

**LOW-CODE JA NO-CODE TYÖKALUT SARASTIA365 HR
KÄYTTÖÖNOTON APUNA**



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus, Hämeenlinnan korkeakoulukeskus
syksy, 2021

Jami Perälä

TIIVISTELMÄ

Opinnäytetyön tarkoituksena oli tutkia low-code ja no-code -ratkaisujen soveltumista asiakkailta saadun aineiston käsittelyyn, siten että se voidaan automatisoinnin avulla ottaa käyttöön Sarastia365 HR -järjestelmässä. Opinnäytetyössä tutkittiin, mitä mahdollisuuksia low-code ja no-code -ratkaisut tarjoavat lähdejärjestelmien datan muuntamiseksi automaattisesti tai puoliautomaattisesti vastaanottavaan HR-järjestelmään. Opinnäytetyössä arvioitiin soveltuvatko Microsoft Power Platform -työkalut automatisoinnin toteuttamiseen. Opinnäytetyön toimeksiantaja oli Sarastia Oy.

Opinnäytetyön tietopohja koostuu low-code ja no-code -ratkaisujen sekä ohjelmistorobotiikan teoriasta, jossa käsitellään kuinka prosesseja voidaan automatisoida. Low-code ja no-code -ratkaisujen avulla voidaan nopeuttaa monia prosesseja ja näitä eri mahdollisuuksia tutkittiin tässä opinnäytetyössä. Lisäksi teorian taustalla oli lähdejärjestelmien data ja vastaanottavan HR-järjestelmän toiminnallisuudet ja ominaisuudet.

Tutkimuksessa havaittiin, että Sarastia365 HR -järjestelmän käyttöönottojen työvaiheisiin liittyy monia manuaalisia työvaiheita, joiden automatisointiin voitaisiin soveltaa low-code ja no-code -ratkaisuja. Soveltuvuusselvityksen perusteella automatisoitavien työvaiheiden vaatimukset pitää ensin määritellä, ennen kuin käyttöönottoprosessia voidaan automatisoida. Tutkimuksessa huomioitiin Microsoft Power Platform -sovellusten mahdollisuudet ja Sarastian nykyinen prosessi Sarastia365 HR-järjestelmän käyttöönottoille.

Avainsanat Low-code, no-code, ohjelmistorobotiikka, automatisointi

Sivut 30 sivua ja liitteitä 2 sivua

ABSTRACT

The aim of this thesis was to research low-code and no-code solutions and how they apply to the data provided by the customer so that the data can be deployed automatically to the Sarastia365 HR system. In the thesis it was observed what possibilities low-code and no-code solutions provide for the customer's data conversion automatically to the HR system. The thesis also evaluated if the Microsoft Power Platform tools can be applied to achieve the automatization. This thesis was commissioned by Sarastia Oy.

The theoretical part of this thesis is based on the theory of low-code, no-code, and robotic process automation. In the theory it is covered how processes can be automated with the above topics. With low-code and no-code solutions many processes can be automated and in this thesis those processes were observed. Customer's system data and the receiving HR system's functionalities and features serve as theory.

As an output of the thesis, it was noticed that Sarastia365 HR system's deployment processes have many manual steps that can be automated by applying low-code and no-code solutions. Based on the proof-of-concept the steps which should be automated should first be defined before automatizing them. Microsoft Power Platform solutions and possibilities and the current deployment process of Sarastia365 HR system were taken into account in the study.

Keywords Low-code, no-code, Robotic Automation Process, automation

Pages 30 pages and appendices 2 pages

Sanasto

| | |
|---------------------|--|
| API | Application programming interface eli ohjelmointirajapinta, jossa ohjelmat voivat toimia toisten ohjelmien ja laitteistojen välillä. |
| Automaatio | Itsenäisesti toimiva laite tai prosessi |
| Avoin lähdekoodi | Ohjelmiston koodi, joka on kenen tahansa tutustuttavissa ja muokattavissa |
| Käyttöliittymä | Laitteen tai ohjelman osa, jonka kautta laitetta käytetään, esimerkiksi tietokoneen työpöytä |
| Low-/no-code | Low-code ja no-code -ratkaisuilla voidaan automatisoida prosesseja graafisen käyttöliittymän kautta ilman tai vähällä ohjelmointiosaamisella |
| Regex | Tietojenkäsittelyssä lauseke, joka määrittelee säännöllisen joukon merkkijonoja. Käyttäjän antamaa syötettä (lauseke) voidaan muotoilla ja se voidaan tarkastaa. |
| RPA | Robotic Process Automation, ohjelmistorobotiikka. Teknologia, jonka avulla voidaan automatisoida rutiiniprosesseja tietotyössä |
| SaaS-palvelu | Ohjelmiston jakelumalli, jossa palveluntarjoaja ylläpitää omalla palvelimellaan sovellusohjelmistoa ja tarjoaa sitä asiakkaille internetin välityksellä |
| Screen-scraping | Tiedonharavointi näytöltä, kun tiedot ovat esimerkiksi tekstimuodossa. |
| Suljettu lähdekoodi | Ohjelmiston koodi, joka on vain ohjelmiston kehittäjän käytössä |
| SQL-tietokanta | Relaatiotietokanta, joka koostuu tauluista, joiden välille voidaan luoda yhteyksiä. Tietokantoja voidaan muokata käyttäjän toimesta. |
| Tekoäly | Tietokone tai ohjelma, joka pystyy tekemään älykkäitä toimintoja ja kykenee tekemään päätöksiä perustuen algoritmeihin |
| OCR | Tekstintunnistusteknologia, jonka avulla tunnistetaan koneellisesti tekstiä esimerkiksi asiakirjasta. |

Sisälllys

| | | |
|-----|--|----|
| 1 | Johdanto | 1 |
| 2 | TUTKIMUKSEN TARKOITUS JA TAVOITTEET | 2 |
| 2.1 | Tarkoitus..... | 2 |
| 2.2 | Tavoitteet..... | 2 |
| 2.3 | Rajaus | 2 |
| 3 | TOIMEKSIANTAJA | 3 |
| 3.1 | Sarastia Oy | 3 |
| 3.2 | Sarastia365 HR | 4 |
| 4 | OHJELMISTOROBOTIIKKA JA AUTOMAATIO..... | 6 |
| 4.1 | Low-code ja no-code | 6 |
| 4.2 | Ohjelmistorobotiikan historia | 6 |
| 4.3 | Ohjelmistorobotiikka ja automaatio yleisesti | 7 |
| 4.4 | Ohjelmistorobotiikan hyödyntäminen..... | 8 |
| 4.5 | Ohjelmistorobotiikan toimintatavat | 9 |
| 5 | MICROSOFT POWER PLATFORM | 12 |
| 5.1 | Microsoft Power Platform yleisesti..... | 12 |
| 5.2 | Power FX | 13 |
| 5.3 | Power Automate | 15 |
| 6 | LOW-CODE JA NO-CODE KÄYTTÖÖNOTTOJEN TUKENA | 18 |
| 6.1 | Power Automaten hyödyntäminen | 19 |
| 6.2 | Power FX:n hyödyntäminen..... | 21 |
| 7 | JOHTOPÄÄTÖKSET JA POHDINTA | 23 |
| 8 | YHTEENVETO | 25 |
| | LÄHTEET..... | 26 |

Kuvat

| | |
|--|----|
| Kuva 1 Sarastia kartalla..... | 3 |
| Kuva 2 Sarastia365-ratkaisut | 4 |
| Kuva 3 RPA:n peruskomponentit | 10 |
| Kuva 4 Microsoft Power Platform | 12 |
| Kuva 5 Power Appsin näkymä | 14 |
| Kuva 6 Power FX:n ja Excelin funktiot | 14 |
| Kuva 7 Power Automate Desktopin näkymä..... | 17 |
| Kuva 8 Palkka- ja HR-järjestelmän käyttöönottoprojektin prosessi..... | 18 |
| Kuva 9 Aineiston ajamisen projekti Sarastia365 HR -järjestelmään | 19 |
| Kuva 10 Esimerkki for-each -silmukan käytöstä | 20 |

Taulukot

| | |
|-------------------------------------|----|
| Taulukko 1 RPA:n tunnusmerkit | 20 |
|-------------------------------------|----|

Liitteet

| | |
|---------|--|
| Liite 1 | Aineistonhallintasuunnitelma |
| Liite 2 | Esimerkki Sarastian vastaavuustaulukosta |

1 Johdanto

Tässä opinnäytetyössä tutkitaan miten henkilöstö- ja palkkapalveluita tuottavan palvelukeskuksen palkkajärjestelmän käyttöönotoissa muunnetaan asiakasorganisaation erillisjärjestelmästä tuotavaa dataa ja onko käyttöönottoprosessia mahdollista automatisoida tai puoliautomatisoida. Käyttöönotot voivat olla vastaanottavan palkkajärjestelmän uusia ominaisuuksia tai olemassa olevien ominaisuuksien laajentamista. Opinnäytetyön toimeksiantaja on Sarastia Oy, joka esitellään luvussa 3.

Jotta asiakasorganisaation tuottamaa dataa voitaisiin käsitellä toimeksiantajayrityksen vastaanottavassa järjestelmässä, tulee data muuntaa järjestelmään sopivaksi ilman, että datan tietosisältö muuttuu. Tavoitteena on tutkia low-code ja no-code -ratkaisujen soveltumista datan muuntamiseen. Low-code ja no-code -ratkaisuilla voidaan automatisoida rutiininomaisia prosesseja, jotka tässä opinnäytetyössä ovat aineiston muuntoprosesseja graafisen käyttöliittymän kautta ohjelmoinnin sijaan. Low-code ja no-code -ratkaisuja on saatavilla monia erilaisia, joista tässä opinnäytetyössä tarkastellaan Microsoft Power Platform -ratkaisuja.

Toimeksiantajayrityksessä käyttöönotoissa on keskusteltu Microsoft Power Platformin low-code ja no-code -ratkaisujen hyödyntämistä, mutta ratkaisuja ei ole vielä toteutettu. Tärkeimpiä kysymyksiä opinnäytetyössä ovat seuraavat:

- Millaisessa muodossa asiakasorganisaation lähdejärjestelmän data on ja miten konvertoidaan vastaanottavaan järjestelmään?
- Millainen datan konvertointiprosessi nykyisin on?
- Miten low-code/no-code -ratkaisuja voidaan käyttää käyttöönottoprosessien automatisoinnissa ja mitä niiden toteuttamisessa tulee huomioida?

2 TUTKIMUKSEN TARKOITUS JA TAVOITTEET

2.1 Tarkoitus

Opinnäytetyössä tarkastellaan Microsoft Power Platform -työkaluja, joiden avulla lähdejärjestelmien dataa voidaan muokata vastaanottajaorganisaation HR-järjestelmään sopivaksi. Opinnäytetyössä arvioidaan ovatko Microsoft Power Platform -työkalut riittäviä käyttöönottoprosessin automatisoinnin toteuttamiseksi. Ideana on tutkia, voidaanko lähdejärjestelmien data muuntaa automaattisesti tai puoliautomaattisesti low-code ja no-code -ratkaisuja käyttäen vastaanottavaan HR-järjestelmään sopivaksi ja mitä ratkaisujen osalta tulee ottaa huomioon. Opinnäytetyö on proof-of-concept (PoC, konseptiselvitys).

2.2 Tavoitteet

Opinnäytetyön käytännön osan tavoite on tuottaa kuvaus siitä, mitä tulee huomioida low-code ja no-code -ratkaisujen käytössä Sarastian HR-järjestelmän käyttöönottojen yhteydessä. Opinnäytetyössä ei luoda konkreettista ratkaisua prosessin automatisoimiseksi, vaan annetaan valmiudet lähteä kehittämään aineiston muuntoprosessia tuotetun konseptiselvityksen pohjalta.

2.3 Rajaus

Opinnäytetyössä tarkastellaan Sarastia Oy:n Sarastia365 HR-järjestelmää, joka toimii vastaanottavana järjestelmänä ja asiakasorganisaation lähdejärjestelmän tuottamaa dataa (esimerkiksi palkkajärjestelmä). Opinnäytetyössä tutkitaan, kuinka Microsoft Power Platform -alustan Power FX ja Power Automate toimivat ja ovatko ne Sarastian kannalta vartenotettavia vaihtoehtoja.

Opinnäytetyössä rajataan pois itse konkreettinen ratkaisu prosessin automatisoinnille ja keskitytään enemmän datan muuntamisen automatisoinnin vaatimukseen aiheen laajuuden vuoksi. Tämän opinnäytetyön tuottaman tiedon perusteella ideaa voidaan työstää eteenpäin toimeksiantajan toimesta.

3 TOIMEKSIANTAJA

3.1 Sarastia Oy

Opinnäytetyö tehdään toimeksiantona Sarastia Oy:lle, joka toimii Suomessa 18 maakunnassa (kuva 1) ja työllistää lähes 900 asiantuntijaa Suomessa. Sarastian tytäryhtiöitä ovat Sarastia Rekry Oy, Suomen Kuntaperintä Oy ja Onvire Oy. Sarastian liikevaihto on 107 miljoonaa euroa ja sillä on 270 omistajaa. (Sarastia, 2021c) Sarastia syntyi, kun kunta-alan KuntaPro Oy ja Taitoa Oy fuusioituvat toukokuussa 2019, muodostaen Suomen suurimman alan toimijan (Sarastia, 2021a).

Kuva 1. Sarastia kartalla (Sarastia, 2021b)

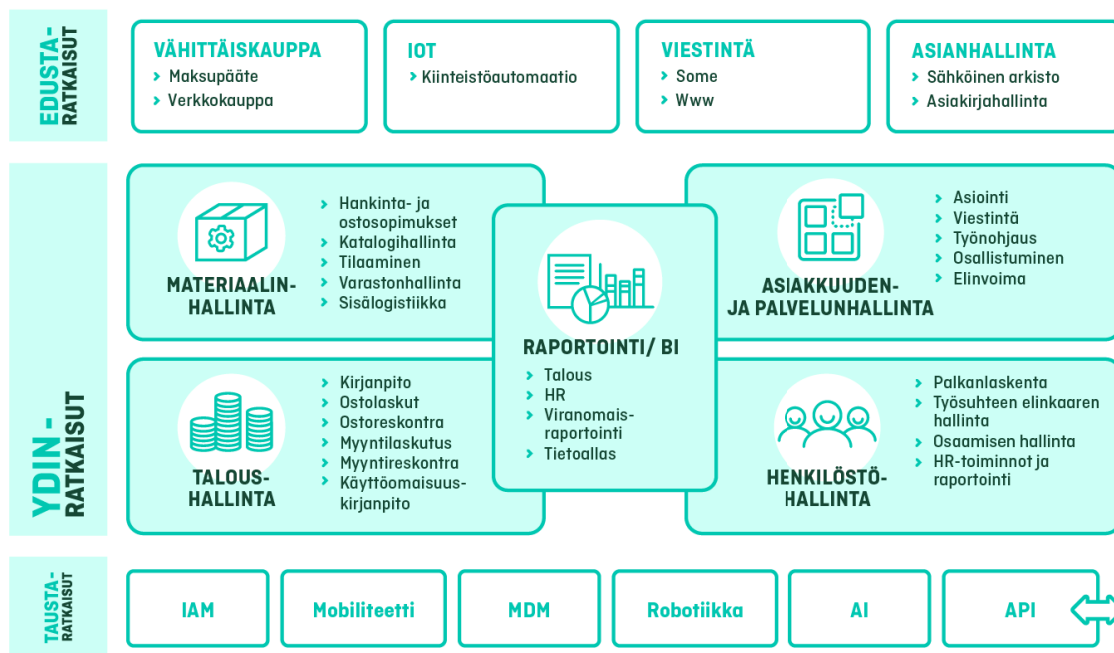


Sarastian palveluihin kuuluvat talous- ja henkilöstöpalvelut, sijais- ja rekrytointipalvelut, hankintapalvelut ja perintäpalvelut. Sarastialla on asiakkaiden prosessien digitalisoimeksi luotu palveluita, jotka kuuluvat Sarastia365 tuoteperheeseen. (Sarastia, 2021e)

3.2 Sarastia365 HR

Sarastia365-ratkaisun palvelut ovat toteutettu SaaS-palveluina (Software as a Service). Ratkaisut koostuvat ydinratkaisusta, joita ovat materiaalinhallinta, taloushallinta, henkilöstöhallinta, asiakkuuden- ja palvelunhallinta ja raportointi (kuva 2). Näiden ydinratkaisujen ympärille on rakennettu edusta- ja taustaratkaisut (Client- ja web-käyttöliittymät). (Sarastia, 2021e)

Kuva 2. Sarastia365-ratkaisut (Sarastia, 2021d)



Sarastia365 HR on osa Sarastian Sarastia365-ydinratkaisuja, joka mahdollistaa helppokäyttöisen henkilöstöhallinnon käyttäjälleen ja joka on mahdollista räätälöidä asiakkaan vaatimusten mukaan. Sarastia365 HR mahdollistaa mobiilikäytön ja sisältää työsuhteen elinkaaren toiminnot, palkanlaskennan ja raportoinnin. (Sarastia, 2021d)

Sarastia365 HR on Sarastian ja Accountor HR Solutionsin kehittämä henkilöstö- ja palkkahallinnon järjestelmä, joka palvelee kuntasektorin vaatimuksia. Sarastia365 HR -järjestelmä on Accountorin MepcoPro -ratkaisun päälle rakennettu järjestelmä. (”Accountor, 2020)

Sarastia365 HR rakentuu moduuleista, joita voidaan käyttää työntekijöiden, esimiesten, palkanlaskijoiden ja järjestelmän pääkäyttäjien toimesta. Järjestelmä koostuu Web Portaalista, jossa

esimiehet ja työntekijät voivat hallita omia tietojaan ja Client-käyttöliittymästä, joka sisältää palkanlaskennan toiminnallisuudet ja raportointia varten tarvittavat työkalut. Sarastia365 HR -järjestelmän palkkamoottoriin on sisälletetty valmiudet soveltaa kunta-alan työehtosopimuksia palkanmaksuun ja määritykset ovat muutettavissa paikallista sopimista vastaaviksi. (Sarastia, 2021d)

Järjestelmään on mahdollista liittää henkilön tiedot, joita voidaan määrittää myös asiakaskohtaisesti. Client-käyttöliittymä ja Web Portaali hyödyntävät samaa tietokantaa, jolloin henkilöstöhallinnon ja palkanlaskennan tietoja voidaan käsitellä reaaliaikaisesti käyttöliittymästä riippumatta. Järjestelmän parametrintien avulla järjestelmän pääkäyttäjät voivat toteuttaa sähköisiä lomakkeita, liittymiä, hakuja ja raportteja, joita asiakkaat voivat käyttää henkilö- ja palkkatietojen ilmoittamiseen. (Sarastia, 2021d)

Opinnäytetyön kannalta on tärkeää huomioida Sarastia365 HR -järjestelmän tiedostojen sisäänluku, kun asiakkailta saatua dataa lähdetään viemään Sarastia365 HR -järjestelmään. Järjestelmästä löytyy toiminto tiedostopohjaisten liittymien tekemiseen, joiden avulla voidaan siirtää järjestelmään tai järjestelmästä palkkatapahtumia, henkilö- ja työsuhdetietoja, loma- ja poissaolojaksoja, parametriluetteloita ja muita henkilöstöhallinnon tietoja. Järjestelmään luettaviin tietoihin voidaan tehdä koodimuuntoja, kun tiedot eriävät lähdejärjestelmien kanssa (asiakkaan aikaisempi järjestelmä). Näihin konversioihin käytetään vastaavuustaulukoita, joissa lähdejärjestelmien tiedot saadaan vastaamaan muodoltaan Sarastia365 HR -järjestelmän vaatimuksia, jotta tiedot voidaan lukea järjestelmään sisään. (Sarastia, 2021d)

4 OHJELMISTOROBOTIIKKA JA AUTOMAATIO

4.1 Low-code ja no-code

Low-code ja no-code työkalut hyödyntävät graafista käyttöliittymää, jonka avulla voidaan kehittää ja luoda sovelluksia. Low-code ja no-code työkalujen käyttöön ei vaadita perinteistä ohjelmointiosaamista, jossa ohjelmakoodia kirjoitetaan rivi riviltä, vaan usein työkalut toimivat ”drag-and-drop” periaatteella. Näin ollen ohjelmien testaaminen ja luominen käy nopeasti ja vaivatta. Low-code ja no-code työkaluilla luodut sovellukset voidaan yhdistää rajapintojen (Application Programming Interface, API) avulla toimimaan yhdessä kolmannen osapuolien ohjelmien kanssa. Ohjelmat rakennetaan eri komponenteista, jotka voidaan tuoda esimerkiksi kulkukaavion avulla, jossa ohjelman eri vaiheet luodaan omina komponentteinaan ja yhdessä ne muodostavat ohjelman. (Pratt, n.d.)

Low-code työkaluissa käyttäjältä voidaan vaatia hieman ohjelmointitietämystä, mutta ammattilaisen tasoa ei vaadita (Pratt, n.d.). Low-code työkalut voivat esimerkiksi käyttää omaa ”matalan ohjelmointitason kieltään”, kuten kappaleessa 5 käsiteltävä Power FX, joka on Microsoftin luoma low-code -ohjelmointikieli. No-code työkalut puolestaan eivät vaadi käyttäjältä mitään ohjelmointiosaamista. No-coden käyttämisessä käyttäjän tulee kuitenkin ymmärtää liiketoiminnan kannalta tarpeet ja säännöt, joita sovelletaan. Esimerkkinä no-codesta on tässä kappaleessa käsitelty ohjelmistorobottiikka, jossa ohjelmistorobotin luomiseen ei tarvita välttämättä lainkaan perinteistä ohjelmointiosaamista tai -tietämystä.

4.2 Ohjelmistorobottiikan historia

Ohjelmistorobottiikkaa (Robotic Process Automation, RPA) pidetään mullistavana teknologiana, mutta sen voidaan katsoa myös olevan vain laajennus jo aikaisemmin kehitetyille teknologioille, jotka juontavat juurensa 1990-luvulle. RPA on kehittynyt nopeasti, mutta sen käyttämiä ominaisuuksia on ollut jo aikaisemmin olemassa. Näitä ovat mm. testiautomaatiojärjestelmät, screen-scrapers-työkalut, työkulkujen automaatio- ja hallinnointityökalut ja tekoäly. RPA terminä voidaan kuitenkin ajoittaa 2000-luvun alkuun. (Ostdick, 2016)

4.3 Ohjelmistorobotiikka ja automaatio yleisesti

Robotteja voidaan käyttää suorittamaan erilaisia toimintoja esimerkiksi kodinkoneissa, kulkuneuvoissa, juoma-automaateissa ja jokapäiväisissä tehtävissä. Termi automaatio tulee Kreikan kielen sanoista *autos*, joka tarkoittaa ”itsestään” ja *motos*, joka tarkoittaa ”liikkuvaa”. Automaation voidaan näin ollen sanoa olevan teknologia, joka hoitaa tuotteiden ja palveluiden tuotannon lähes ilman ihmisen avustusta. (Tripathi, 2018, s. 6)

Ohjelmistojen kanssa automaatio voi tarkoittaa esimerkiksi kahden eri järjestelmän integroimista ohjelmistokehityksen avulla. Prosessin automatisoinniksi voidaan kutsua toimintoa, jossa kahden tai useamman järjestelmän välille ohjelmoidaan toimintoja yhdistävä porras. Näin eri järjestelmät voivat käyttää ja hyödyntää toistensa dataa. (Tripathi, 2018, s. 6) Verrattuna perinteiseen automaatioon ohjelmistorobotiikka on huomattavasti käyttäjäystävällisempi. Perinteisessä automaatiossa ohjelmointi on pääosassa eri järjestelmien integraatioissa. Järjestelmiä yhdistävät ratkaisut ovat API:ja. Jotkut järjestelmät saattavat asettaa rajoituksia integraatioiden toteutuksille, koska järjestelmien muokkaaminen voi olla rajattua suljetun lähdekoodin vuoksi. Automaatio perinteisen ohjelmoinnin kanssa tuo mukanaan myös kustannuskysymyksen, kun prosessien automatisointia aletaan suunnittelemaan. Toteutukseen voidaan joutua palkkaamaan ulkopuolista työvoimaa tai hankkimaan konsultointia. (Bhatt, 2018)

RPA on etukäteen määriteltyä logiikkaa, sääntöjä ja dataa hyödyntävä teknologia, jonka avulla voidaan automatisoida prosesseja. Ohjelmistorobotiikassa ”robotit” suorittavat ihmisen tekemiä työvaiheita, joita robotille annetaan. Näitä työvaiheita voivat olla tietyt esimerkiksi hiiren tai näppäimistön painallukset käyttöliittymässä, datan tunnistaminen ja poimiminen ja ymmärtäminen tietokoneen käyttöliittymän tapahtumista. Ohjelmistorobotti pystyy suorittamaan rutiininomaiset ja toistuvat työvaiheet nopeammin ja tarkemmin kuin ihminen. (UiPath, n.d.)

RPA:ssa robotti ymmärtää käyttäjän syöttämiä toimintoja käyttöliittymätasolla ja toteuttaa niitä kuten käyttäjä itsekin toteuttaisi. Näin ollen käyttäjältä ei vaadita ohjelmointiosaamista ja automatisoinnin toteuttamiseen ei vaadita API:en osaamista tai tietämystä. RPA:lla organisaatiot voivat toteuttaa nopeita, fiksua ja kustannustehokkaita työkaluja tuottavuuden kasvattamiseksi, kun prosessien automatisointi voidaan hoitaa organisaatiossa sisäisesti ilman ulkopuolista konsultaatiota. (Bhatt, 2018)

RPA:lla voidaan automatisoida työtehtäviä, jotka ovat työvaiheiltaan toistuvia, vaativat paljon henkilöresursseja ja kuuluvat eri riskiluokkiin. Näiden työtehtävien ja -vaiheiden tulee olla sääntöihin perustuvia, tarkoin määriteltyjä ja loogisia, jotta ne voidaan automatisoida RPA:lla. Esimerkiksi haluttu syöte tulee olla ohjelmiston purettavissa ja ymmärrettävissä ja vastaanottavan järjestelmän saatavissa. RPA:ssa robotti jäljittelee ihmisen toimintoja käyttämällä käyttöliittymän ohjelmistoja ja toteuttamalla sääntöihin perustuvia toimintoja, kuten esimerkiksi näppäimistöllä kirjoittaminen. Robotille voidaan antaa tehtäväksi täyttää tietty tekstikenttä halutulla tekstillä, jonka robotti suorittaa annettujen ohjeiden mukaisesti. (Tripathi, 2018, s. 7-10) Ohjelmistorobotin käyttöönotto voidaan toteuttaa hyvin lyhyessä ajassa, mutta se vaatii organisaatiolta prosessien optimointia ja organisaation roolien ja tehtävien sopimista käyttöönoton eri vaiheissa. Automatisoidut prosessit vaativat valvontaa, koska manuaalinen työ ei siirry robotille ilman toteutusta ja testaamista. (Kovalainen, 2021a) Taulukossa 1 on kuvattu RPA:n ja ohjelmistorobotin käyttöönoton prosessin tunnusmerkit.

Taulukko 1. RPA:n tunnusmerkit. (Anagnoste, 2017, s. 2)

| |
|--|
| 1. Käyttäjien kouluttama |
| 2. Toimii käyttäjän käyttöliittymän kanssa |
| 3. Käsittelee jäseneltyjä, toistettavia ja tietokoneisiin perustuvia työvaiheita |
| 4. Toimii saumattomasti eri järjestelmien välillä |
| 5. Toimii eri tiedostomuotojen ja -formaattien kanssa (PDF, MS Excel jne.) |
| 6. Suorittaa tarkastuksia ja ottaa huomioon validoinnit aikaisemmin määriteltyjen sääntöjen mukaan |
| 7. Havaitsee helposti poikkeamat (esimerkiksi tietokantapoikkeamat, poikkeamat koodille syötetyissä ehdoissa) |
| 8. Toimii 24/7 ja myös lomien ja viikonloppujen aikana |
| 9. Lokit säilötään ohjelmassa, mutta voidaan asettaa lähetettäväksi sähköpostitse tietyinä aikana, päivänä tai toistuvasti |
| 10. Tarjoaa analysoinnille tarpeet |

4.4 Ohjelmistorobotiikan hyödyntäminen

RPA:sta on tullut osa organisaatioiden arkipäivää. Gartnerin teettämässä tutkimuksessa suomalaisista organisaatioista 81 % hyödynsi tai aikoi hyödyntää ohjelmistorobotiikkaa vuonna 2019. RPA:ta hyödyntämällä voidaan automatisoida henkilöiden tyypillisiä työtehtäviä, joiden automatisoinnilla voidaan käyttää resursseja parantamaan asiakastytyväisyyttä, asiantuntijuutta ja

työnlaatua, kun virheiden määrä vähenee ja työaika voidaan käyttää tehokkaammin. (Pursiainen, 2020) RPA:ta voidaan hyödyntää asiakaspalvelussa, myynti- ja ostoprosesseissa, palkkahallinnossa, tietovarastojen käsittelyssä, henkilöstöhallinnossa, rekrytointiprosesseissa ja datan käsittelyssä eri formaateissa. (Automationedge, 2021)

HR-tietojen käsittely on aikaa vievä prosessi. RPA:n avulla voidaan kerätä, organisoida ja käsitellä HR-tietoja, kuten työntekijän historia-, palkka- ja koulutustietoja. Tämä data voi olla eri muodoissa, kuten muokattavissa oleva tiedosto tai käsin kirjoitettu tai skannattu asiakirja. (Automationedge, 2021) Muokattavissa oleva tiedosto tai data voidaan käsitellä yleensä sellaisenaan RPA:n avulla, mutta skannattua tai käsin kirjoitettua asiakirjaa ei. Tätä varten kuitenkin RPA hyödyntää optista merkkien tunnistusta (optic character recognition, OCR), jonka avulla voidaan ”lukea” tekstiä kuvamuodosta muokattavaksi dataksi. OCR:llä on kuitenkin omat rajoituksensa, sillä se ei ole kognitiivinen ja asiakirjoista eristetty data tulee tarkastaa mahdollisilta virheiltä. (Tzeng, 2020)

RPA käyttää myös tekoälyä, jonka avulla esimerkiksi aiemmin mainittu OCR toimii. Tekoäly perustuu tekoälyalgoritmeihin, jotka puolestaan perustuvat perinteiseen ohjelmointiin, matematiikkaan ja tilastotieteisiin. Algoritmi käsittelee sille syötetyn datan ja vastaukset, joita tekoäly vertaa sääntöihin ja päättelee tilastollisesti vastauksen. (Kananen, 2019, s. 24) RPA:n rinnalle on nostettu myös termi hyperautomaatio, jonka ideana on kehittää automaatiota prosessien alusta aina niiden loppuun (end-to-end). Hyperautomaatiossa RPA on osana kokonaista automaatioprosessia. (Pursiainen, 2020)

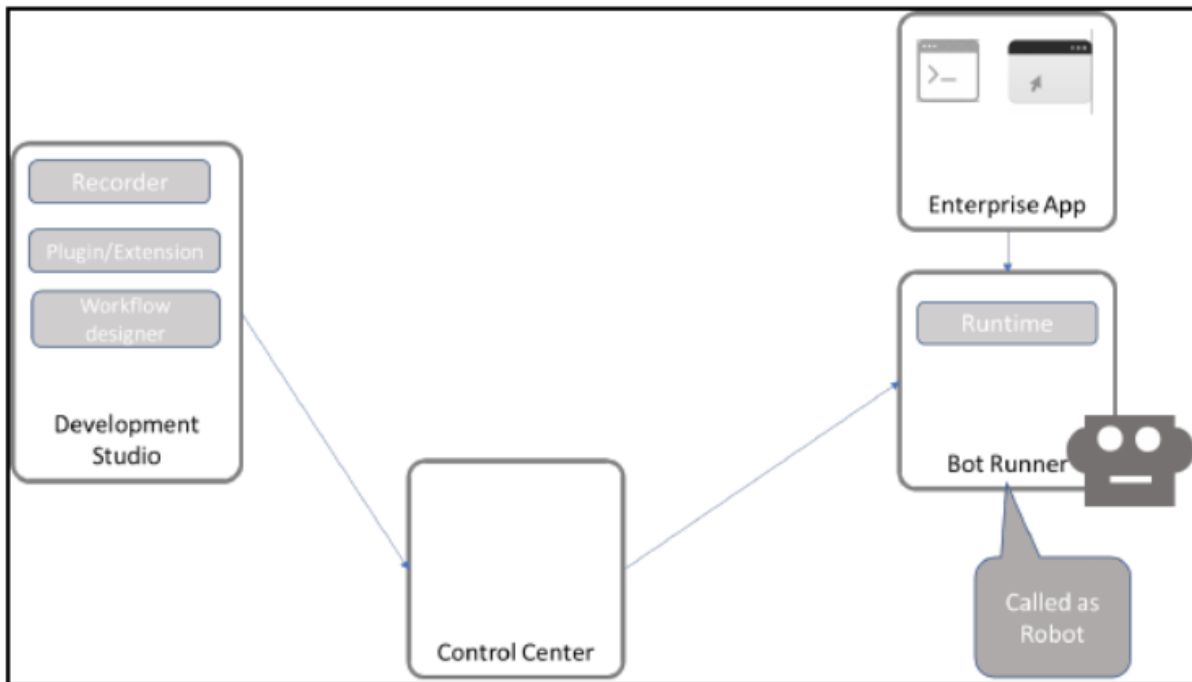
4.5 Ohjelmistorobotiikan toimintatavat

RPA:lla voidaan tuoda paljon lisäarvoa organisaation toimintaan. Robottien käyttöönotolla voidaan suorittaa työtehtävät samalla tarkkuudella ja tehokkuudella toistuvasti. Robotit toimivat laajalti olemassa olevien järjestelmien kanssa. Tällaisia järjestelmiä eri ERP-järjestelmät, SAP, Oracle ja Microsoftin työkalut sekä eri ohjelmat kuten .NET, Java ja komentorivi. (Tripathi, 2018, s. 11)

RPA:n toteuttamiseen on monia eri alustoja, kuten UiPath, Automation Anywhere, BluePrism, TruBot, AutomationEdge, Microsoft Power Automate, Appian RPA, WorkFusion Intelligent Automation Cloud, Kofax RPA ja monia muita. Kaikkien toimintaperiaate on kuitenkin sama ”jos,

sitten, muutoin”, jonka avulla robotit toimivat. (Gartner, 2021) Jokainen alusta tarjoaa peruskomponentit robottien rakentamiseen, joita ovat esimerkiksi recorder (nauhoittaja), development studio (tuotantoympäristö), plugin/extension (lisäosat), bot runner (robotti), control center (ohjauskeskus). (Tripathi, 2018, s. 15) Näiden toimintaa ja suhteita on kuvattu kuvassa 3.

Kuva 3. RPA:n peruskomponentit. (Tripathi, 2018, s. 15)



Development studion avulla konfiguroidaan robotit toimimaan halutulla tavalla. Yleisesti development studio -alustat sisältävät kulkukaavion tyylisen suunnittelun, jolla robotille voidaan määrittellä eri työvaiheet. Vaikka RPA ei välttämättä tarvitse ohjelmointitaitoja, on niistä hyötyä, sillä development studion avulla voidaan hyödyntää ohjelmoinnissa käytettävää logiikkaa ja toimintoja kuten esimerkiksi if-else -rakenteita (jos-muutoin), silmukka- ja toistorakenteita, muuttujia ja säännöllisiä lausekkeita (regular expression, regex tai regexp). Monet RPA-alustat tarjoavat kolmannen osapuolen luomia lisäosia, joilla voidaan laajentaa robotin toimivuutta. (Tripathi, 2018, s. 15–16)

Recorderin tehtävänä on konfiguroida robotteja toimimaan nauhoittamalla robotille graafiselta käyttöliittymältä ne työvaiheet, joita halutaan sen suorittavan. Recorder nauhoittaa hiiren ja näppäimistön syötöt ja liikkeit, joiden perusteella se toimii käyttöliittymällä toistamalla työvaiheet.

(Tripathi, 2018, s. 15) Esimerkkinä nauhoituksesta voi olla selaimen avaaminen ja navigointi halutulle verkkosivulle ja kaikki siihen kuuluvat työvaiheet.

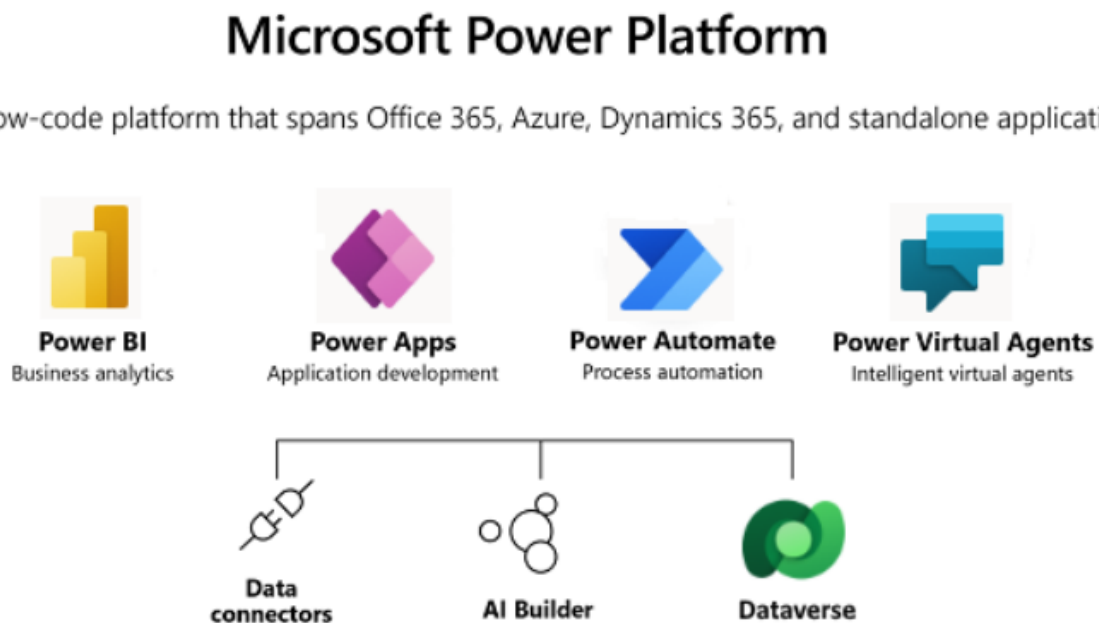
Control centerin tehtävänä on kattaa robottien ylläpitoon ja tarkkailuun vaadittavat operaatiot. Sen avulla voidaan käynnistää tai sulkea robotti, aikatauluttaa robotin toiminta, ylläpitää ja julkaista koodia ja ottaa robotteja käyttöön eri tehtäviä varten. (Tripathi, 2018, s. 16)

5 MICROSOFT POWER PLATFORM

5.1 Microsoft Power Platform yleisesti

Microsoft Power Platform on kehitysalusta, joka sisältää työkalut liiketoiminnan tukemiseksi. Power Platform koostuu neljästä eri työkalusta: Power Apps, Power Automate, Power BI, Power Virtual Agents (kuva 4). Power Platformin avulla voidaan automatisoida liiketoiminnan prosesseja ja raportointia ja se voidaan integroida Office 365- ja Dynamics 365-ratkaisuihin. (Digia, 2021)

Kuva 4. Microsoft Power Platform. (Microsoft, 2020c)



Power Platformin itsenäiset työkalut yhdistyvät Microsoftin Dataverseen, jonka avulla voidaan tallentaa ja hallita Power Platformin yksittäisten sovellusten dataa. Tiedot tallennetaan taulukkojoukkoihin, jotka koostuvat riveistä ja sarakkeista, joita voidaan muokata tietotyyppikohtaisesti ja organisaatiotasolla halutuksi. (Microsoft, 2020b)

Opinnäytetyön kannalta tarkastellaan Power FX- ja Power Automate-ratkaisuja, joiden avulla voidaan luoda low-code- ja no-code -ratkaisuja Sarastia365 HR-järjestelmän käyttöönottojen tueksi. Power FX- (Power Apps) ja Power Automate -ratkaisuja on toimeksiantajayrityksessä mietitty käyttöönottoprosessin tueksi.

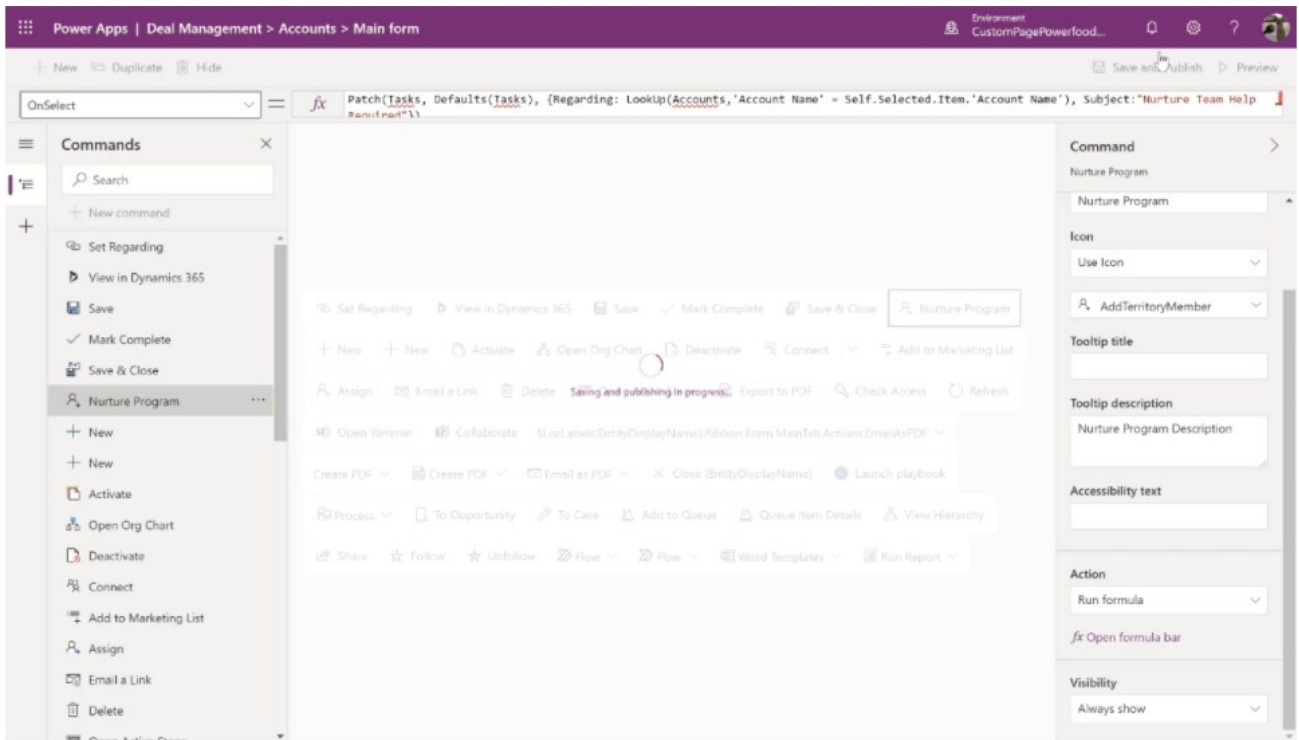
5.2 Power FX

Microsoft esitteli maaliskuussa 2021 uuden avoimen lähdekoodin low-code-ohjelmointikielen nimeltä Power FX. Power FX on tyyliltään hyvin saman kaltainen kuin Microsoft Excelin kaavakieli. Power FX käyttäytyy kuten Excelin kaavat, jossa kirjoittaakseen koodia käyttäjän ei tarvitse tietää, kuinka ohjelma toimii, vaan ohjelma laskee koodin tai datan muuttuessa tarvittavat muutokset ohjelmassa. (Melanson, 2021) Power FX on sama kieli, joka on jo ollut käytössä Microsoft Power Apps -ohjelmassa. Vuonna 2021 Microsoft lanseerasi Power FX:n omaksi avoimen lähdekoodin kieleksi. Power FX mahdollistaa ohjelmien tuottamisen no-code -tasolta aina pro-code -tasolle, joka puolestaan mahdollista erilaisten tiimien yhteistyön projekteissa. (Lindhorst, 2021b)

Power Apps koostuu sovelluksista, palveluista, yhdistimistä ja tietoaalustasta, jolla voidaan rakentaa erilaisia sovelluksia liiketoiminnan tueksi. Power Appsilla voidaan muodostaa yhteys joko taustalla olevaan Microsoft Dataverse -tietoympäristöön tai paikallisesti tai pilvessä toimiviin tietolähteisiin tallennettuun dataan. Power Appsilla luodut sovellukset tukevat liiketoimintaa, kun liiketoimintaprosesseista voidaan tehdä digitaalisia ja prosesseja voidaan automatisoida. Sovelluksia luodaan low-coden avulla, jolloin sovellusten luomiseen ei tarvita perinteisen ohjelmoinnin osaamista. Power Appsissa low-code -ohjelmointikieli on Power FX. (Microsoft, 2021c)

Power Apps -sovelluksia voidaan luoda kolmesta erilaisesta pohjasta: kaaviosovellukset, mallipohjaiset sovellukset ja portaalit. Kaaviosovellukset luodaan tyhjästä kaaviosta, johon voidaan yhdistää tietolähde (esim. Excel-tiedosto tai SQL-tietokanta), jonka pohjalta voidaan luoda sovelluksia eri tarkoituksiin. Mallipohjaiset sovellukset käyttävät tietomallissa (Microsoft Dataverse) olevaa dataa, joista voidaan tehdä lomakkeita, näkymiä ja muita komponentteja mallista luodulla käyttöliittymällä, joka skaalautuu eri laitteille (esim. web, älypuhelimet ja tabletit). Microsoft Dataversen avulla prosessit voidaan ottaa käyttöön eri mallipohjia hyödyntävissä sovelluksissa tehokkaasti. Portaaleissa luodaan sivustoja ja sovelluksia organisaation ulkopuolisille käyttäjille, jotka voivat luoda, muokata ja tarkastella Dataversessä olevia tietoja. (Microsoft, 2021d) Power Appsin näkymä on esitelty kuvassa 5.

Kuva 5. Power Appsin näkymä. (Lindhorst, 2021a)



Power FX toimii kaavamaisesti, kuten Microsoftin Excel. Excel-taulukon on mahdollista rakentaa taulukkolaskennalla kaava, joka suorittaa haluttua tehtävää (funktioita). Excelin käyttäjät voivat helposti ottaa käyttöönsä Power FX -kielen ja käyttää Power Apps -ohjelmaa työnsä automatisointiin, koska Power FX:n kieli perustuu Excelissä käytettäviin funktioihin. Kuvassa 6 on listattu Power FX:n ja Excelin funktiot. Vihreällä merkityt funktiot ovat identtisiä tai lähes identtisiä Excelin funktioiden kanssa. Excelissä olevia funktioita tullaan tulevaisuudessa lisäämään enemmän Power FX:ään. (Lindhorst, 2021b)

Kuva 6. Power FX:n ja Excelin funktiot. (Lindhorst, 2021b)

| | | | | | | | |
|------------|----------------|-------------|-----------|----------|---------------|---------------|----------------|
| Abs | Color | DateValue | GUID | Left | Param | Sequence | Time |
| Acos | ColorFade | Day | HashTags | Len | Pi | ShowColumns | TimeValue |
| Acot | ColorValue | Defaults | Hour | Ln | PlainText | Shuffle | TimeZoneOffset |
| AddColumns | Concat | Degrees | If | LookUp | Power | Sin | Today |
| And | Concatenate | Distinct | IfError | Lower | Proper | Sort | Trim |
| Asin | Count | DropColumns | IsBlank | Match | Radians | SortByColumns | TrimEnds |
| AsType | Cos | EncodeUrl | IsEmpty | MatchAll | Rand | Split | Ungroup |
| Atan | Cot | EndsWith | IsError | Max | RenameColumns | Sqrt | Upper |
| Atan2 | CountA | Errors | IsMatch | Mid | Replace | StartsWith | User |
| Average | CountIf | Exp | IsNumeric | Min | RGBA | StdevP | Validate |
| Blank | CountRows | Filter | IsToday | Minute | Right | Substitute | Value |
| Calendar | DataSourceInfo | Find | IsType | Mod | Round | Sum | VarP |
| Char | Date | First | JSON | Month | RoundDown | Switch | Weekday |
| Choices | DateAdd | FirstN | Language | Not | RoundUp | Table | With |
| Clock | DateDiff | ForAll | Last | Now | Search | Tan | Year |
| Coalesce | DateTimeValue | GroupBy | LastN | Or | Second | Text | |

Power FX:n koodia voidaan käsitellä Excelin kaavojen tyyppisinä kaavariveinä Power Apps -sovelluksessa tai Visual Studio Coden teksti-ikkunaa käyttämällä, joten se soveltuu niin ohjelmoinnin ammattilaisille kuin vähemmän ohjelmointia osaaville. Power FX:llä luodaan kaikkien ohjelmointikielten omaisesti lausekkeita, joilla voidaan luoda laskentaa. Laskennan tulos voidaan sijoittaa muuttujaan, jota voidaan käyttää argumenttina tai sijoittaa muualle. Power FX:ssä lauseke ei kuitenkaan varsinaisesti kerro mitä lasketaan, vaan käyttäjän tehtävänä on sijoittaa laskettava kohde muuttujaan tai välittää se funktioon. Power FX:ssä kirjoitetaan kaava, joka sitoo lausekkeen tunnisteeseen. Esimerkiksi kaavassa

```
voima = massa * kiihtyvyys
```

tunniste "voima" muuttuu aina, kun kaavan muuttujia "massa" tai "kiihtyvyys" päivitetään. (Microsoft, 2021b)

Power FX:n kieli on deklarativinen (eli paradigma esittää asiat niiden välisillä suhteilla), jossa tekijä kuvaa mitä haluaa ohjelman logiikan tekevän. Jotta käyttäjät pystyvät käyttämään logiikkaa, heidän tulee tietää käytössä olevia käsitteitä, jotka usein ovat saman tyyliä kuin Excelin käsitteet. Deklaratiivinen logiikka määrittelee kohteelle laskettavat tai määritettävät ominaisuudet, esimerkiksi tekstille: onko teksti kursivoitua tai lihavoitua, minkä värinen teksti on ja mikä on tekstin koko. Deklaratiivisen logiikan lisäksi Power FX käyttää imperatiivista logiikkaa, joka tarkoittaa, että kielelle voidaan komentaa tehtäväksi eri asioita, kuten datasettien päivitystä. (Microsoft, 2021b)

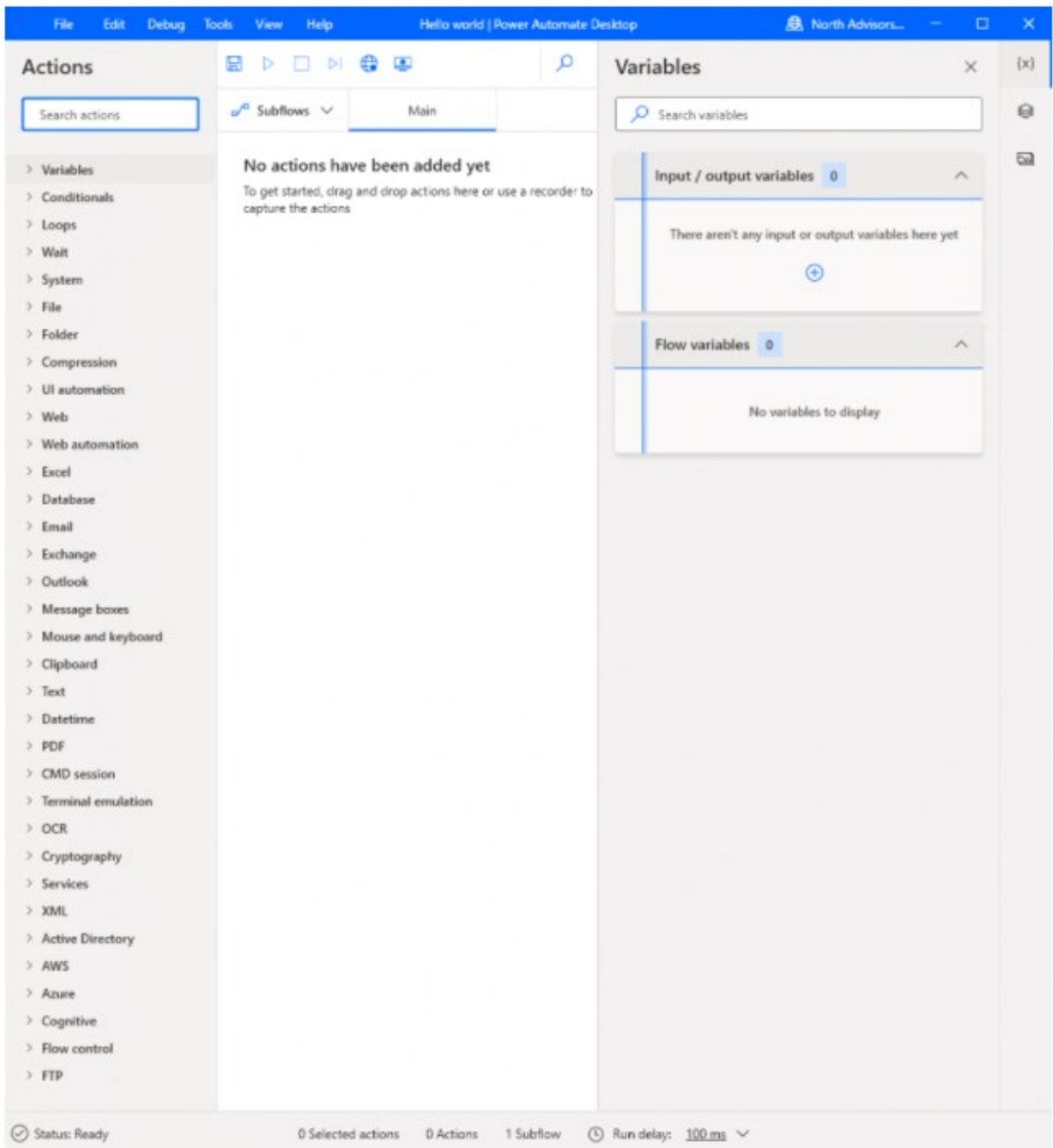
5.3 Power Automate

Power Automate on Microsoftin palvelu prosessien automatisointiin RPA:n avulla, joka toimii "drag-and-drop" -periaattella eikä vaadi perinteistä ohjelmointiosaamista (no-code). Power Automate toimii pilvessä, mutta siitä on saatavilla myös Power Automate Desktop. Power Automaten avulla voidaan automatisoida liiketoimintaprosesseja, luoda ajastettuja toimintoja (kuten muistutukset, automaattiset viestit yms.), siirtää dataa järjestelmien välillä ajastetusti ja automatisoida paikallisesti prosesseja esimerkiksi Excelissä. Näitä automatisointeja kutsutaan nimellä "flow". Power Automaten pilvisovelluksessa voidaan luoda yksinkertaisia automatisointeja. Power Automatessa kuka tahansa voi luoda prosessin ja jakaa sen yhteisön kanssa kaikkien käytettäväksi.

Power Automatessa voidaan hyödyntää API:ja, joka mahdollistaa automatisointien luomisen monelle eri ohjelmalle tai alustalle. Luotuja valmiita automatisointimalleja on yli 380 kappaletta. (Microsoft, 2021a)

Power Automate Desktop (PAD) laajentaa Power Automaten mahdollisuuksia automatisoitavista prosesseista. PAD toimii nimensä mukaisesti graafisen käyttöliittymällä eli työpöydällä (kuva 7). PAD:lla voidaan luoda automatisointeja riippumatta tiedostomuodoista, ohjelmista tai järjestelmistä. (Microsoft, 2020a) PAD toimii paikallisesti omana ohjelmanaan, jossa voidaan automatisoida prosesseja. Samoin kuin Power Automatessa, voidaan PAD:ssa valita listasta haluttu toiminto drag-and-drop -periaatteella. Näitä toimintoja ovat esimerkiksi jonkin järjestelmän tai ohjelman suorittaminen, kuvakaappauksen ottaminen tai jopa koodin suorittaminen (JavaScript, Python yms.). Lisäksi toimintoja löytyy myös eri tiedostoformaattien käsittelylle, kuten Excel, Outlook ja PDF. Luotaessa uutta flowta, annetaan tapahtumien eri työvaiheille muuttujia, joista PAD pitää kirjaa. Automatisointeja voidaan luoda hakemalla suoraan listasta toiminto, mutta myös nauhoittamalla hiiren ja näppäimistön syötteitä. (Roine, 2021)

Kuva 7. Power Automate Desktopin näkymä. (Roine, 2021)



6 LOW-CODE JA NO-CODE KÄYTTÖNOTTOJEN TUKENA

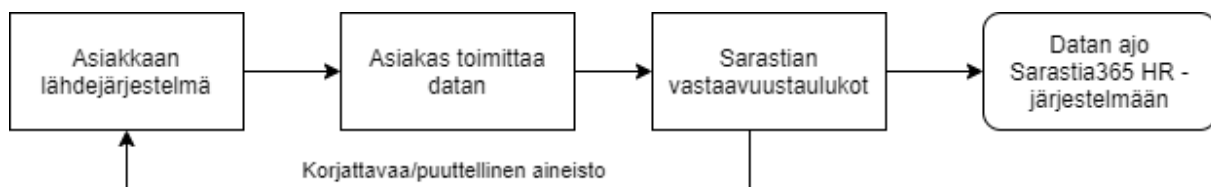
Opinnäytetyön tarkoituksena on tutkia, voidaanko Power FX:ää (low-code) ja Power Automatea (no-code) hyödyntää Sarastia365 HR -järjestelmän käyttöönoton tukena. Nykyisellään käyttöönottoprosessin datan konvertointivaihe sitoo paljon henkilöresursseja, kun työtä tehdään manuaalisesti täyttämällä asiakkaan toimittamat aineistot lähdejärjestelmistä vastaavuustaulukoihin. Mitä suurempi asiakas on, sen suurempi manuaalisen työn osuus on. Käyttöönottoprosessi on seitsemänvaiheinen aina projektisuunnitelmasta tuotantokäyttöönottoon. Opinnäytetyössä keskitytään neljänteen vaiheeseen ”Datan konvertointi”. Koko käyttöönottoprosessi on kuvattu kuvassa 8.

Kuva 8. Palkka- ja HR-järjestelmän käyttöönottoprojektin prosessi. (Sarastia, 2021e)



Käyttöönnotossa asiakas toimittaa lähdejärjestelmästäan käyttöönnottoon vaadittavat tiedot, esimerkiksi työntekijöiden perustietoja, palkkatietoja ja työhistoriatietoja. Nämä tiedot käsitellään Sarastian tekemien vastaavuustaulukoiden (konversiotaulukot) avulla Excelissä csv-muotoon, jonka jälkeen ne voidaan ajaa Sarastia365 HR -järjestelmään. Prosessi kuvattu kuvassa 9. Tapauskohtaisesti myös asiakas voi täyttää vastaavuustaulukkopohjiin lähdejärjestelmistä saadut tiedot ja joissain tapauksissa on hyödynnetty ohjelmistorobotiikkaa. Hyödynnetty ohjelmistorobotiikka on toteutettu Python-ohjelmoinnilla ja skripteillä, joten opinnäytetyön tekohetkellä Sarastia ei käytä Sarastia365 HR-järjestelmän käyttöönotoissa low-code- tai no-code -ratkaisuja.

Kuva 9. Aineiston ajamisen projekti Sarastia365 HR -järjestelmään.



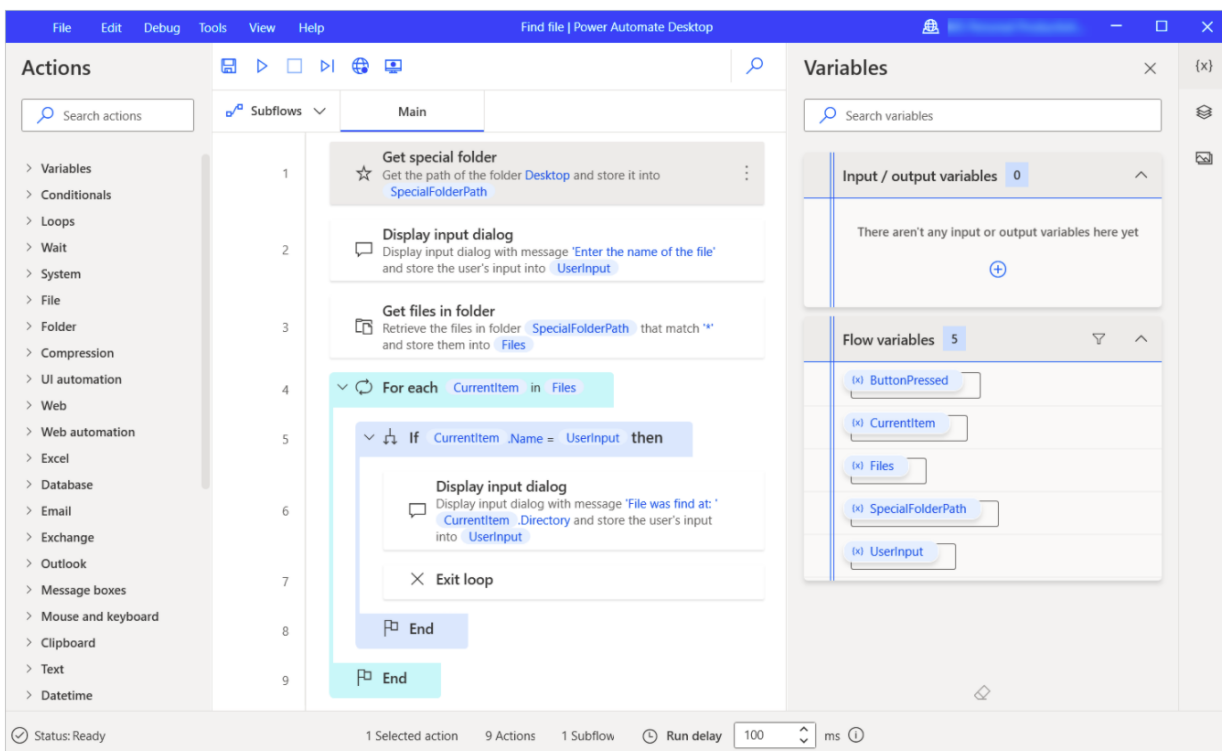
6.1 Power Automaten hyödyntäminen

RPA:n mahdollistamaa automatisointia voidaan käyttää työvaiheisiin, jotka ovat toistettavissa ja perustuvat sääntöihin. Automatisoinnin kannalta tulee huomioida, millaisessa muodossa asiakkaan toimittama materiaali on. Aineistot toimitetaan taulukkomuotoisina tiedostoina (xlsx- tai csv-muodossa), mutta eri järjestelmät tuottavat taulukoita eri asetteluilla. Jotta aineistoa voitaisiin käsitellä RPA:n avulla, tulisi lähdejärjestelmän tuottama aineisto olla asettelultaan vakioimuotoista. Näin aineistoa ei tarvitse muuttaa manuaalisesti, vaan voidaan hyödyntää RPA:ta. Muuten jokaiselle erilaiselle asettelulle vaadittaisiin oma robotinsa, koska työvaiheet eivät ole samanlaisia robotin näkökulmasta. Esimerkiksi jos työntekijän nimi on asiakkaan A aineistossa solussa A1 ja asiakkaan B aineistossa solussa B4, ei voida samaa robottia käyttää, koska robotti etsisi molemmista aineistoista solua, joka sen toimintaan on määritelty. Robotin toiminta tulee olla ”jäsenneltyä, toistettavaa ja tietokoneisiin perustuvaa”, kuten taulukossa 1 on kuvattu.

PAD tarjoaa mahdollisuuden käsitellä tiedostoja tavalla, jossa voidaan jättää käsin tehtävä tiedonsiirto robotille. Oletetaan, että kaikilta asiakkailta saatu on tietyt aineistot (esim. henkilö-

palkka- tai työhistoriatiedot) olisivat ennalta määrättyssä vakiomuodossa, jossa RPA:n avulla aineisto voitaisiin käsitellä yhdellä robotilla per aineisto. Kun lähdejärjestelmästä saatava aineisto on csv-muodossa, voitaisiin käyttää esimerkiksi PAD:n ominaisuuksia käsitellä Excel-tiedostoja. Robotille kerrotaan mitä tiedostoja sen tulee käsitellä ja missä sijainneissa. PAD:lla voidaan käyttää Exceliä varten luotuja valmiita työvaiheita, joiden avulla voidaan esimerkiksi avata uusi tai jo olemassa oleva Excel-tiedosto, käsitellä tiedoston sisältöä ja siirtää sisältö toiseen Excel-tiedostoon. Tämä prosessi voidaan toistaa usealle eri tiedostolle luomalla for-each -silmukka (kullekin -silmukka), joka voisi toimia vastaavuustaulukoiden täyttöprosessin automatisoinnissa (kuva 10). For each -silmukassa iteroidaan työvaiheelle annettu luettelo läpi ja haluttu poimittava tieto tai tietojoukko tallennetaan muuttujaan, jota voidaan käyttää myöhemmin. Käytännössä jokaiselle tarvittavaa dataa sisältävälle sarakkeelle ja datan vastaanottavalle taulukon sarakkeelle tulee tehdä oma silmukkansa. Näiden silmukoiden muuttujien kanssa voidaan toteuttaa datan siirtäminen. (Microsoft, 2020d)

Kuva 10. Esimerkki for-each -silmukan käytöstä. (Microsoft, 2020d)



PAD tarjoaa myös mahdollisuuden automatisoida muiden RPA-työkalujen tavoin prosesseja hiiren ja näppäimistön syötteiden perustella. Tässä käytetään screen-scraping -työkalua, joka nauhoittaa näytöllä tehdyt klikkaukset ja näppäimistön syötöt. Koska RPA:ssa ei tarvitse välittää järjestelmien

rajapinnoista, voidaan käyttöönottoaineistojen sisäänajo Sarastia365 HR-järjestelmään teoriassa tehdä määrittelemällä robotille samat työvaiheet, jotka ihminen tekisi. Screen scraping -työkalun käyttämisessä tulee kuitenkin huomioida Sarastia365 HR -järjestelmään kirjautuminen, joka tapahtuu jokaisen työntekijän henkilökohtaisilla tunnuksilla. Toisaalta tämä ei ole ongelma, jos robotti suoritetaan järjestelmään sisään kirjautuneen henkilön toimesta. Robotin toiminnan kannalta on otettava huomioon myös eri asiakkaat, joilla Sarastia365 HR-järjestelmän Client-käyttöliittymässä saattaa olla eroavaisuuksia käyttöliittymän näkymässä ja valikoissa. Mikäli järjestelmä haluttaisiin käynnistää kokonaan robotilla, tulisi screen scraping -työkalua käyttää myös kirjautumiseen, esimerkiksi kysymällä käyttäjän kirjautumistiedot robotilla ja tallentamalla ne väliaikaisesti muuttujiin. Tämä tapa voi kuitenkin olla tietoturvan kannalta huono toteutus.

6.2 Power FX:n hyödyntäminen

Power FX:n käyttäminen käyttöönottoprosessin automatisoinnissa on myös mahdollista. Sarastialla on käytössä Power Apps, jonka avulla voidaan käsitellä dataa ja automatisoida sen käsittelyä luomalla siihen soveltuvia sovelluksia. Power Apps käyttää Power FX:ää sovellusten toteuttamisessa. Nykyisellään Sarastiassa on Power Appsilla toteutettu sovelluksia Sarastia365 Talous -ratkaisun tueksi. Sovellukset on Microsoft Dynamics 365 Dataversen päälle rakennettuja käyttöliittymiä ja sovelluksia on suunniteltu kehitettävän lisää.

Yksinkertaisena esimerkkinä voidaan mainita kuvitteellinen tilanne, jossa luodaan esimerkiksi Power Appsilla käyttöönottoprosessin työvaiheiden automatisointia varten Microsoft Dataverseä hyödyntäviä sovelluksia, joilla voidaan taulukkojoukkojen avulla käsitellä siihen tuotua tai vietyä dataa. Tämän taulukkojoukoissa olevan datan käsittelyyn voidaan luoda omat sovellukset ja portaalit, joiden avulla myös asiakkaatkin voivat käsitellä dataa.

Power Appsiin on mahdollista integroida kolmannen osapuolen sovelluksia yhdistimien (connector) avulla, joka toimii siltana datalähteen ja käyttöliittymän välillä, toisin sanoen API:na. Power Appsissa yhdistimiä löytyy valmiiksi luotuina ja myös kolmannen osapuolen luomina. Yhdistimien käyttäminen sovellusten välillä suoraviivaistaa Power Apps -sovellusten toimintaa, mutta API:n luominen vaatii kuitenkin usein perinteistä ohjelmointia, joka puolestaan vaatii resursseja suunnitteluun ja toteutukseen. API:n luominen low-code ohjelmoinnilla, tässä tapauksessa Power FX:n avulla, säästäisi näitä resursseja. Olettaen, että Sarastialla olisi tarjota API:t eri järjestelmiin,

voitaisiin niihin luoda Power Appsin ja muiden Power Platform -työkalujen avulla dataa käsitteleviä sovelluksia hyödyntäen Power FX:ää datan käsittelyssä. Esimerkiksi Power Apps toimii myös yhdessä Power Automaten kanssa ja niiden yhdistämiseen on satoja flow-vaihtoehtoja. Power FX:n integroiminen Power Automateen (tai PAD:iin) laajentaa automatisoinnille kehitettäviä vaihtoehtoja, mutta opinnäytetyön tekohetkellä Power FX:ää ei ole mahdollista käyttää Power Automatessa.

7 JOHTOPÄÄTÖKSET JA POHDINTA

Prosessien automatisoinnissa ei ole tarkoitus kattaa kaikkea kerralla, vaan lähinnä keskittyä ratkaisemaan prosessin eri työvaiheet pilkkomalla ne pienempiin kokonaisuuksiin. Kun prosessin perusasiat on kuvattu ja on tarkasteltu mitä voidaan ja kannattaa automatisoida, voidaan näiden perustusten päälle lähteä testaamaan koko prosessin automatisointia. (Kovalainen, 2021b)

Low-code ja no-code -ratkaisujen yleistymisen on tuonut perinteisten ohjelmoinnin osaajia ja niin sanottuja ”kansalaisohjelmoijia” lähemmäs toisiaan. Nämä kaksi osapuolta voivat työskennellä yhteisissä projekteissa, jossa molemmat voivat osallistua esimerkiksi juuri prosessien automatisoinnin toteuttamiseksi. HR-asiantuntijalla voi olla tiedot esimerkiksi siitä, mitä tietoja tulee käsitellä ja sisällyttää käyttöönottoprosessin kannalta järjestelmään, kun taas ohjelmoija ymmärtää miten ohjelmointia voidaan hyödyntää käyttöönotoissa. Low-code ja no-code -ratkaisujen myötä myös HR-asiantuntija voi automatisoida eri työvaiheita, jolloin ohjelmoijien resurssit voidaan käyttää vaativimpiin tehtäviin.

Sarastia 365 HR -järjestelmän käyttöönottojen kannalta voidaan todeta myös, että prosessissa on paljon manuaalista työtä vaativia vaiheita, joita voitaisiin automatisoida low-code ja no-code -ratkaisuilla. Esimerkiksi RPA:n käytölle on selkeitä sääntöihin perustuvia, toistettavissa olevia ja tiettyä logiikkaa noudattavia työtehtäviä. Low-coden näkökulmasta voidaan jo käytössä olevalla Power Appsilla luoda datan käsittelylle ratkaisuja, jotka ainakin voisivat tehostaa työvaiheita. Käyttöönottoprosessin työvaiheita kannattaa automatisoida, koska nykyisellään se sitoo resursseja, jotka voitaisiin käyttää hyödyksi muualla. Automatisoinnin kannalta kuitenkin käyttöönottoprosessissa tulee arvioida:

- mitä prosessin työvaiheita kannattaa ja voidaan automatisoida?
- mitä resursseja työvaiheiden automatisoinnit vaativat?
- käsiteltävän datan yhtenäisyyden huomioiminen käsittelyn kannalta.
- kuinka työvaiheet automatisoidaan, mitä ratkaisuja käytetään?

Power Platform -työkalujen yhteensopivuus ja integraatio keskenään on Sarastian kannalta kannattava vaihtoehto, verrattaessa muiden palveluntarjoajien low-code ja no-code -ratkaisuihin, jotka eivät välttämättä ”keskustele” suoraan toistensa kanssa. Power FX:n on tarkoitus tulevaisuudessa olla Power Platform -sovellusten (Power BI, Power Virtual Agents, Microsoft Dataverse, Power Automate) yhteinen kieli, jonka avulla voidaan saavuttaa kattavia integraatioita sovellusten välillä. Power FX:n ja Power Automaten välillä ei opinnäytetyön tekohetkellä ole vielä olemassa integraatiota. Power FX tullaan integroimaan Power Platformin työkaluihin tulevaisuudessa, joten näin ollen voidaan olettaa, että Power FX:n käyttäminen Power Automaten kanssa tulee laajentamaan automatisoitavien prosessien mahdollisuuksia. Power FX tullaan integroimaan ja ottamaan käyttöön Power Platform -sovelluksille seuraavan muutaman vuoden aikana (Cunningham, 2021).

8 YHTEENVETO

Sarastia 365HR -järjestelmän käyttöönottojen automatisointi on laaja aihe, jossa on paljon automatisoinnille mahdollisia työvaiheita. Low-code ja no-code työkalujen käyttö käyttöönottoprosessin automatisoimiseksi tai puoliautomatisoimiseksi on Sarastian näkökulmasta kannattavaa. Nykyisellään Sarastia365 HR -järjestelmän käyttöönotto asiakkaalle käyttää paljon henkilöstöresursseja eri tiimeistä: HR-asiantuntijoita ja -arkkitehteja, ohjelmoijia ja robotiikan henkilöstöä. Low-code ja no-code työkalujen yleistymisen puolesta puhuu puolestaan työtehtävien automatisoinnista ja niiden kautta saavutetusta liiketoimintaedusta, kun resursseja voidaan käyttää tehokkaammin. Nykyisellään prosessi on aikaa vievä, vaikka ohjelmistorobotiikkaa onkin jo käytössä. Kun low-code ja no-code -ratkaisut yleistyvät, voidaan niiden avulla saavuttaa suurempi automatisoinnin aste ja työtehtävät voidaan toteuttaa entistä tehokkaammin, jolloin organisaation resursseja voidaan käyttää tehokkaammin liiketoimintaa tukevaan työhön.

Power Platform -sovellukset ovat alustana hyvä pohja lähteä automatisoimaan Sarastia 365HR -järjestelmän käyttöönottoprosesseja ja Power Platform- sovelluksille suunnitellut kehitykset laajentavat sen käyttömahdollisuuksia. Power Appsin ja Power FX:n hyödyntämisen puolesta puhuu myös Sarastia365 Talous -ratkaisun tueksi luodut Power Apps -sovellukset. Power FX:n ollessa avoimen lähdekoodin ohjelmointikieli ja ottaen huomioon jo nyt monet eri Power Platform -yhteisön luomat ratkaisut eri järjestelmien yhdistämiseen, tulee Power Platform -sovellusten käyttö laajentamaan mahdollisuuksia liiketoiminnan tehostamiselle.

LÄHTEET

Accountor. (11.12.2020) *Helsinki ja Espoo valitsivat HR- ja palkkahallintoonsa uuden ratkaisun, joka pohjautuu Accountorin MepcoPro-järjestelmään.* (11.12.2020). Haettu 17.6.2021 osoitteesta <https://www.accountor.com/fi/finland/uusi/helsinki-ja-espoo-uusi-hr-ja-palkka-ratkaisu-pohjautuu-mepco-pro>

Anagnoste, S. (2017) *Robotic Automation Process – The next major revolution in terms of back office operations improvement. Proceedings of the International Conference on Business Excellence.* Haettu 1.7.2021 osoitteesta <https://doi.org/10.1515/picbe-2017-0072>

Automationedge. (10.3.2021). *10 Best Use Cases to Automate using RPA in 2021. RPA Blogs.* Haettu 13.7.2021 osoitteesta <https://automationedge.com/blogs/10-best-use-cases-to-automate-using-rpa-in-2021/>

Bhatt, S. (21.9.2018) *The Big Fight: RPA vs. Traditional Automation. BoTree Technologies Blog.* Haettu 30.6.2021 osoitteesta <https://www.botreetechnologies.com/blog/the-big-fight-robotic-process-automation-vs-traditional-automation/>

Cunningham, R. (2.3.2021). *Introducing Microsoft Power FX: the low-code programming language for everyone.* Microsoft Power Apps. <https://powerapps.microsoft.com/en-us/blog/introducing-microsoft-power-fx-the-low-code-programming-language-for-everyone/>

Digia. (2021). *Microsoft Power Platform – Helppokäyttöinen kehitysalusta liiketoimintasi tarpeisiin.*
Digia. Haettu 17.7.2021 osoitteesta <https://digia.com/palvelumme/liiketoimintajärjestelmat/microsoft-power-platform/>

Gartner. (2021). *Robotic Process Automation Software Reviews and Ratings.* Gartner. Haettu 13.7.2021 osoitteesta <https://www.gartner.com/reviews/market/robotic-process-automation-software>

Kananen, H. (2019). *Tekoäly: Bisneksen uudet työkalut.* Alma Talent Oy.

Kovalainen, T. (2021a). Automatisointi muuttaa työntekoa. *CGI Suomen blogi*. Haettu 30.6.2021 osoitteesta <https://www.cgi.com/fi/fi/blogi/automatisointi-muuttaa-tyontekoa>

Kovalainen, T. (2021b). Robotiikan PoC:ia ei ole olemassa. *CGI Suomen blogi*. Haettu 27.7.2021 osoitteesta <https://www.cgi.com/fi/fi/blogi/robotiikan-pocia-ei-ole>

Lindhorst, G. (2021a) *Power FX coming to Model-driven Power Apps, Dataverse, and more*. Microsoft Power Apps. <https://powerapps.microsoft.com/en-us/blog/power-fx-coming-to-model-driven-power-apps-dataverse-and-more/>

Lindhorst, G. (2021b). *What is Microsoft Power FX?*. Microsoft Power Apps. <https://powerapps.microsoft.com/en-us/blog/what-is-microsoft-power-fx/>

Melanson, M. (6.3.2021). This Week in Programming: Microsoft's Power FX 'Low Code' Language. *The New Stack*. <https://thenewstack.io/this-week-in-programming-microsofts-power-fx-low-code-language/>

Microsoft. (2021a). *Get Started with Power Automate*. Microsoft Docs. Haettu 22.7.2021 osoitteesta <https://docs.microsoft.com/en-us/power-automate/getting-started>

Microsoft. (2020a). *Introduction to Power Automate Desktop*. Microsoft Docs. Haettu 22.7.2021 osoitteesta <https://docs.microsoft.com/en-us/power-automate/desktop-flows/introduction>

Microsoft. (2020b). *Mikä on Microsoft Dataverse?*. Microsoft Docs. Haettu 20.7.2021 osoitteesta <https://docs.microsoft.com/fi-fi/powerapps/maker/data-platform/data-platform-intro>

Microsoft. (2021b). *Microsoft Power FXin yleiskatsaus*. Microsoft Docs. Haettu 20.7.2021 osoitteesta <https://docs.microsoft.com/fi-fi/power-platform/power-fx/overview>

Microsoft. (2021c). *Power Appsin kuvaus*. Microsoft Docs. Haettu 20.7.2021 osoitteesta <https://docs.microsoft.com/fi-fi/powerapps/powerapps-overview>

Microsoft. (2020c). *Power Apps -yrityksen käyttöönoton hallinta*. Microsoft Docs. Haettu 20.7.2021 osoitteesta <https://docs.microsoft.com/fi-fi/power-platform/admin/admin-powerapps-enterprise-deployment>

Microsoft. (2020d). *Use loops*. Microsoft Docs. Haettu 20.7.2021 osoitteesta <https://docs.microsoft.com/en-us/power-automate/desktop-flows/use-loops>

Microsoft. (2021d). *Yleiskuvaus sovellusten luomisesta Power Appsissa*. Microsoft Docs. Haettu 20.7.2021 osoitteesta <https://docs.microsoft.com/fi-fi/powerapps/maker/>

Ostdick, N. (26.6.2021). The Evolution of Robotic Process Automation (RPA): Past, Present, and Future. *RPA*. Haettu 30.6.2021 osoitteesta <https://www.uipath.com/blog/rpa/the-evolution-of-rpa-past-present-and-future>

Pratt, M. (n.d.). *Low-code and no-code development platforms*. TechTarget. <https://searchsoftwarequality.techtarget.com/definition/low-code-no-code-development-platform>

Pursiainen, T. (16.9.2021). Hyperautomaation lyhyt historia. *CGI Suomen blogi*. Haettu 10.7.2021 osoitteesta: <https://www.cgi.com/fi/fi/blogi/hyperautomaation-lyhyt-historia>

Roine, J. (28.4.2021) An in-depth look at Microsoft Power Automate Desktop and building Robotic Process Automations. *Jussi Roine*. <https://jussiroine.com/2021/04/an-in-depth-look-at-microsoft-power-automate-desktop-and-building-robotic-process-automations/>

Sarastia. (2021a) Meidän tarina. Haettu 16.6.2021 osoitteesta <https://www.sarastia.fi/sarastia-yrityksena/meidan-tarina/>

Sarastia. (2021b) Meille Töihin. Haettu 16.6.2021 osoitteesta <https://www.sarastia.fi/rekrytointi/>

Sarastia. (2021c) Sarastian Identiteetti. Haettu 16.6.2021 osoitteesta <https://www.sarastia.fi/sarastia-yrityksena/identiteetti/>

Sarastia. (2021d) Sarastia365 HR Ratkaisukuvaus.

Sarastia. (2021e) Sarastia365-ratkaisut. Haettu 17.6.2021 osoitteesta <https://www.sarastia365.fi/sarastia365-ratkaisut/>

Tripathi, A. (2018). *Learning Robotic Automation*. Packt Publishing.

Tzeng, T. (29.9.2020). Combining OCR With AI and RPA for Advanced Data Analysis. *Artificial Intelligence*. Haettu 13.7.2021 osoitteesta <https://www.uipath.com/blog/ai/unstructured-data-analysis-with-ai-rpa-and-ocr>

UiPath. (n.d.). *Robotic Process Automation (RPA)*. UiPath. Haettu 30.6.2021 osoitteesta <https://www.uipath.com/rpa/robotic-process-automation>

Liite 1: Aineistonhallintasuunnitelma

Tässä opinnäytetyössä käsitellyn kolmannen osapuolen materiaalin omistaa Sarastia Oy. Sarastia Oy:n materiaalia ovat Sarastia365 HR -järjestelmän järjestelmäkuvaus ja Sarastia Oy:n edustajan luvalla on kuvattu järjestelmää niiltä osin, kun sitä voidaan opinnäytetyössä käsitellä. Opinnäytetyötä varten on saatu Sarastia Oy:ltä Sarastia365 HR -järjestelmän ratkaisukuvaus ja käyttöönotoissa käytettäviä konversiotaulukoita, joiden avulla käyttöönottoprosessin tietoja käsitellään lähdejärjestelmästä vastaanottavaan järjestelmään. Käsitellyt konversiotaulukot sisältävät tekaistua dataa ja opinnäytetyössä on käytetty vain dummy-dataa sisältävän esimerkkikuvan konversiotaulukosta. Sarastia Oy:ltä saatu aineisto säilytetään tekijän kannettavan tietokoneen C-asemalla työn tekoprosessin ajan. Aineistoa ei säilytetä opinnäytetyön valmistumisen jälkeen, koska aineisto on luottamuksellista ja aineistot tuhotaan noudattaen toimeksiantajan ohjeita. Aineisto voidaan pyytää toimeksiantajalta, mikäli opinnäytetyön tulokset täytyy todentaa.

Liite 2: Esimerkki Sarastian vastaavuustaulukosta

| | | | | |
|----|---|---|-------------------------------|-----------------------------------|
| 5 | Ruksi niihin, jotka tuodaan konversioliittymällä, muutoin tallennetaan manuaalisesti. | X | X | |
| 6 | Muut kommentit | | | |
| 7 | | Henkilö- ja ts. excelin B-sarakkeen tieto | saman rivin V sarakkeen tieto | poiminta RNG Toimi - välilehdeltä |
| 8 | | | | |
| 9 | | 1001060971-9999 | 03.01.2020 | XXXXXXXXXX |
| 10 | | 1002070971-9989 | 15.02.2020 | XXXXXXXXXX |
| 11 | | 1003070971-9999 | 01.01.2020 | XXXXXXXXXX |
| 12 | | 1004080971-9989 | 03.03.2020 | XXXXXXXXXX |
| 13 | | 1005080971-9999 | 09.01.2020 | XXXXXXXXXX |
| 14 | | 1006090971-9989 | 01.02.2020 | XXXXXXXXXX |