle: "Lähetäikkuna - Painike - Lähetä". Yksinkertaisen ja selkeän nimeämisen avulla elementit on helppo tunnistaa ja lopuksi sijoittaa haluttuihin toimintoihin, kun objektin tehtävisivua aletaan rakentamaan. (Blue Prism Docs 2023a.)

Kun elementti on mallinnettu, tulee sille määrittää Match?-sarakkeesta tunnistettavat attributit. Näiden attribuuttien avulla Blue Prism osaa etsiä halutun elementin sovelluksesta (Kuva 6). Tunnistettavien attribuuttien määrittämisessä tulee olla erityisen tarkka varisinkin silloin, kun käytettävältä sivulta löytyy useampia samanlaisia elementtejä. (Blue Prism Docs 2023a.)



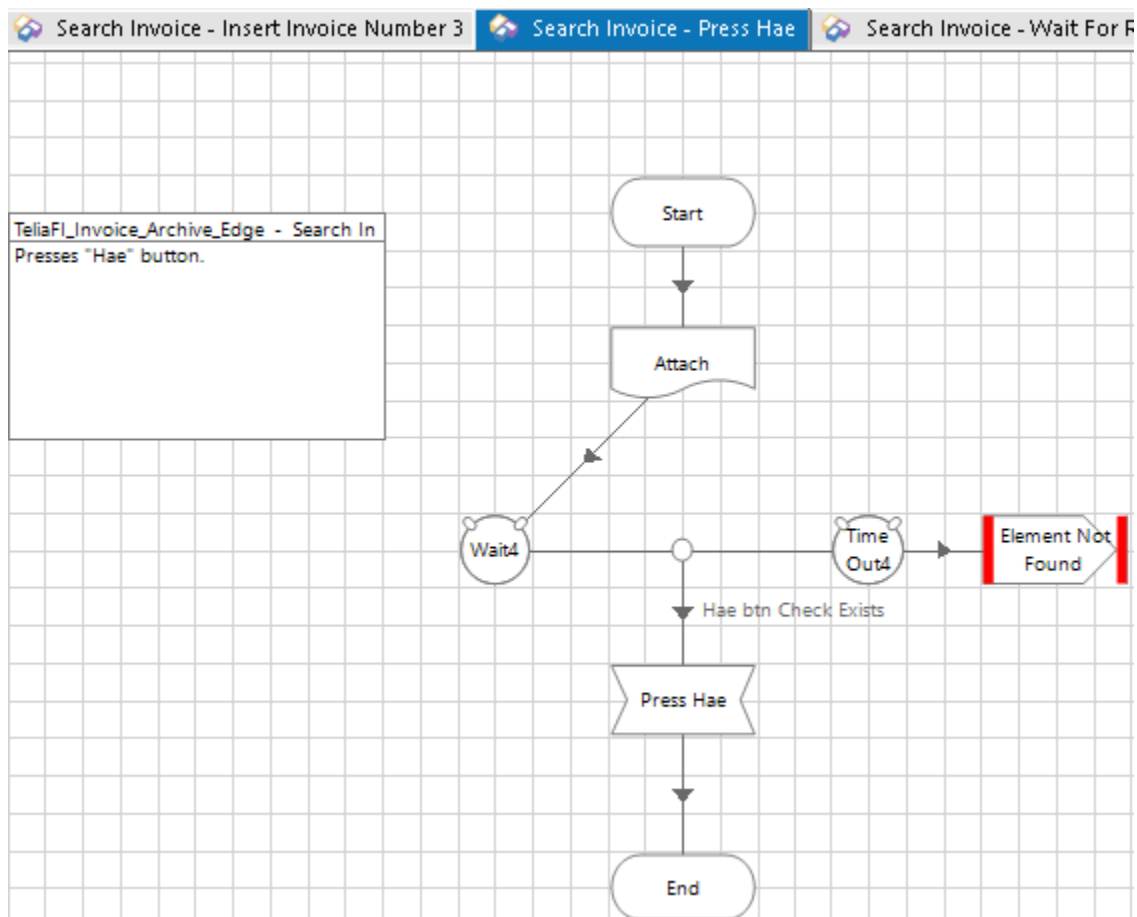


Kuva 6. Application Modeller, elementtien mallintaminen (Blue Prism Docs 2023b)

### 3.5.2 Objektin viimeistely

Kun kaikki vaaditut tehtävät on mallinnettu Application Modellerilla, tehdään jokaiselle työvaiheelle oma sivu, jonne mallinnettu tehtävä sijoitetaan. Mallinnetut tehtävät sijoitetaan joko Read, Write, Navigate, tai Wait -toimintoon, riippuen siitä, mitä kyseisellä työvaihesivulla halutaan tehdä. (Blue Prism Docs 2023i.)

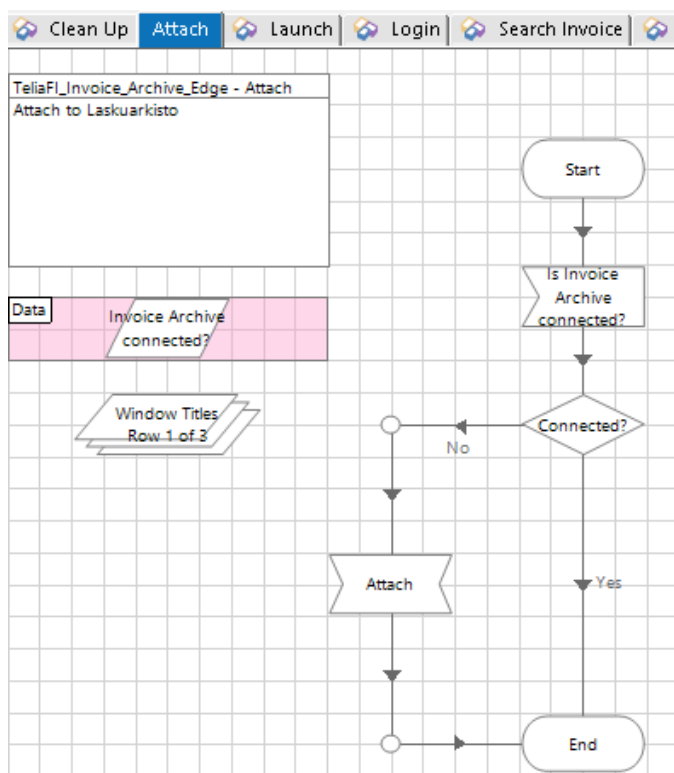
Objektin työvaihesivut tulee pitää mahdollisimman yksinkertaisina niin, että sivulla on vain viittaus objektin Attach-sivuun, sivulla suoritettavan tehtävän mukainen toiminto, sekä Wait-toiminto, joka menee virheeseen, jos käytettävää elementtiä ei löydy (Kuva 7). Tämä helpottaa lopullisen prosessin rakentamista. Jos objektisivulle rakentaa logiikkaa tai useamman toiminnon kokonaisuuden, voi lopullisen prosessin rakentamisesta tulla tarpeettoman monimutkaista. (Blue Prism Docs 2023i.)



Kuva 7. Objektin Press Hae -sivu

Attach-toiminnon tarkoituksena on auttaa Blue Prismia tunnistamaan oikea sovellus tai ikkuna. Tunnistamisen voi tehdä sovelluksen nimen, ikkunan otsikon tai muiden ominaisuuksien perusteella. Attach-toiminto voi sisältää odotusajan tai uudelleenyrityslogiikan varmistaakseen, että sovelluksen tunnistaminen onnistuu luotettavasti. Tämä on tärkeää, jotta automaatio voi toimia luotettavasti ja välttää virheet, sekä epäonnistumiset. Tämä toiminto on erityisen hyödyllinen silloin, kun automaatioprosessin on vuorovaikutettava useiden eri sovellusten tai järjestelmien kanssa saman prosessin aikana. (Kuva 8.)

Objektien rakentaminen Blue Prismissa vaatii tarkkuutta ja ymmärrystä käsiteltävistä sovelluksista ja järjestelmistä. Hyvin määritellyt objektit mahdollistavat tehokkaan ja tarkan automaation, joka voi vähentää manuaalista työtä ja parantaa liiketoiminnan tehokkuutta.



Kuva 8. Objektin Attach-sivu

### 3.6 Virheen käsittely

Virheen käsittelyä varten Blue Prismistä löytyy Process studion sekä Object Studion puolelta toiminnot Block, Exception, Recover ja Resume. Näitä toimintoja hyödyntämällä voidaan rakentaa virheen käsittelylogiikkaa, jossa tunnistetaan, missä kohtaa virhe on tullut, mikä on virheen syy ja mitkä ovat jatkotoimenpiteet. (Blue Prism Docs 2023j.)

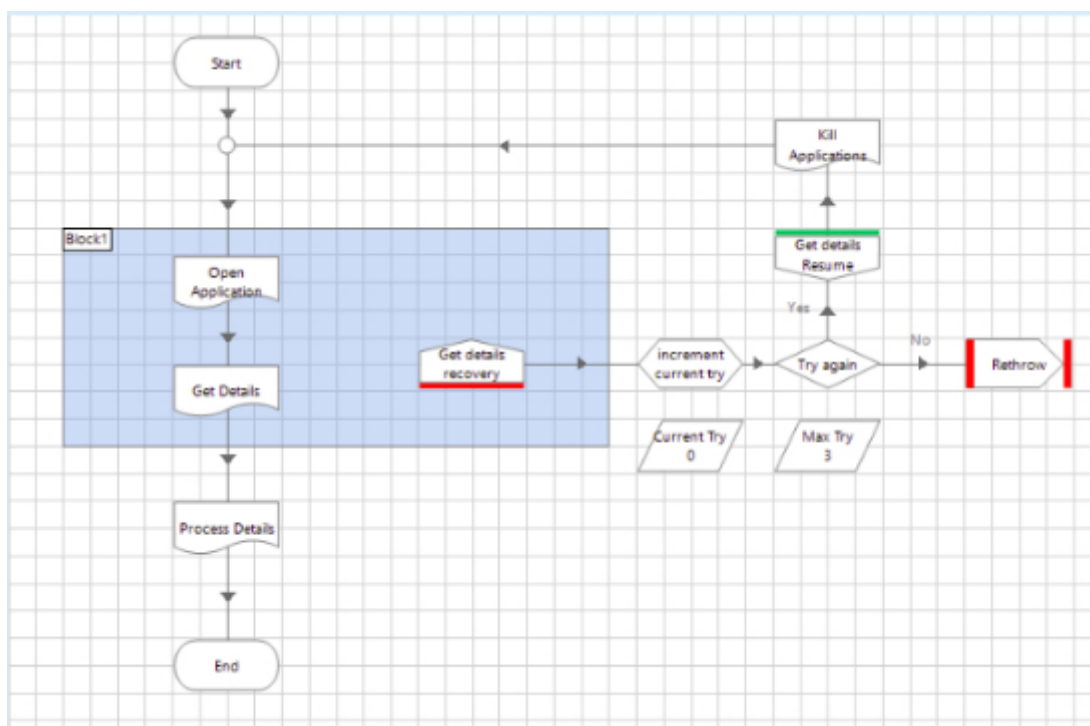
Block on sininen laatikko, jolla voidaan ryhmittää halutut toiminnot tai sivut yhteen, niin että kaikille laatikon sisällä oleville toiminnoille tehdään samat jatkotoimenpiteet virheen tapahtuessa (Kuva 9). Kun Block-laatikon sisällä olevassa toiminnossa tapahtuu virhe, prosessi hyppää saman laatikon sisällä olevaan Recover-toimintoon ja jatkaa siitä eteenpäin. (Blue Prism Docs 2023j.)

Exception-toiminnon voi laittaa haluamaansa kohtaan, jos tiedossa on, että tietyssä kohdassa prosessi on virheessä. Exception-toiminnon sisään voidaan määritellä virheen tyyppi Exception Type -kenttään ja virheviesti Exception Detail -kenttään. (Blue Prism Docs 2023j.)

### 3.7 Virhetiedot

Virheen tapahtuessa, prosessi menee Recover-toimintoon ja virhetiedot tallentuvat prosessin muistiin. Resume-toiminto poistaa prosessin virhetilasta. Virheen tiedot saadaan muistista `ExceptionType()` ja `ExceptionDetail()` -funktiolla. `ExceptionType()`-funktio antaa virheen tyyppin, joka voi olla joko Internal Exception, System Exception tai joku itsemääritetty virhetyyppi esimerkiksi Data Exception. (Blue Prism Docs 2023j.)

System Exceptionit ovat poikkeuksia, jotka liittyvät yleensä järjestelmään tai sovellukseen, jonka kanssa Blue Prism -prosessi vuorovaikuttaa. System Exceptioneita voi syntyä esimerkiksi silloin, kun sovellus kaatuu odottamattomasti tai kun käyttöliittymässä tapahtuu jokin odottamaton muutos (Tutorialspoint). Tällaisia järjestelmäpoikkeuksia voidaan käsitellä niin, että prosessiin lisätään virheenkäsittelylohkoja, jotka sisältävät esimerkiksi sovelluksen uudelleen käynnistys tai sivun päivittämisen (Kuva 9).



Kuva 9. Sovelluksen uudelleen käynnistys virheen tapahtuessa (Hopetutors)

Internal Exceptionit ovat poikkeuksia, jotka liittyvät enemmän itse Blue Prism -ympäristöön ja sen toimintaan. Tällaiset poikkeukset voivat johtua Blue Prism -prosessin ajonaikaisista ongelmista tai rajoituksista. Internal Exceptioneita voidaan käsitellä tarkentamalla prosessin rakennetta, kuten lisäämällä odotusaikoja, lisäämällä virheiden tunnistusta ja muokkaamalla prosessin loogista kulkua. (Tutorialspoint.)

Omaksi virhetyypiksi voi määritellä virheitä, jotka johtuvat esimerkiksi prosessissa ajettavasta datasta. Jos prosessissa ajettavassa työssä on virheellistä dataa ja tästä johtuen prosessi ei pääse etenemään kohdesovelluksessa, on turha yrittää työn ajoa enää uudelleen, koska lopputulos on aina sama. Tämänlaisia virheitä varten tarvitaan yleensä manuaalista käsittelyä ja tällaisen virheen tunnistessa on hyvä lähettää virheellinen työ määritelyyn manuaalikäsittelyyn. (Tutorialspoint.).

ExceptionDetail()-funktio antaa sattuneesta virheestä yksityiskohtaisemman tiedon kuten määritellyn virheilmoituksen tai virhekoodin. Tällä tiedolla voidaan auttaa diagnoosia ja virheenkorjausta. (Blue Prism Docs 2023j.)

### 3.8 Työjono

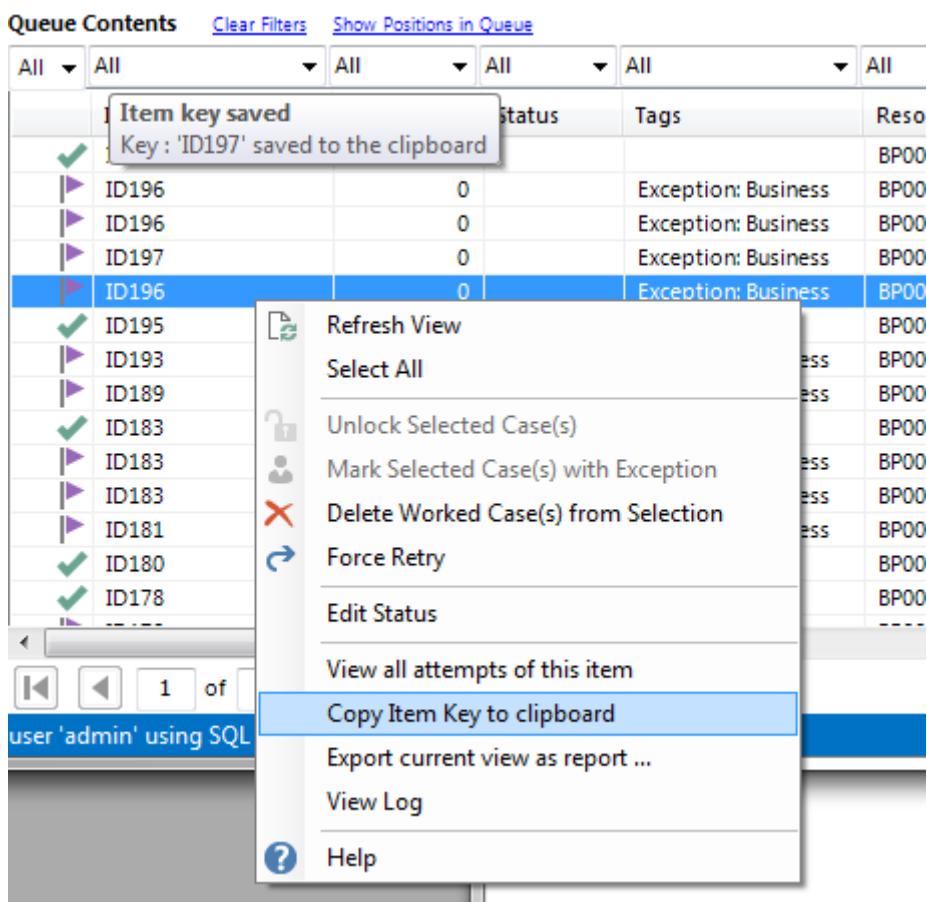
Työjono, eli Work Queue on keskeinen osa Blue Prismin automaatioalustaa. Työjonot mahdollistavat tehtävien hallinnan ja automaatioprosessien suorittamisen tehokkaasti ja skaalautuvasti. Prosessin työjono on eräänlainen tehtäväluettelo, johon voidaan lisätä tehtäviä, jotka automaatio suorittaa. (Blue Prism Docs 2023e.)

Jotta automaatioprosessi voi käyttää työjonoa, tulee prosessille luoda oma työjono Blue Prismin System Managerin kautta (Kuva 10). Työjonon luomiseen sisältyy jonon nimen sekä jonotyön avaimen nimen määrittäminen. (Blue Prism Docs 2023k.)

The screenshot displays the 'Workflow - Work Queues' window in Blue Prism. On the left, a tree view shows the 'Queues' folder expanded, with sub-items 'In Tray', 'Emails', 'Out Tray', 'Reports', 'Processing', and 'Invoices'. The 'Emails' queue is selected. The main area shows the 'Queue Detail' for 'Emails'. Fields include: Name (Emails), Key Name (Field 1), Maximum Attempts (1), Status (Running), and a 'Pause Queue' button. There is an 'Encrypted' checkbox and a 'using key' dropdown set to 'Default Encryption Scheme'. At the bottom, there are 'Assigned Process' and 'Assigned Resource Group' dropdowns, and an 'Active Queue' checkbox. Navigation buttons at the bottom include 'New', 'Delete', 'Find References', and 'Apply'.

Kuva 10. Prosessin työjonon luonti (Blue Prism Docs 2023f)

Jonotyöllä (Work Item) on niin sanottu oma erilaisia tietokenttiä sisältävä tietue eli Item Data, johon kaikki kyseiseen työhön liittyvä tieto tallennetaan. Jonotyön nimiavain eli Item Key määrittää, mistä tietokentästä jonossa olevan työn yksilöllinen tunnistenimi otetaan (Blue Prism Docs 2023e). Kun nimiavain on määritetty, jonotyö näkyy prosessin työjonossa omalla yksilöllisellä nimellään kuvan 11 mukaisesti.



Kuva 11. Prosessin työjonossa näkyvä nimiavain eli Item Key (Blue Prism Docs 2023k)

## 4 Automatisoitavan prosessin tunnistus

### 4.1 Prosessin arviointi

Automaattioratkaisun toteutus alkaa prosessin analyysillä, jonka esivaiheena on prosessin arviointi. Prosessin arvioinnissa käydään läpi, mikä on prosessin nykytilanne ja automaation tuottama tulevaisuuden näkymä. Nykytilanteessa käydään läpi, mitkä työprosessin vaikutukset tällä hetkellä ovat ja kuinka paljon resursseja prosessi nykyisellään vaatii. Tulevaisuuden arvioinnissa käydään läpi, mitkä automaation tuottamat vaikutukset ovat resurssien ja tehokkuuden kannalta. (Motivitalabs.)

Prosessin arviointiin kuuluu työhön liittyvien järjestelmien ja niihin vaadittavien tunnusten ja oikeuksien kartoittaminen sekä työn eri vaiheiden selvittäminen. Arvioinnin loppuvaiheessa käydään läpi työprosessiin liittyvien esimiesten ja käytettävien järjestelmien ylläpidosta vastaavien henkilöiden kanssa, onko työn automatisaatio kannattavaa tai sallittua. (Motivitalabs.)

Lopullisen painoarvon työn automatisoinnille antaa business case, joka lasketaan automatisoinnin luomista säästöistä ja tuotoista. Kun kaikki vaiheet on prosessin arvioinnin osalta suoritettu ja automatisointi on todettu kannattavaksi, alkaa prosessin tarkempi analyysi. (Motivitalabs.)

### 4.2 Prosessin analyysi

Analyysivaiheessa pyritään ensin saamaan kattava käsitys siitä, miten prosessin työtehtävät suoritetaan manuaalisesti. Tämä tarkoittaa prosessin yksityiskohtaista ja perusteellista kartoitusta yhdessä niiden työntekijöiden kanssa, jotka tällä hetkellä hoitavat kyseessä olevia työtehtäviä. Tavoitteena on kirjata ylös jokainen vaihe, toimenpide sekä variaatio, jotka liittyvät kyseiseen työprosessiin. Käytännössä tämä tarkoittaa jokaisen klikkauksen, syötetyn tiedon ja vuorovaikutuksen tarkkaa dokumentointia. (Abhimanyu 2022.)

Kun manuaalinen työprosessi on tarkasti kuvattu, seuraava vaihe on muuntaa tämä kuva robotille sopivaksi. Tämä edellyttää usein prosessin yksinkertaistamista ja selkeyttämistä, jotta robotti voi suorittaa tehtävät tehokkaasti. Yksityiskohdat, kuten tarvittavat syötteet, oletetut tulokset ja mahdolliset poikkeukset, otetaan huomioon tässä vaiheessa. Tavoitteena on luoda ohjeet, jotka ovat yhtä yksityiskohtaiset ja täsmälliset kuin manuaalisten työntekijöiden ohjeet. (Abhimanyu 2022.)

Robotti tulee tekemään työn juuri niin kuin ihminenkin sen tekee, joten ohjeiden pitää olla sen mukaiset. Tarkkaan tehty työprosessin dokumentaatio takaa suoraviivaisen RPA-

prosessin kehityksen ja vähentää kehitysvaiheessa tehtävien jatkoselvitysten määrää.  
(Abhimanyu 2022.)



## 5 Laskuarkiston virheilmoituksen automatisointi

### 5.1 Alustus

Automaatiovinkkikanavaa pitkin on tullut ehdotus työtehtävästä, jonka automatisoinnin mahdollisuutta halutaan selvittää. Automaatio koskee työprosessia, joka suoritetaan yrityksen laskuarkistossa. Työprosessin automatisointi arvioidaan kannattavaksi, koska työprosessin automatisoinnista aiheutuvat resurssien säästöt ovat merkittävät ja prosessi on suhteellisen yksinkertainen ja nopea kehittää.

Analyysissä laaditaan manuaalisesta työprosessista kuvaus, joka yksinkertaistetaan robotin suorittaman prosessin kuvaukseksi. Kun prosessi on saatu kuvattua robotille sopivaksi, alkaa kehitystyö.

#### 5.1.1 Manuaalisen prosessin kuvaus

Työntekijälle tulee päivittäin laskuarkistosta Excel-raportti, joka sisältää asiakkaita, joiden puhelinnumero yrityksen asiakasportaaliin vaatii päivitystä. Työntekijä käy Excelistä rivi kerrallaan jokaisen asiakkaan läpi ja lähettää hänelle laskuarkistosovelluksen kautta laskukopion, johon on lisätty saatetekstiksi kehoitus puhelinnumeron päivittämisestä asiakasportaaliin. Jotta asiakaspalvelu saa tiedon kyseisestä toimenpiteestä, kirjaa työntekijä laskukopion lähetyksen jälkeen yrityksen CRM-järjestelmään asiakkaan tietoihin, että asiakkaalle on lähetetty laskukopio kehoitusviestillä.

Tarkoitus on saada asiakasportaaliin asiakkaan päivitetty puhelinnumero, jotta asiakasta voidaan tiedottaa SMS-viestillä asiakasportaaliin saapuneesta laskusta. Laskukopio saapuu asiakkaalle sähköpostiviestinä.

#### 5.1.2 Automatisoitavan prosessin kuvaus

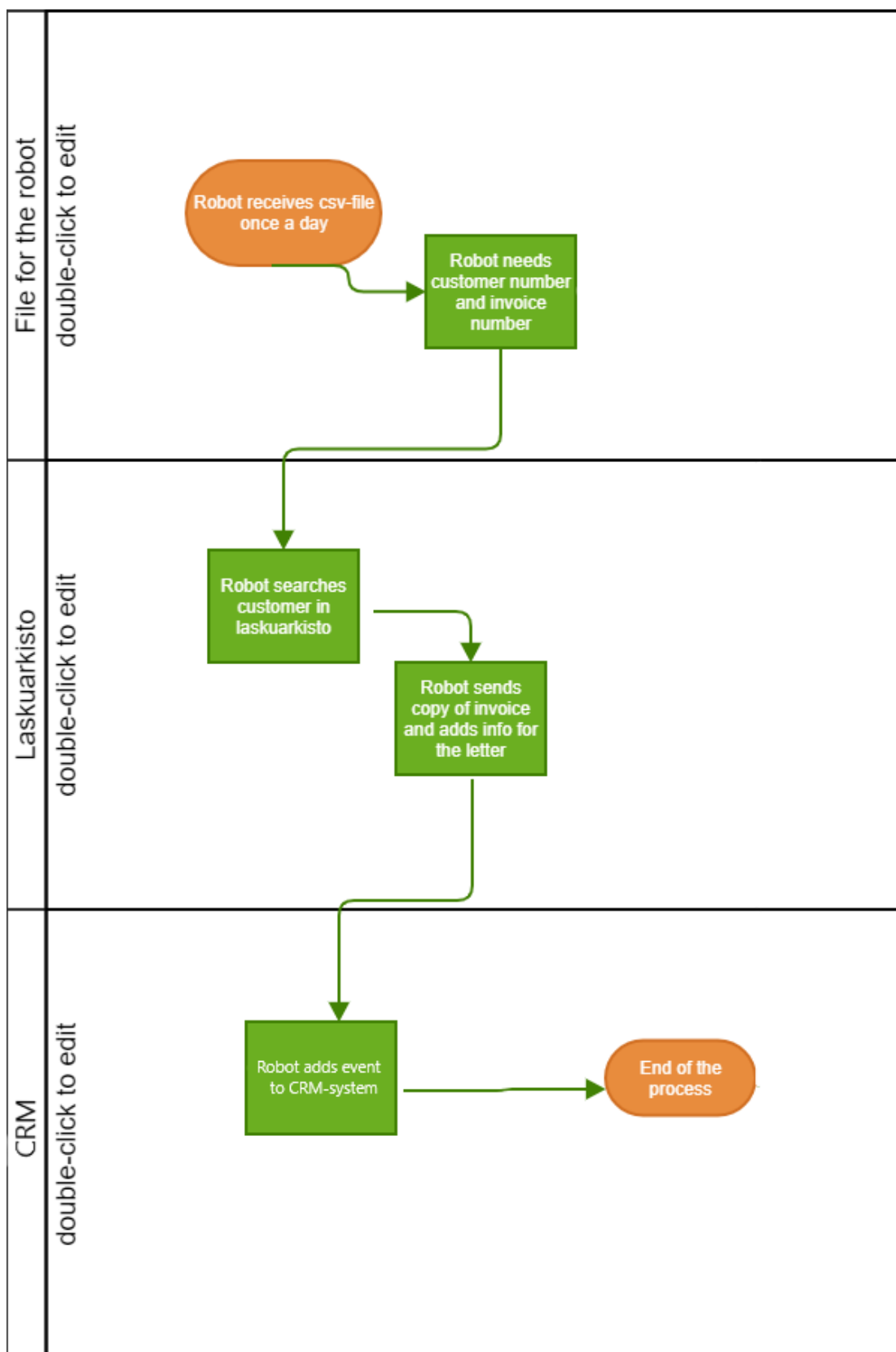
Ohjelmistorobotin toiminnasta on laadittu yksinkertaistettu vuokaavio kuvassa 12. Robotti saa laskuarkistosta päivittäin Excel-raportin verkkolevylle, jossa jokainen rivi sisältää laskun tyypin, asiakkaan asiakasnumeron, laskunumeron, henkilötunnuksen sekä nimen. Robotti tarvitsee raportista vain asiakkaan asiakastunnuksen, laskunumeron ja laskun tyypin. Robotti kerää raportista kaikki laskut, jonka jälkeen robotti avaa laskuarkiston ja hakee laskunumerolla oikean laskun. Kun oikea lasku on avattu, robotti valitsee tarvittavat valintaruudut valituiksi, lisää saatetekstin ja painaa lähetä-painiketta.

Laskuarkisto-osuuden jälkeen robotti avaa yrityksen CRM-järjestelmän, josta se etsii asiakastunnuksella oikean asiakkaan ja kirjaa tämän tietoihin, että asiakkaalle on lähetetty

laskukopio, jossa asiakasta on kehoitettu päivittämään puhelinnumeronsa asiakasportaaliiin. Käsittelyn jälkeen robotti poistaa Excel-raportin.

Robotti käsittelee vain laskut, joihin on merkattu tyyppitieto NORMAL. Jos raporteissa saapuu muun tyyppisiä laskuja, robotti kerää ajon aikana näistä laskuista listan ja lähettää kerätyn listan manuaalikäsittelyyn. Saapuvissa raporteissa ei välttämättä ole yhtäkään käsiteltävää laskua. Jos tällainen tyhjä raportti löytyy, robotti poistaa raportin, sulkeutuu ja jää odottamaan seuraavaa käynnistysaikaa.

Analyysissä määritetään, että Excel-raportti saapuu robotille joka päivä klo 11:00, joten robotti täytyy käynnistää klo 11:00 jälkeen. Robotti ajastetaan käynnistymään joka päivä klo 12:00.



Kuva 12. Prosessin kuvaus vuokaaviona

## 5.2 Järjestelmien kartoitus

Järjestelmiä, joita robotin tulee osata käyttää, ovat Excel, laskuarkisto ja yrityksen CRM-järjestelmä. Excelin käsittelyä varten Blue Prism tarjoaa omat työkalut eli Business Objectit. Yritysportaalia on mahdollista käyttää rajapinnan avulla, mutta laskuarkistoa ei. Laskuarkistoa varten tarvitsee siis rakentaa oma Business Object.

Laskuarkistoa käytetään Edge-selaimen kautta, joten rakennettavaan objektiin pitää määrittää kaikki toiminnot, mitä laskuarkistossa tehdään. Laskuarkistoon sisäänkirjautumista varten tarvitaan robotille tunnukset, joilla robotti voi kirjautua laskuarkistoon.

Käytettävää CRM-järjestelmää varten on jo olemassa objekti, joka käyttää rajapintaa. Kyseessä olevaa objektia voidaan hyödyntää asiakastietojen haussa ja tiedon kirjaamisessa asiakastietoihin.

## 5.3 Prosessin luonti

Prosessin luonti aloitetaan Blue Prismin tarjoamalla Work Templatella, jossa on kaikki hallinnolliset sivut valmiina. Työprosessiin liittyvät toiminnalliset sivut, jotka prosessiin tullessaan lisäämään, ovat Load Queue, Invoice archive usage ja Add event to CRM.

Prosessin luonnin yhteydessä tehdään prosessille oma työjono, johon robotti lisää ja josta se hakee töitä. Työjono nimetään prosessin mukaisesti niin, että nimen loppuun tulee teksti \_Queue. Jonossa olevan jonotyön nimiavaimeksi määritetään kenttä BPN. Tämä tarkoittaa, että jokainen jonotyö nimetään työn Item Datasta löytyvän asiakastunnuksen mukaan. Työt tulevat näkymään jonossa kuvan 13 mukaisesti.

<input type="checkbox"/>	State	Key Value	Status	Tags	Attempt	Loaded	Finished	Deferred
	All	Search by Key	All	Search by Tags	All	Search by Loaded Date	Search by Finished Date	Search by Deferred Date
<input type="checkbox"/>	✓	1000264	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:29:17	
<input type="checkbox"/>	✓	3001080	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:32:54	
<input type="checkbox"/>	✓	3001141	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:33:37	
<input type="checkbox"/>	✓	3001904	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:32:10	
<input type="checkbox"/>	✓	3002165	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:30:43	
<input type="checkbox"/>	✓	3002334	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:35:47	
<input type="checkbox"/>	✓	3002334	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:31:26	
<input type="checkbox"/>	✓	3002336	Message sent	NORMAL	1	2023-10-27 11:27:26	2023-10-27 11:30:00	

Kuva 13. Prosessin työjonossa näkyvät työt

Ennen uusien sivujen luontia täytyy määrittää prosessin hallinnolliset toiminnot, kuten työjonon hallinta ja järjestelmien käyttö. Prosessille tehty työjono määritetään prosessin pääsivulla sijaitsevaan muuttujaan Queue Name. Tämän muuttujan avulla sivulla Get Next

Item sijaitseva toiminto osaa hakea oikeasta jonosta työn, jonka prosessi lukitsee käsitte-  
lyn ajaksi.

Ainoa prosessissa käytettävä järjestelmä, joka robotin täytyy käynnistää, on laskuarkisto. Koska käytössä on Process Template, sivulta Control Applications löytyy valmis järjestelmien hallintakaavio, johon tarvitsee lisätä vain laskuarkiston käynnistäminen sekä sulkeminen. Laskuarkiston käynnistykseen sisältyy Edge-selaimen käynnistäminen, laskuarkiston nettisivulle navigointi sekä sisäänkirjautuminen. Laskuarkiston sulkemiseen riittää Edge-selaimen sulkeminen.

Koska työ robotille tulee Excel-tiedostona verkkolevylle, pitää prosessiin rakentaa Load Queue-sivu, jossa robotti hakee Excel-tiedoston määritetyltä verkkolevyltä rajapinnan avulla ja tallentaa sen tietokoneen Temp-kansioon. Kun tiedosto on haettu, robotti avaa ja lukee Excel-tiedoston ja tekee jokaisesta rivistä oman jonotyön ja tallentaa tämän jonotyön haluttuine tietoineen prosessin työjonoon.

Invoice archive usage -sivulle tullaan rakentamaan kaikki laskuarkiston käsittelyyn vaadittava logiikka ja Add event to CRM -sivulle CRM-järjestelmän käsittelyyn vaadittava logiikka. Ennen kuin laskuarkiston käsittelyä mahdollistavan logiikan voi rakentaa, tulee luoda objekti laskuarkiston toiminnallisuuksia varten.

#### 5.4 Objektin luonti

Laskuarkiston käyttöä varten luodaan objekti, johon suoritettavat toiminnallisuudet tullaan mallintamaan Application Modelleria hyödyntäen. Toiminnallisuudet, jotka laskuarkistossa suoritettavien tehtävien kannalta pitää mallintaa ja luoda logiikka ovat seuraavat:

- laskuarkistoon sisäänkirjautuminen
- laskunumeron syöttäminen laskunumerokenttään
- HAE-painikkeen painaminen
- Lähetä laskukopio -painikkeen painaminen, joka avaa laskukopion lähetyssivun
- valintalaatikoiden tikkaaminen
- viestin lisääminen saateteksti kenttään
- Lähetä-painikkeen painaminen.

Objektin nimeksi tulee Invoice\_Archive\_Edge. Edge-pääte nimen lopussa kuvaa, että objektin toiminnallisuudet on mallinnettu Edge-selaimelle.

## 5.5 Mallinnus- ja objektisivujen rakentaminen

Ennen mallintamisen aloittamista luodaan objektiin sivu Attach. Attach-sivulle lisätään Read-toiminto, joka vertaa selaimen valittuna olevan välilehden nimeä laskuarkiston tunnistettuihin nimiin. Vertaamisen jälkeen toiminto palauttaa muuttujaan True- tai False -vastauksen, riippuen siitä onko välilehden nimi oikea. Tämän jälkeen tulee Choice-toiminto, joka katsoo muuttujasta, onko vastaus True vai False. Jos vastaus on True voidaan todeta, että käytettävä välilehti on oikea ja logiikka tulee pois Attach-sivulta. Jos vastaus on False, logiikka menee Navigate-toimintoon, joka yrittää avata oikean nimisen välilehden. (Kuva 8, 15.)

Laskuarkistoon sisäänkirjautumista varten mallinnetaan kirjautumissivun Käyttäjätunnus- sekä salasana kenttä. Mallintamisen jälkeen objektiin luodaan sivu Login. Objektisivulle lisätään Write-toiminnot, jotka kirjoittavat muuttujissa olevat tiedot kirjautumissivun käyttäjätunnus- ja salasana kenttään.

Laskun etsimistä varten mallinnetaan siihen vaadittava laskunumerokenttä sekä HAE-painike (Kuva 14). Mallintamisen jälkeen objektiin luodaan sivut Insert invoice number ja Press Hae. Insert invoice number -sivulle lisätään Write toiminto, joka kirjoittaa muuttujassa olevan tiedon hakukenttään. Press Hae -sivulle lisätään Navigate-toiminto, joka painaa mallinnettua hakupainiketta.

Laskuttava yritys:

Asiakastyyppi:

Laskunumero:

Asiakasnimi:

Asiakasnumero:

Y-tunnus:

Laskun päiväys:  -

Eräpäivä:

Liittymänumero:

Henkilötunnus:

Käyttäjänimi:

Yksilöintitieto:

Voit rajata hakua valitsemalla lähtöjärjestelmän, jonka laskuihin haku kohdistetaan, laskuttavan yrityksen ja/tai asiakastyypin

Molemmat muodossa pp.kk.vvvv

Muodossa pp.kk.vvvv

Haku kohdistetaan kahden viimeisimmän vuoden laskuihin ☒ Poistamalla valinnan ruudusta saat hakutuloksena yli kaksi vuotta vanhempia laskuja

**HAE** TYHJENNÄ

Kuva 14. Laskuarkiston hakusivun mallinnettavat elementit

Kun lasku on löydetty, tulee robotin painaa avautuneelta laskusivulta Lähetä laskukopio -painiketta. Tätä varten mallinnetaan kyseinen painike ja luodaan sivu Click lähetä laskukopio. Sivulle lisätään Navigate-toiminto, joka painaa kyseistä mallinnettua painiketta.

Lähetä laskukopio -painikkeen painamisesta aukeaa uusi sivu, jossa robotin tulee valita tietyt valintalaatikot. Tarvittavat valintalaatikot mallinnetaan ja jotta objekti pysyy mahdollisimman skaalautuvana, jokaista valintaa varten luodaan oma sivu. Näille sivuille lisätään Navigate-toiminnot, jotka painavat tarvittavia valintalaatikkoja.

Laatikoiden valitsemisen jälkeen aukeaa saateviesti tekstilaatikko, jonne täytyy kirjoittaa viesti puhelinnumeron päivittämisestä. Mallinnetaan tekstilaatikko ja luodaan objektiin sivu Write saateteksti. Sivulle lisätään Write-toiminto, joka kirjoittaa muuttujassa olevan tekstin mallinnettuun tekstikenttään.

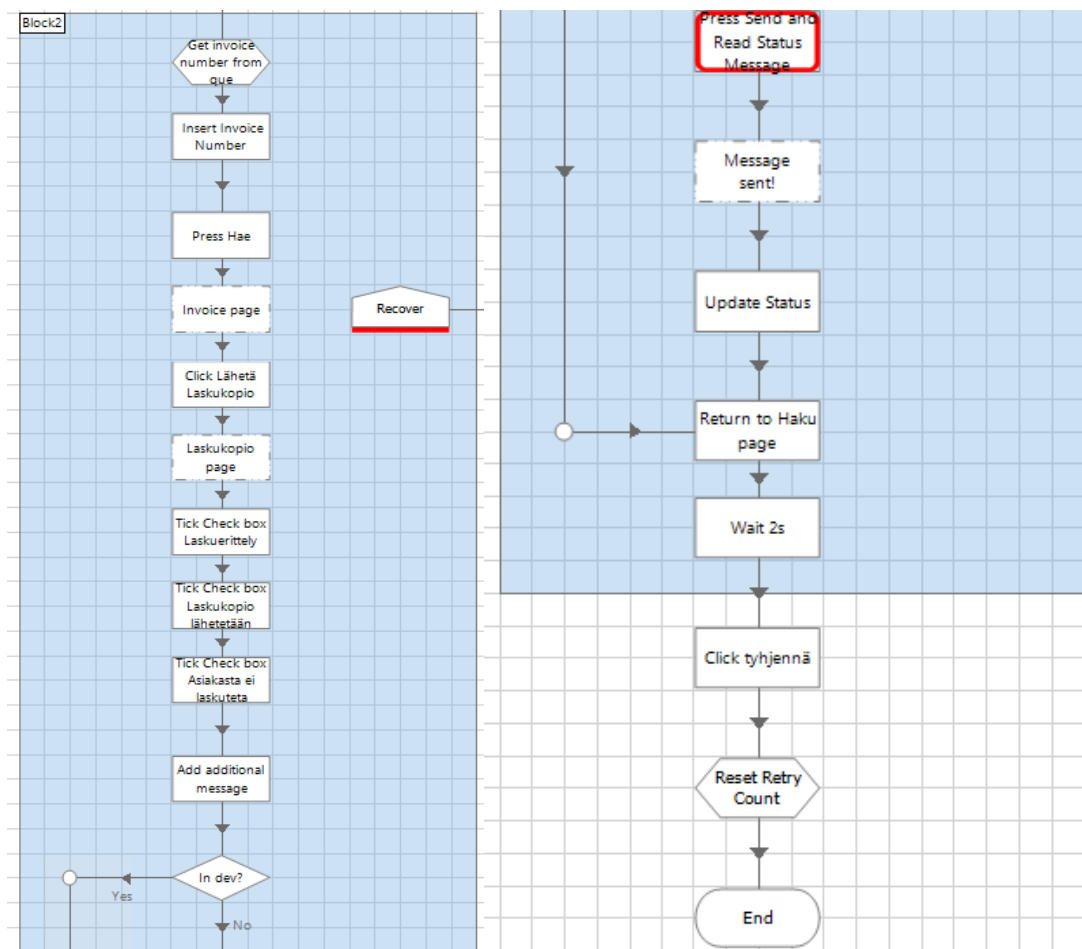
Yllämainittujen tehtävien suorittamisen jälkeen painetaan Lähetä-painiketta. Mallinnetaan siis sivun alalaidassa oleva Lähetä-painike ja luodaan sivu Press Lähetä, jonne lisätään Navigate-toiminto, joka painaa kyseistä mallinnettua painiketta.

Jokainen luoduista sivuista sisältää suoritettavan logiikan alussa viittauksen Attach-sivuun, joka varmistaa, että käytettävä järjestelmä on tunnistettu. Jokaisella sivulla on myös ennen jokaista käytettävää toimintoa Wait-toiminto, jonka tarkoituksena on etsiä sivulta haluttu elementti ennen kuin sitä käytetään. Jos käytettävää elementtiä ei löydy, objekti menee virhetilaan. Virhetila ilmoittaa, miltä objektin sivulta ja mistä etsittävästä elementistä virhe on tullut. Objektin sivuilla käytettävien muuttujien arvot tulevat prosessin puolelta.

## 5.6 Laskuarkiston käytösivu

Kun laskuarkiston käyttöä kuvaavat toiminnalliset sivut on saatu rakennettua objektiin, on prosessin puolelle mahdollista lisätä halutut toiminnallisuudet erillisinä rakennuspaloina. Paloista rakennetaan prosessin Invoice archive usage -sivulle suoraviivainen logiikka, jossa robotti lähettää asiakkaalle laskukopion halutulla saatetekstillä.

Lopullinen laskuarkiston käyttöön tarkoitettu sivu prosessin puolella on kuvattuna kuvassa 15. Laskuarkiston käyttöön tarkoitettujen toiminnallisuuksien lisäksi sivulle on lisätty Wait-toiminto ja Note-laatikoita. Wait-toiminnon tarkoituksena on odottaa 2 sekuntia, jotta varmistetaan, että Haku-sivu on latautunut ennen sivulla olevaa Tyhjennä-painikkeen painamista. Note-laatikot ovat katkoviivaisia laatikoita ja niistä jää jälki prosessin suorituksen aikana tapahtuvaan lokikirjaukseen. Jäljen jättäminen auttaa virheenkorjausta, jos tulevaisuudessa tarvitsee selvittää, millä sivulla on tapahtunut virhe.

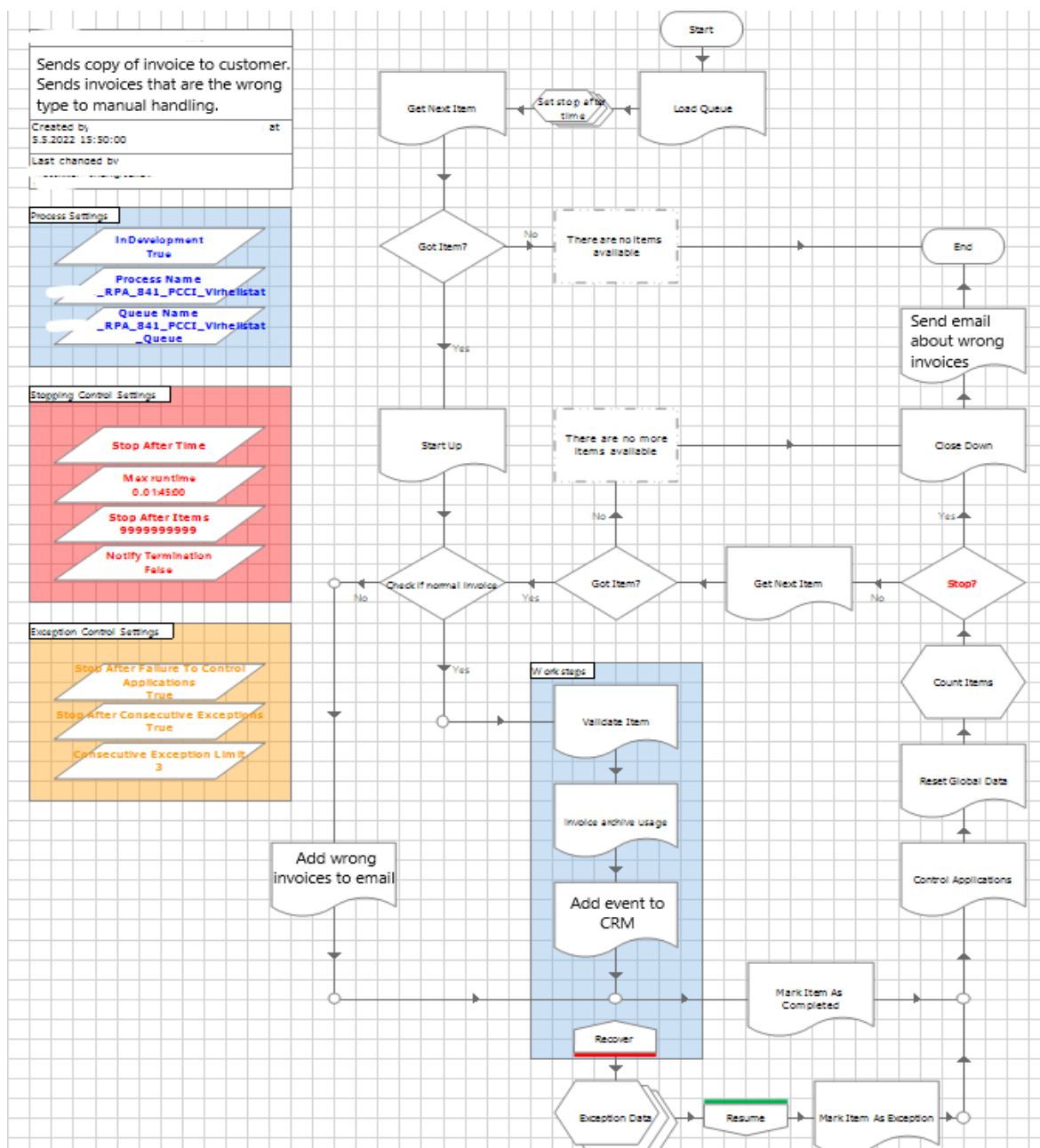


Kuva 15. Prosessiin luotu logiikka



## 5.7 Lopullinen prosessi

Lopullisen prosessin pääsivulle muodostui kuvan 16 mukainen, silmukkamainen vuokaavio. Prosessin alussa Load Queue -sivulla ladataan käsiteltävät laskut työjonoon. Ensimmäinen työ otetaan jonosta Get Next Item -sivulla ja Laskuarkisto käynnistetään Start up -sivulla. Tämän jälkeen prosessissa alkaa itse työn suorittamiseen tarkoitettu silmukka, jota robotti käy läpi niin kauan, kun työjonossa on töitä.



Kuva 16. Prosessin pääsivu

Prosessin alun lisäksi, myös työsil mukassa on Get Next Item -sivu, jossa robotti ottaa uuden työn sisään. Got Item niminen Choice-toiminto vertaa onko jonosta löydetty työ.

Jos työ on löydetty, prosessi menee Work Stages nimiseen siniseen laatikkoon, jossa sijaitsevat itse työhön tarkoitetut toiminnalliset sivut.

Validate Item -sivu tarkistaa löytyykö työn Item Data -tietueesta kaikki prosessin suorittamiseen vaadittavat tiedot. Kun tiedot on tarkistettu, suorittaa robotti Invoice archive usage -sivulla laskuarkistossa tehtävät työvaiheet. Laskuarkiston käytön jälkeen robotti kirjaa CRM-järjestelmään vaadittavat tiedot sivulla Add event to CRM.

Kun kaikki työvaiheet on suoritettu onnistuneesti, robotti merkkää jonotyön valmiiksi Mark Item As Completed -sivulla. Merkkaamisen jälkeen silmukassa tarkistetaan Control Applications -sivulla, onko laskuarkiston käyttöön tullut jotain erityisiä ohjeita ja resetoidaan prosessin globaalit muuttujat Reset Global Data -sivulla.

Kun töitä ei enää ole tai kun maksimi suoritus aika on saavutettu, robotti sulkee Close Down -sivulla laskuarkiston. Kun laskuarkisto on suljettu, sivulla Send email about wrong invoices tarkistetaan, onko prosessin aikana löytynyt vääränlaisia laskuja. Jos tällaisia laskuja on löydetty, robotti lähettää niistä kootun listan manuaalikäsittelijöille. Kun vääränlaiset laskut on tarkistettu, prosessi päättyy robotin osalta.

## 5.8 Lopullisen prosessin virheenkäsittely

Tässä prosessissa kaikki itsemääritetyt virheet ovat Business Exceptioneita. Jos prosessi ajautuu virheeseen, joka tyypiltään on Business Exception, jonotyötä ei yritetä ajaa uudelleen ja virheestä lähetetään viesti manuaalikäsittelyyn. Viestiin sisällytetään kaikki jonotyön tiedot sekä tieto tapahtuneesta virheestä.

Muiden virheiden kuten System Exceptioneiden ja Internal Exceptioneiden osalta virheenkäsittely on määritetty niin, että tapahtuneista virheistä lähetetään vain ilmoitusviesti, että prosessi on ajautunut virheeseen. Viestin lähetyksen jälkeen jonotyö ajastetaan takaisin jonoon, niin että prosessi voi yrittää sen uudelleenajoa.

Kuten kuvan 16 alaosassa näkyy, työvaiheisiin liittyvät sivut on ryhmitetty Work Stages nimisen sinisen laatikon sisälle. Jos jollain näistä työsivuista tulee virhe, prosessi siirtyy laatikon sisällä olevaan Recover-toimintoon. Recover-toiminnon jälkeen otetaan prosessin virhetiedot talteen Exception Data -toiminnolla, joka tallentaa ExceptionType() ja ExceptionDetail() -funktioista saadut tiedot Reset Global Data -sivulla sijaitseviin globaaleihin virhetietomuuttujiin.

Kun virhetiedot on otettu talteen, Resume-toiminto palauttaa prosessin virhetilasta ja prosessi siirtyy Mark Item As Exception -sivulle. Tällä virheenkäsittelysivulla verrataan onko tapahtunut virhe Internal Exception, System Exception vai Business Exception. Verrattava

tieto saadaan globaaleista muuttujista, joihin virhetieto tallennettiin aiemmin. Analyysissa määritetyn dokumentaation mukaisesti, Internal ja System Exceptionit ajetaan uudestaan ja virheistä lähetetään tieto ylläpidolle. Business Exceptionit ohjataan manuaalikäsittelyyn ja uudelleen ajoa ei yritetä.

Mark Item As Exception -sivun jälkeen prosessissa tapahtunut virhe on käsitelty ja vaadittavat toimenpiteet on suoritettu. Tämän jälkeen prosessi palaa tehtävien suoritussilmukkaan ja uusi työ otetaan jonosta.

On mahdollista, että prosessissa tulee virheitä muillakin sivuilla kuin Work Stages -laatikon sisällä olevilla sivuilla. Nämä virheet aiheuttavat prosessin suoran kaatumisen ja Blue Prism merkkää työn automaattisesti virheeksi ja laittaa työn uudelleen jonoon. Tällaiset virheet ovat kuitenkin harvinaisia ja johtuvat todennäköisesti ajonaikaisesta ongelmasta tai logiikkavirheistä.

## 6 Yhteenveto

Työn tavoitteena oli automatisoida manuaalisesti tehty työprosessi ja tavoitteisiin päästiin ilman suurempia ongelmia. Työn tuloksena saatiin kehitettyä ohjelmistorobotti, jonka käyttöönotosta seurasi, että aikaisemmin manuaalisesti hoidettu työ tapahtuu nyt automaattisesti. Työprosessin suorittaminen ei vaadi enää ihmisen työvoimaa ja prosessin tehokkuus parani. Ihmiseltä meni yhden asiakkaan käsittelyyn keskimäärin noin 5 minuuttia kun taas robotilta kuluu samaan työhön noin 10 sekuntia.

Työn automatisoinnista seurasi säästöä resursseissa, niin ihmisissä kuin rahassakin. Ihmistyövoima voitiin ohjata vaativimpiin työtehtäviin, joita robotti ei pysty tekemään ja säästetty raha investoitiin uusien robottien kehittämiseen.

Kehittäminen Blue Prismillä on osoittautunut todella suoraviivaiseksi, ainakin kun kohdejärjestelmiä on vain muutama. Eniten aikaa vienyt vaihe oli laskuarkiston mallinnus, koska jokaisen eri painalluksen ja vaiheen tarkka mallinnus vei aikaa. Ongelmia mallinnuksessa ei tosin ollut, koska laskuarkisto toimii selaimessa ja Blue Prism tarjoaa hyvät työkalut HTML-pohjaisen sivun käytön mallintamiseen. Ongelmia olisi varmasti ollut enemmän, jos käytettäviä järjestelmiä olisi ollut useampia tai jos järjestelmä olisi toiminut jollain muulla alustalla.

Käyttökokemuksena prosessin valvominen helpottui ja läpi menneistä töistä sekä virheistä saadaan nopeasti ja helposti vertailukelpoista dataa ulos. Jatkokehitystä robotti ei tällä hetkellä kaipaa, koska se suoriutuu tehtävästä halutulla tavalla. Jos kuitenkin suorituksessa alkaa ilmetä virheitä, vaatii robotti lisätarkistelua ja mahdollisten virheiden korjausta. Mahdollisia lisäominaisuuksia robotilta voidaan pyytää, jolloin ne myös hyväksymisen jälkeen tullaan robottiin tekemään.

Loppuhuomiona voidaan todeta, että virheenkäsittelyn väärentyyppisten laskujen suhteen olisi voinut tehdä paremmin. Nykyisellään robotti kerää tiedot jokaisesta vääranlaisesta laskusta ajonaikaiseen listaan ja lähettää tämän listan eteenpäin vasta työprosessin loppuksi. Tämä tarkoittaa sitä, että jos prosessi kaatuu ajon aikana, niin kaikki listaan kerätyt laskut katoavat muistista.

Tärkeiden tietojen kerääminen ajonaikaiseen listaan ei ole niin sanotun parhaan käytännön mukaista. Koska prosessi suorittaa paljon töitä, on riskinä, että robotti kaatuu ennen kuin lista päästään loppuksi lähettämään. Ratkaisuna tähän mahdolliseen riskiin olisi, että jokaista ajoa varten luodaan väärän tyyppisille laskuille oma jonotyö, jossa on lista. Tähän jonotyön listaan kerättäisiin tieto jokaisesta löydetystä väärän tyyppisestä laskusta. Koska jonotyöt pysyvät prosessin jonon muistissa, ei robotin kaatumisella ole vaikutusta näihin

tietoihin. Näin vältettäisiin tilanne, jossa ajonaikaisen listan pitäisi selvitä ajon loppuun asti, jotta kerätyt tiedot eivät katoaisi.

## Lähteet

Abhimanyu, V. 2022. The Process Analysis Phase in RPA. Viitattu 12.11.2023. Saatavissa <https://www.tutorialspoint.com/the-process-analysis-phase-in-rpa>

Also. 2023. Robotic process automation (RPA): Potential for business. Viitattu 1.11.2023. Saatavissa [https://www.also.com/ec/cms5/en\\_6000/6000/company/blog/article/future-technologies/robotic-process-automation-rpa-potential-for-businesses.jsp](https://www.also.com/ec/cms5/en_6000/6000/company/blog/article/future-technologies/robotic-process-automation-rpa-potential-for-businesses.jsp)

Asiakastieto. Telia Finland Oyj. Viitattu 16.11.2023. Saatavissa <https://www.asiakastieto.fi/yritykset/fi/telia-finland-oyj/14756079/taloustiedot>

Aslani, O. 2020. 7 Biggest Benefits of RPA (Robotic Process Automation). Viitattu 1.11.2023. Saatavissa <https://www.kofax.com/learn/blog/benefits-of-rpa>

Blue Prism Docs. 2023a. Application modeller. Viitattu 12.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-6-7/en-us/frmlIntegrationAssistant.htm>

Blue Prism Docs. 2023b. Application modeller spying. Viitattu 12.11.2023. Saatavissa [https://bpdocs.blueprism.com/capture-3-0/en-us/user-guide/application\\_modeller\\_spying.htm](https://bpdocs.blueprism.com/capture-3-0/en-us/user-guide/application_modeller_spying.htm)

Blue Prism Docs. 2023c. Blue Prism. Viitattu 1.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-7-0/en-us/helpIntroduction.htm?>

Blue Prism Docs. 2023d. Flowcharts. Viitattu 12.11.2023. Saatavissa [https://bpdocs.blueprism.com/bp-7-0/en-us/helpFlowcharts.htm?tocpath=Interface%7CStudio%7CProcess%20Studio%7C\\_\\_\\_\\_\\_1](https://bpdocs.blueprism.com/bp-7-0/en-us/helpFlowcharts.htm?tocpath=Interface%7CStudio%7CProcess%20Studio%7C_____1)

Blue Prism Docs. 2023e. Work Queues. Viitattu 12.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-6-7/en-us/helpQueues.htm?TocPath=Interface%7CStudio%7CProcess%20studio>

Blue Prism Docs. 2023f. Workflow – work queues. Viitattu 12.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-6-10/en-us/sysman-queues.html>

Blue Prism Docs. 2023g. Process Studio. Viitattu 12.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-6-10/en-us/frmProcess.htm>

Blue Prism Docs. 2023h. Page stage. Viitattu 12.11.2023. Saatavissa [https://bpdocs.blueprism.com/bp-7-0/en-us/frmStagePropertiesSubSheet.htm?tocpath=Interface%7CStudio%7CStage%20types%7C\\_\\_\\_\\_\\_18](https://bpdocs.blueprism.com/bp-7-0/en-us/frmStagePropertiesSubSheet.htm?tocpath=Interface%7CStudio%7CStage%20types%7C_____18)

Blue Prism Docs. 2023i. Object Studio. Viitattu 12.11.2023. Saatavissa [https://bpdocs.blueprism.com/bp-7-1/en-us/helpObjectStudio.htm?tocpath=Interface%7CStudio%7CObject%20Studio%7C\\_\\_\\_\\_\\_0](https://bpdocs.blueprism.com/bp-7-1/en-us/helpObjectStudio.htm?tocpath=Interface%7CStudio%7CObject%20Studio%7C_____0)

Blue Prism Docs. 2023j. Processes/objects - exception types. Viitattu 12.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-6-10/en-us/helpExceptions.htm>

Blue Prism Docs. 2023k. Queue management. Viitattu 12.11.2023. Saatavissa <https://bpdocs.blueprism.com/bp-6-7/en-us/control-queues.html>

Blue Prism Portal. 2017. Process Template 1 - Instructions. Viitattu 12.11.2023. Saatavissa <https://portal.blueprism.com/system/files/2017-09/Process%20Template%201%20-%20Basic%20-%20Instructions.pdf>

Cision PR Newswire. 2022. SS&C Completes Acquisition of Blue Prism Group PLC Viitattu 1.11.2023. Saatavissa <https://www.prnewswire.com/news-releases/ssc-completes-acquisition-of-blue-prism-group-plc-301503933.html>

Hopetutors. Blue Prism Interview Questions and Answers. Viitattu 12.11.2023. Saatavissa <https://hopetutors.com/blog/interview-questions/blue-prism-interview-questions-and-answers-for-experienced/>

Katya, L. 2023. Is Low-Code Automation Really That Simple? Viitattu 1.11.2023. Saatavissa <https://www.blueprism.com/resources/blog/low-code-automation/>

Motivitalabs. RPA Development Life Cycle. Viitattu 12.11.2023. Saatavissa <https://motivitalabs.com/robotic-process-automation-development-life-cycle/>

Ron, M. 2019. Blue Prism looks to partners to expand robotic process automation with AI. Viitattu 1.11.2023. Saatavissa <https://techcrunch.com/2019/01/24/blue-prism-looks-to-partners-to-expand-robotic-process-automation-with-ai/>

Sahiti, K. 2023. RPA Life cycle. Viitattu 12.11.2023. Saatavissa <https://www.edureka.co/blog/rpa-lifecycle>

Simplilearn. 2023. What Is Blue Prism: Architecture, Components, Features and More. Viitattu 12.11.2023. Saatavissa <https://www.simplilearn.com/tutorials/rpa-tutorial/what-is-blue-prism>

Software Testing Help. 2023. 10 Most Popular Robotic Process Automation RPA Tools In 2023. Viitattu 1.11.2023. Saatavissa <https://www.softwaretestinghelp.com/robotic-process-automation-tools/>

Tutorialspoint. Blue Prism - Exceptions Handling. Viitattu 12.11.2023. Saatavissa [https://www.tutorialspoint.com/blue\\_prism/blue\\_prism\\_exceptions\\_handling.htm](https://www.tutorialspoint.com/blue_prism/blue_prism_exceptions_handling.htm)

Warwick, A. 2007. Co-operative bank extends automation with Blue Prism Viitattu 1.11.2023. Saatavissa <https://web.archive.org/web/20220707225417/https://www.computerweekly.com/news/2240082707/Co-operative-bank-extends-automation-with-Blue-Prism>