



Excel-automaatio IT-työssä

Ammattikorkeakoulututkinnon opinnäytetyö

Tieto- ja viestintäteknikka

Kevät, 2025

Timi Lapiolahti

Koulutus Tieto- ja viestintätekniikka
Tekijä Timi Lapiolahti
Työn nimi Excel-automaatio IT-työssä
Ohjaaja Teemu Järvenpää

Vuosi 2025

Tämä opinnäytetyö käsitteli Excel-ympäristön yksitoikkoisten tehtävien automatisointia. Työssä tarkasteltiin ensin automaatiota käsitteenä, sen historiaa ja kuinka vaikuttava asia se on IT-alalla. Esitettiin työelämän projektista tulleita haasteita, jotka koostuivat suuren Excel-aineistomäärän muokkaamisesta. Nämä ratkaistiin Excel-makroilla sekä Visual Basic for Applications (VBA)-ohjelmointikielellä. Nämä ovat sisäänrakennettuja työkaluja Excelissä. Työssä käytiin läpi makrojen ja VBA-ohjelmoinnin perusteita ja miten automatiikka oli toteutettu näillä työkaluilla.

Työn varsinainen tutkimuskohde kohdistui vaihtoehtoihin automatiikkatyökaluihin, joilla mahdollisesti pystytään toteuttamaan samat ongelmat eri menetelmällä. Työssä tutkittiin kolmea automatiikka- ja robotiikkatyökalua; Power Automate, UiPath ja Blue Prism. Jokainen näistä työkaluista käytiin läpi aloittelijan näkökulmasta. Tutkittiin näiden ominaisuuksia ja helppokäyttöisyyttä, sekä kaikilla luotiin demoprosessi vertailun vuoksi. Verrattiin työkaluja keskenään ja niistä valittiin yksi, jolla lähdettiin toteuttamaan suurempaa prosessia, joka korvaisi makrot ja VBA-ratkaisut. Blue Prism osoittautui aloittelijalle hankalimmaksi työkaluksi, joten valinta oli Power Automaten ja UiPathin välillä.

Power Automate -työpöytärobotiikkatyökalu osoittautui parhaaksi vaihtoehdoksi, sillä kyseessä oli Microsoftin tuote ja sen ansiosta Excelin kanssa oli saumaton integraatio. Verrattuna VBA-ohjelmointiin, saatiin pienemmällä vaivalla ja osaamisella toteutettua vastaava automaatioprosessi. Suorituskyvyltään se pärjasi suurien aineistomäärien kanssa. Automatiikkaa saatiin myös kehitettyä pidemmälle lokitoiminnallisuudella. Osoittautui, että VBA-ohjelmoinnin ja robotiikan yhteistoiminen voisi olla ihanteellisin vaihtoehto toteuttaa Excel-ympäristön automatiikkaa. Tällä tavoin lievitettäisiin VBA-ohjelmoinnin hankaluutta robotiikalla, ja Excel-makroilla vähennetään robotin suorittamia vaiheita.

Avainsanat Automaatio, Excel, robotiikka, tietokoneohjelmat, Visual Basic for Applications
Sivut 34 sivua

DP Information and Communication Technology
Author Timi Lapiolahti
Subject Excel automation in IT operations
Supervisor Teemu Järvenpää

Year 2025

This thesis explores the automation of redundant tasks in Excel environment. The thesis first examines automation as a concept, its history and importance in the IT industry. Challenges from a work-related project are presented, which consisted of modifying large amounts of Excel spreadsheets. The challenges were solved by using Excel macros and Visual Basic for Applications programming language (VBA). These are built-in tools in Excel. The basics of macros and VBA are introduced, and how automation was implemented using these tools.

The focus of this thesis considers alternative solutions to the previous challenges using third-party automation tools. Three automation tools are examined; Power Automate, UiPath and Blue Prism. Each one of these tools is introduced from a beginner's perspective. The features and the ease of use of these tools were examined, also demo-processes were created individually for comparison. The tools were compared to each other, and one was selected to be used to create a larger process to replace the macros and the VBA solutions. Blue Prism was the most challenging to use as a beginner, so the choice lay between Power Automate and UiPath.

The best choice was Power Automate desktop robotic tool, because it is made by Microsoft and therefore offers seamless integration with Excel. A similar automation process was created using Power Automate with less effort and expertise compared to VBA programming. As far as performance is concerned, Power Automate handled large amounts of data with ease. Automation was also developed further by adding logging functionality. It turned out that the co-operation between VBA programming and robotics would be the most ideal in terms of Excel automation. In this way the challenges of VBA programming could be alleviated with robotic tools. Furthermore, Excel macros could reduce the number of steps necessary in a robotic process.

Keywords Automation, Excel, robotics, software, Visual Basic for Applications
Pages 34 pages

Sisällys

1	Johdanto	3
2	Automaatiosta ja sen historiasta.....	4
2.1	Automaatio tietojenkäsittelyssä	5
2.2	Sovellusalueet.....	6
3	Excel ja VBA	7
3.1	Toiminnallisuus	8
3.1.1	Makrot	9
3.1.2	VBA	12
3.2	Automaatiikan hyödyntäminen Excelissä	16
4	Automaatiolle vaihtoehtoja	19
4.1	Power Automate	20
4.2	UiPath	21
4.3	Blue Prism	23
5	Tarkempaa katsausta työkaluihin	24
5.1	Käyttöönotto.....	24
5.2	Vertailu.....	27
6	Vanhojen ongelmien uudet ratkaisut	28
6.1	Toteutus.....	28
6.2	Jatkokehitys	32
7	Pohdintaa.....	33
	Lähteet.....	35

Kuvat

Kuva 1.	Excelin asetukset, mukauta valintanauha.....	9
Kuva 2.	Luottamuskeskus, Makrojen asetukset.	10
Kuva 3.	Kehitystyökalupalkki	10
Kuva 4.	Tallennetut makrot.	11
Kuva 5.	Visual Basic näkymä.....	11
Kuva 6.	VBA If-lause.....	13
Kuva 7.	VBA Case.	13
Kuva 8.	VBA FOR-lauseke.....	13

Kuva 9. VBA Do Loop.	14
Kuva 10. VBA While Wend.	14
Kuva 11. VBA merkkijonojen yhdistäminen.	15
Kuva 12. VBA MsgBox syntaksi.	15
Kuva 13. MsgBox sanomaikkuna.	15
Kuva 14. VBA Inputbox funktio.	16
Kuva 15. Inputbox syöttöruutu.	16
Kuva 16. VBA Poista otsikko koodi.	17
Kuva 17. VBA Puolipisteiden poisto.	18
Kuva 18. VBA Poistorivi.	18
Kuva 19. VBA tiedostomuutos csv-tyyppisestä xlsx-tyyppiseksi.	19
Kuva 20. Power Automate Malli.	20
Kuva 21. Power Automate RPA.	21
Kuva 22. UiPath Workflow-näkymä.	22
Kuva 23. Blue Prism -editori.	24
Kuva 24. UiPath Aktiviteetit -näkymä.	26
Kuva 25. Power Automate RPA Toteutus.	29
Kuva 26. Power Automate CSV-Exceliksi.	30
Kuva 27. Power Automate -prosessin lokitus.	31
Kuva 28. Power Automate -lokin sisältö.	32

Taulukot

Taulukko 1. Power Automate, UiPath, Blue Prism vertailutaulukko	27
---	----

1 Johdanto

Automaatio toimii suurilta osin peruspilarina osana IT-ratkaisuja tarjoten avustusta suureen työkuormaan ja itseään toistaviin tehtäviin. Automaatiikan toteuttaminen on välttämätöntä monissa projekteissa ajan ja rahallisen säästön lisäksi myös työntekijöiden hyvinvoinnin tueksi.

Opinnäytetyössä käydään läpi automatiikan ja robotiikan hyödyntämistä työelämässä, erityisesti tietojenkäsittelyssä Excel-ympäristössä. Tarkoituksena on tuoda esiin työelämässä vastaan tulleita automaatiotarpeita, sekä niiden ratkaisuja Excelin sisäänrakennettujen työkalujen avulla, joita ovat makrot ja VBA. Tämän jälkeen selvitetään muita nykyaikaisempia automatiikka- ja robotiikkatyökaluja, joilla saadaan toteutettua vastaavia ja ehkä joustavampiakin ratkaisuja työprosessien automatisointiin.

Tämä opinnäytetyö vastaa kysymyksiin: Mitä automatiikka on? Miksi automaatiota kannattaa toteuttaa? Millä eri tavoin Exceliä voidaan automatisoida?

Opinnäytetyö sai alkunsa opinnäytetyön kirjoittajan työelämän kokemuksista. Työprojektin tehtävissä vaadittiin Excel aineiston käsin muokkaamista haluttuun muotoon. Aluksi tehtiin näitä tehtäviä muutamia kertoja manuaalisesti. Suurena haasteena oli tarkistettavien ja muokattavien asioiden määrä. Aineistomäärän takia työ olisi ollut miltei mahdotonta tehdä täysin manuaalisesti, näin ollen alettiin tutkimaan mahdollisuuksia automatiikkaan. Kun kyseessä oli Excel-tiedostojen käsittely, niin tämä tarkoitti sisäänrakennettujen makrojen käyttöä. Tosin projektin edetessä oli tarve monimutkaisemmillekin ratkaisuillekin, jotka pystyttiin toteuttamaan suurella VBA-ohjelmointikielen osaamisella. Kollegalla oli yli kahdenkymmenen vuoden kokemusta kyseisestä kielestä, joten tästä syystä lähdettiin rakentamaan hyvinkin mutkikasta ja moniin tarpeisiin soveltuvia makroja suoraan Excelin VBA-ympäristöön.

Nämä mahdollistivat työn laajan sujuvoittamisen, myös tietoturvaliselta osalta, sillä virheitä ei päässyt inhimillisistä syistä sattumaan. Aineistomäärien ollessa tuhansissa, muuta vaihtoehtoa ei ole, kuin automatiikka.

2 Automaatiosta ja sen historiasta

Automatisoidun ja mekanisoidun työn eri käsitteitä on ollut olemassa vuosikymmeniä. Automaatiolla tarkoitetaan koneiden hyödyntämistä ihmisten tekemiin tehtäviin. Jatkuvasti useammin sellaisiin tehtäviin, jotka muuten olisivat mahdottomia tehdä käsin. Mekanisointi-termillä usein viitataan yksinkertaiseen ihmisen työn korvaamiseen koneilla, automaatio yleisesti ottaen tarkoittaa koneiden integroimista itsehallinnolliseen järjestelmään. (Janssen ym., 2019, ss. 99–107)

Automaatio käsitteenä on yleistynyt n. 1700-luvun puolivälissä. Alun perin kreikan kielestä substantiivi *autómaton* kantautui ranskaan (*automatique*) ja Englantiin, sittemmin Saksaan sekä muualle. Tuolloin termillä tarkoitettiin ei-koneellista automatiikkaa, kuten automaattista verenkiertoa ihmisissä ja eläimissä. Vaikkakin tuolloin eläinorganismeja yleisesti nähtiin ikään kuin koneina, eikä yksilöinä. Automatiikkaa ja koneita alettiin aluksi hyödyntää teollisuudessa yksinkertaisilla liukuhihnoilla ja höyrykoneilla. Skottilainen fyysikko ja kemisti Andrew Ure kirjoitti vuonna 1835 kuvitelman tulevaisuuden automatisoidusta tehtaasta:

Imagine, if you will, a factory as clean, spacious and continuously operating as a hydroelectric plant. The production floor is barren of men. Only a few engineers, technicians, and operators walk about a balcony above, before a great wall of master controls, inserting and checking records, watching and adjusting batteries of control instruments.

Ihmistä siis tulisi tulevaisuudessa enemmänkin koneiden valvojia ja hallinnoijia. Tätä kuvitelmaa voidaan hyvin peilata nykyajan tehtaisiin, joissa koneet ja ihmiset tekevät yhteistyötä. Monimutkaisten automatisoitujen koneiden mahdollistajia ovat tietokoneet ja niissä pyörivät ohjelmistot, joilla operoidaan näitä koneita. Nämä ovat yleistyneet 1950–1960-luvuilta lähtien. Digitaalisten tietokoneiden yleistyessä tehtaissa, tuli automaatiostakin tarkempaa. Kuitenkin toteutus rajoittui yksittäisiin laitteisiin, esim. CNC-koneisiin. (Smith ym., 2020, ss. 15–21)

Tietokoneet ovat olleet automaation mahdollistajia esim. tehdastöissä, mutta sittemmin tietokoneella itsessään suoritettavat rutiininomaiset näyttöpäätetyöt ovat yleistyneet, muun muassa tietojenkäsittelyyn liittyvät tehtävät. Näihin on myös alettu toteuttaa viime vuosikymmeninä automaatiota (Montecino, 2010). Tässä opinnäytetyössä keskitytään juuri tällaiseen IT-ympäristöissä toteutettavaan automaatioon. Tietojenkäsittelyn automaation päätavoitteena on automaatioprosessien ja ohjelmien luominen sekä toteuttaminen

korvaamaan aikaa vieviä manuaalisia tehtäviä. Tämä mahdollistaa työntekijöiden ajan priorisoinnin tärkeämpiin ja korkeamman tason tehtäviin. Automaatio kasvaa yhä keskeisemmäksi asiaksi IT-palveluntuottajien keskuudessa (IBM, 2023-a). Erilaisia työkaluja ja automaatioprosesseja käydään tarkemmin läpi myöhemmissä luvuissa.

2.1 Automaatio tietojenkäsittelyssä

Robottiprosessiautomaatio, pilviautomaatio ja tekoäly ovat keskeisessä asemassa tietojenkäsittelyn automatiikassa (IBM, 2023-b). Tässä luvussa avataan näitä käsitteitä.

Robottiprosessiautomaatio (RPA) on kätevä toteuttamaan toistuvia ja ennustettavia tehtäviä, joita ihmiset joutuvat usein manuaalisesti tekemään. RPA on sovellustason robotti, sillä ei ole fyysistä tai mekaanista olomuotoa. Tämä työkalu imitoi ihmistyöläistä menetelmillään. RPA on kustannustehokas, ketterä sekä nopea vaihtoehto automatiikalle. RPA totelee valmiiksi annettujen ohjeiden perusteella (Jovanović ym., 2019, ss. 34–39). Ympäristöjen tai työkalujen muuttuessa RPA ei kuitenkaan osaa mukautua, vaan täytyy antaa uudet ajantasaiset ohjeet. Tällainen työkalu on muun muassa UiPath.

Pilviautomaatiolla tarkoitetaan työkalujen ja prosessien hyödyntämistä, joilla automatisoidaan pilviympäristöjen aiheuttamaa manuaalista työtä. Esimerkiksi liian raskaat tai monimutkaiset työt manuaalisesti toteutettavaksi, kuten resurssien varaukset, käyttöönotot sekä päivitykset. Tämä on erittäin tärkeä ominaisuus, sillä se tarjoaa käyttäjille joustavasti tarvittavat resurssit nopeasti. Tällä tarkoitetaan itse pilvipalvelimien automatisointia (IBM, n.d.). Toinen tulkinta pilviautomaatiolle on automaatiotyökalu, joka pyörii pilvipalvelussa. Tällä on mahdollista toteuttaa monenlaisia automaatioratkaisuja muun muassa omaan tai organisaation työasemiin tai ohjelmiin, ei vaadi ohjelman paikallista asennusta työasemalle. Esimerkkinä tällaisesta ohjelmasta on Microsoft Power Automate.

Tekoälyn suosio on huimasti noussut, se on aiemmin ollut taka-alalla esimerkiksi koneoppivien mainosalgoritmien osalta, jotka mukautuvat käyttäjän kiinnostusten kohteiden mukaan. Algoritmi käsitteenä tarkoittaa melkein samaa kuin robotiikka, eli annettujen ohjeiden mukaan suoritetaan tehtävä. Koneoppivalla tekoälyllä on taito toimia epävarmassa ympäristössä, ja se osaa käsitellä todennäköisyyksiä (Suomen Koodikoulu, 2018, ss. 3–10). Suurten kielimallien (LLM) tekoälyt, kuten ChatGPT ovat tulleet tavallisten

ihmisten käyttöön. Tekoälyn kehitys on ollut niin nopeaa viime vuosien ajan, että nykyään voidaan jo automaattisesti generoida ohjesanoilla kokonaisia esseitä, kuvia, taidetta tai videoita uskottavasti. Nykyisin on entistä vaikeampaa havaita, mikä on oikean ihmisen luomaa ja mikä generoitua. Toisaalta esimerkiksi tekoälytaiteen takana on ihminen, joka on antanut ohjeet, näin ollen teknisesti perimmäinen luoja on ihminen. Nimenomaan tällainen generatiivinen tekoäly on imitaattori. Se uudelleenkäyttää vain dataa, mihin sillä on pääsy (Baker, 2024). Tämä ei ollut mahdollista muutama vuosi sitten. Tekoälyä valjastetaan jo esimerkiksi potilas- ja asiakastietojärjestelmiin, jossa se käy läpi potilaan tekstit ja tekee niistä tiivistelmän ammattilaisen avuksi. Tarkoituksena ei ole korvata lääkäreitä vaan toimia tehokkaana avustajana (CGI, 2024). Tekoälyn tulevaisuus onkin hieman mysteeri, sillä kehitys on niin nopeaa ja erilaisia innovaatioita tehdään usein.

Työelämän näkökulmasta tärkeimmät hyödyt automaatiolla tulevat IT-operaatioiden sujuvoittamisesta ja näin ollen tehokkuuden lisäämisestä. Vuoden 2019 tutkimuksessa IT-tiimit ilmoittivat käyttävänsä noin 70 % ajastaan yksitoikkoisten ja aikaa vievien töiden tekemiseen (Bommadevara ym., 2019). On siis selvää, että työntekijän aikaa säästyisi huomattavasti ja he voivat keskittyä merkityksellisempiin töihin. Tämä tehostaa innovaatioiden syntymistä ja työntekijöiden yleistä tyytyväisyyttä. Automaatio suurilta osin vähentää inhimillisten virheiden sattumisia ja parantaa tarkkuutta. (Lee, 2024)

Tällaisella automaatiolla on kuitenkin varteenotettavia varjopuoliakin. Suurimpana huolenaiheena on töiden liiallinen väheneminen ja töiden katoaminen. Mikäli työntekijän tehtävät ovat ennustettavia alhaisen tason töitä, niin automaatio saattaisi syrjäyttää kyseisen henkilön. Mahdollista on, ettei tämän kaltaisten työtehtävien paikkoihin enää palkata uusia työntekijöitä. Automaation toteutuskustannukset voivat nousta suuriksi, vaikka pitkällä aikavälillä säästetään rahaa. Laitteet, ohjelmistot, henkilöstön perehdytys ja prosessien uudelleenrakentaminen maksavat, mikä on haaste pienemmille yrityksille. Laitteisiin ja sovelluksiin myös aika ajoon syntyy häiriöitä, tällaiset voivat hidastaa yrityksen toimintaa. Liiallinen automaatio voi myös aiheuttaa negatiivisia tuloksia, mikäli ihmiskontakti puuttuu, muun muassa automatisoitu asiakaspalvelubotti. (Badkar, 2024)

2.2 Sovellusalueet

Automaation sovellusalueita on lukuisia. Näitä eri menetelmiä voidaan hyödyntää eri sovellusalueissa vastaamaan omia tai yrityksen tarpeita. Nämä ovat muun muassa ohjelmistojen ja järjestelmien käyttöönoton automatisointi ja konfigurointi.

Valvonta ja hälytykset, eli monitoroidaan järjestelmän suorituskykyä ja tehdään ilmoitus, jos jossakin vikaa tai hitautta. Automaatiolla toteutetaan ennakoivia toimenpiteitä valvontahälytysten perusteella, kuten palvelimen uudelleenkäynnistys.

Asiakaspalvelun automaattinen tiketin luokittelu, vastaaminen, nimeäminen, kategorisointi. Nämä ovat vain alkutoimenpiteitä, viimeisen tarkistuksen ja ratkaisun tekee asiantuntija. (Şimşek, 2024)

Aiemmin mainittua prosessiautomaatiota hyödynnetään pääosin sellaisiin tehtäviin, joissa käyttäjältä vaaditaan muun muassa näppäimistön tai hiiren painalluksia ja eri sovelluksissa toimimista. Robotti käyttää samaa rajapintaa, kuin oikea ihminen. RPA:ta siis käytetään hyvin ketterästi eri toimialoilla, joissa tietokoneet ovat keskeisessä osassa. Muun muassa talouspalveluihin ja pankkeihin, vakuutuksiin, terveydenhuoltoon, teollisuusalaan ja moneen muuhun. (Veenendaal, n.d.)

Hyvä automaation kohde on myös sellaiset työt, jossa täytyy tietyn kaavan, tai tiettyjen ehtojen mukaan muokata esimerkiksi useita määriä Excel-tiedostoja. Tähän sovelletaan makroja, eli ikään kuin pieniä ohjeita siitä, mitä käyttäjä tekisi. Kuten kirjoittamalla tiettyyn Excelin soluun tekstiä tietyillä ehdoilla tai poistamalla otsikot ja tallentamalla tietynlaisena tiedostotyyppinä eteenpäin käsiteltäväksi. Excel itsessään tukee makroja hyvinkin joustavasti mahdollistaen myös koodipohjaisen muokkauksen. Excel-makroja ja sen ohjelmointia tutkitaan seuraavassa luvussa syvemmin ja käydään läpi, mitä ongelmia kyseisillä työkaluilla ratkaistu oikeassa työelämän projektissa.

3 Excel ja VBA

Microsoftin Excel on hyvin suosittu taulukkolaskentaohjelma ja monissa liiketoiminnoissa hyödynnetään Excelin ominaisuuksia. Mikäli työnkuvat vaativat Excelin kanssa työskentelyä niin on suotavaa automatisoida mahdollisimman moni ennalta arvattava tehtävä.

Kuten opinnäytetyön johdantoluvussa mainittiin, tätä menetelmää käytettiin työelämän projektissa, jossa käsiteltiin ja muokattiin Excel-aineistoja tietyin tavoin. Projektin alussa aineistoihin tehtiin manuaalisesti muutokset, tämä oli todella yksityiskohtaista työtä ja virheitä sattui. Uusien työntekijöiden perehdyttäminen oli haastavaa, sillä työvaiheita oli paljon, eikä kaikilla ollut Excel-osaamista.

VBA eli Visual Basic for Applications -ohjelmointikieli ja makrot Excelin työkaluina ovat vähemmän tunnettuja ominaisuuksia, mutta täysin tuettuja ja ilmaisia ratkaisuja. VBA:lla luodaan kooditasolla Exceliä automatisoivia makroja. VBA on ohjelmointikieli, mutta samanaikaisesti toimii makron ohjauskielenä. Excelin ohjeessa VBA:lla tehtyjä ohjelmia kutsutaan makroiksi. (Kusleika, 2022)

3.1 Toiminnallisuus

Excelin automatiikkaa käytetään joustavasti vastaamaan omia tarpeita. Periaatteessa kaiken mitä voi käsin tehdä Excelillä, makro pystyy samaan. Käyttötapauksia on siis useita ja taipuu moneen tehtävään.

Tehtävät ovat esimerkiksi suurien tekstimäärien täyttäminen asiakirjoille, kuten oma nimi, puhelinnumero tai muu. Makro pystyy hoitamaan tämän automaattisesti. Tätä voi kehittää niin pitkälle kuin mielikuvitus ja osaaminen antaa myöden.

Automaattinen raporttien ja kaavakkeiden muodostaminen onnistuu myös VBA:ta hyödyntäen. Automaation toteuttamisen ehtona on vain se, että tehtävä työprosessi olisi selkeä ja mutkaton.

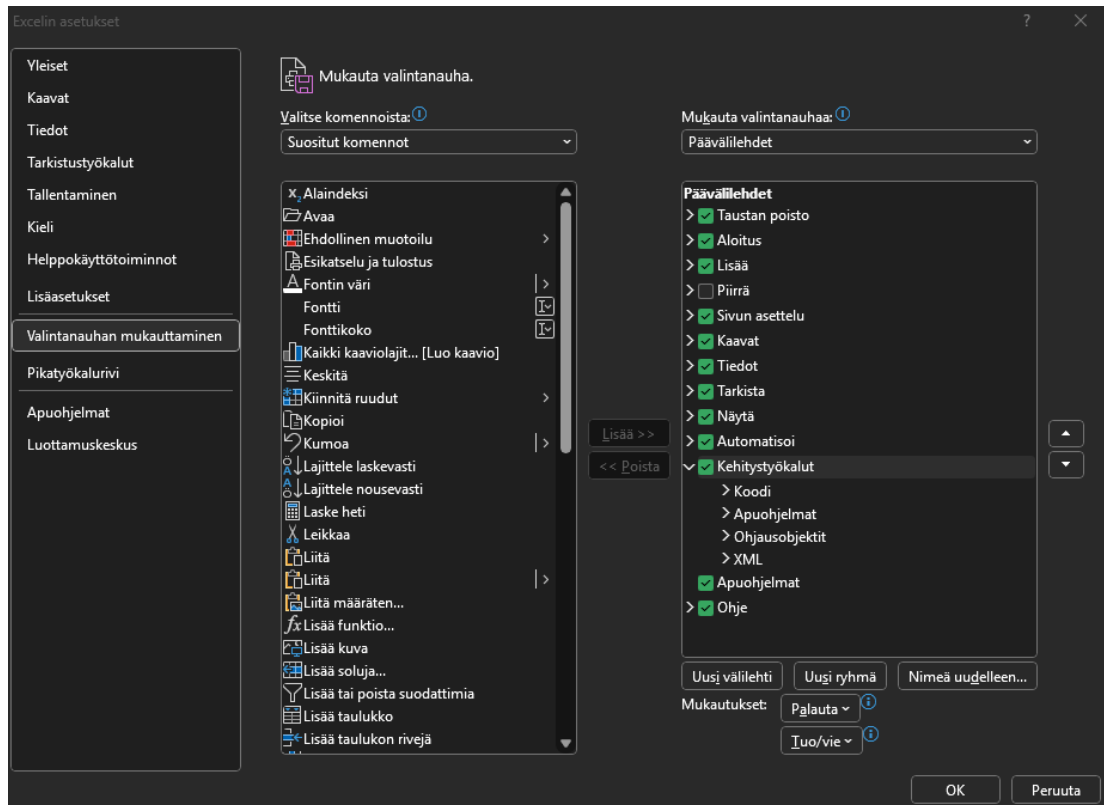
Excelissä on sisäänrakennettuna ominaisuutena makrojen nauhoittaminen, eli tallennetaan ohjeet Exceliin tehtävistä toimenpiteistä. Mikäli on useita asiakirjoja, joille pitäisi kaikille tehdä samat tehtävät, niin ensimmäisen asiakirjan muokkauksen yhteydessä nauhoitetaan makro. Tämän jälkeen ei tarvitse muuta kuin muille jäljellä oleville asiakirjoille suorittaa äskettäin luotu makro, joka tekee samat asiat automaattisesti. Usein käytettyjä komentoja voidaan tallentaa ja ajaa tarpeen mukaan, kuten tietyt tyylittelyt tai solujen värjäykset.

Vaikka Excelissä on valmiina lukuisia asiakirjan funktioita, kuten SUMMA tai JOS-funktiot, niin omia mukautettuja funktioitakin on mahdollista tehdä. Näin saadaan yksinkertaistettua pitkiä funktiolausekkeita, jotka voivat auttaa asiakirjan automaattisessa käsittelyssä. Kaikki tehdyt makrot säilyvät omina prosesseinaan ja niihin pääsee käsiksi esimerkiksi pikatyökaluriviltä. (Kusleika, 2022)

3.1.1 Makrot

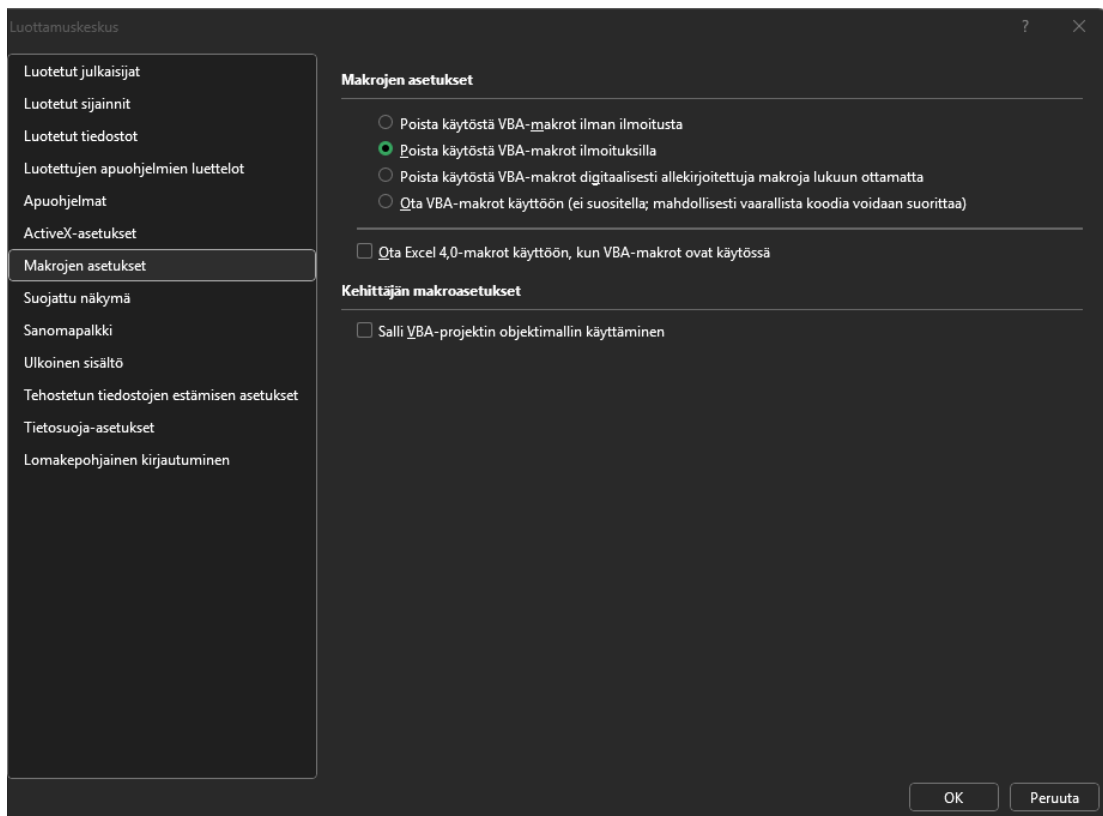
Käytännössä makrojen asettaminen on yksinkertaista, eikä edellytä aiempaa osaamista. Ensin täytyy tuoda Excelin kehitystyökalut näkyviin valintanauhaan. Tämän saa päälle asetuksista laittamalla rasti Kehitystyökalut kohtaan Mukauta valintanauha -näkyvässä (kuva 1).

Kuva 1. Excelin asetukset, mukauta valintanauha.



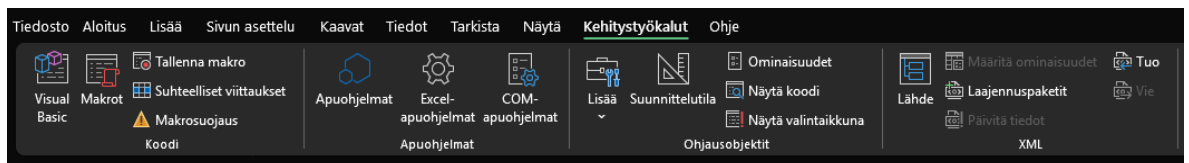
Oletuksena Excel ei salli VBA-makrojen ajamisia turvallisuussyistä. VBA-makrojen ajamisoikeudet asetetaan päälle Excelin Luottamuskeskuksesta (sivu 10, kuva 2). Suositellaan kuitenkin, ettei ajeta muita kuin omia makroja.

Kuva 2. Luottamuskeskus, Makrojen asetukset.



Makroja tallennetaan Kehitystyökalu-ikkunasta Tallenna makro -painikkeesta (kuva 3) tai asiakirjan vasemmassa alalaidassa olevasta symbolista. Ohjelma pyytää vielä täyttämään lisätietoja makrolle kuten nimi ja mahdollinen pikanäppäin, sekä makron tallennuskohde. Tämän jälkeen nauhoitetaan työvaiheet, eli tavalliseen tapaan tehdään asiakirjaan halutut muutokset. Tallennuksen lopetus onnistuu samoista painikkeista.

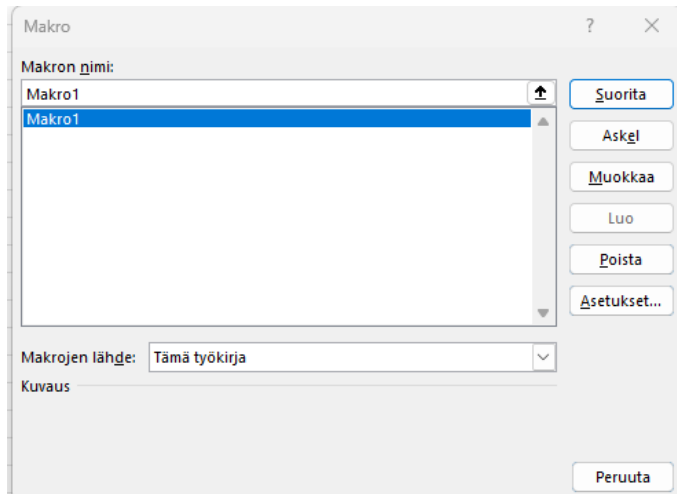
Kuva 3. Kehitystyökalupalkki.



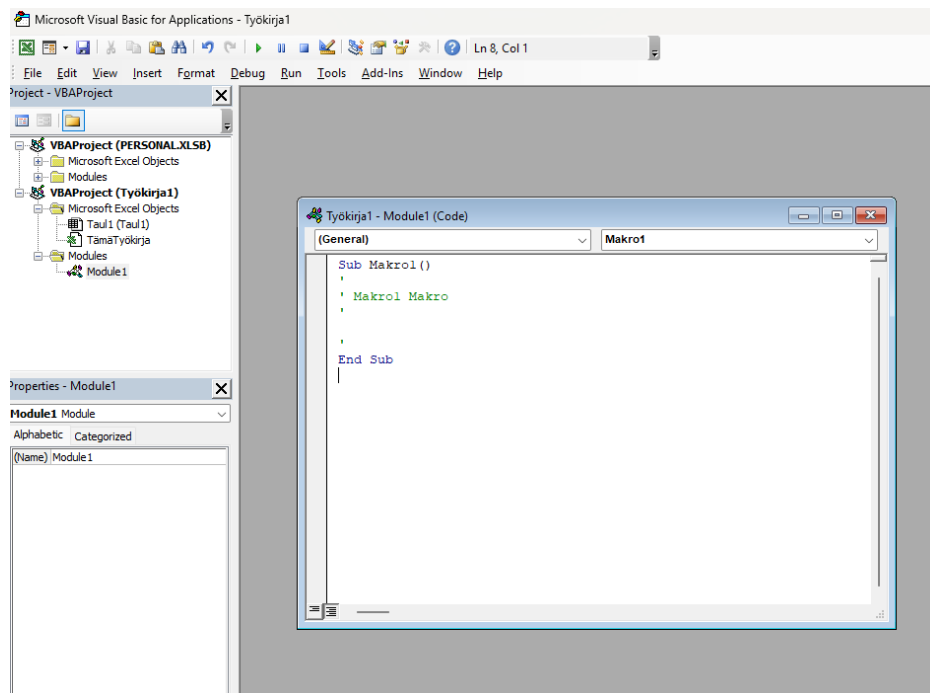
Tallennetut makrot löytyvät kuvan 3 kehitystyökalupalkista Makrot-painikkeen alta, niiden ajo onnistuu myös samasta paikasta (sivu 11, kuva 4). Makrot ovat VBA-ohjelmia, eli kun tallennetaan makro niin Excel luo siitä VBA koodipohjaisen ohjelman. VBA-ympäristöön

pääsee kehitystyökalupalkista Visual Basic -painikkeella, ALT+F11 näppäinyhdistelmällä tai tallennettujen makrojen ikkunan muokkaa-painikkeesta. Ympäristö aukeaa erilliseen ikkunaan, jossa näkyy VBA-koodina luodut makrot. Tässä ympäristössä muokataan makroja kooditasolla (kuva 5).

Kuva 4. Tallennetut makrot.



Kuva 5. Visual Basic näkymä.



3.1.2 VBA

Kaikkia ehtoja tai muita yksityiskohtia ei saada pelkällä makrojen tallennuksella täysin automatisoitua. Tähän tarvitaan siis konepellin alle katsomista ja VBA-ohjelmointia. Ohjelmointi tehdään edellisen alaluvun kuvan 5 koodieditorissa. Tässä alaluvussa tehdään pieni pintaraapaisu VBA-koodirakenteeseen. Tässä kohtaa yleinen ohjelmointikokemus auttaa, VBA:ssa on samanlaisia rakenteita kuin muissakin ohjelmointikielissä.

Makrot ovat aina omien aliohjelmiensa alla, kuten kuvassa 5 huomataan. Tallennetun Makro1:n aliohjelma on määritelty aloittaen *Sub* ja lopetetaan *End Sub*. Mikäli makroja on useita, tulevat ne allekkain saman moduulin alle, mikäli tallennuspaikka on sama.

Muuttujiin määritetään erilaisia tietotyyppejä, eli millaista dataa milläkin muuttujalla on. Tietotyyppejä on enemmän kuin yleisimmissä ohjelmointikielissä. Tietotyypit ovat seuraavat: Boolean, Byte, Collection, Currency, Date, Decimal, Dictionary, Double, Integer, Long, Object, Single, String, Variant. (Microsoft, 2023)

Muuttujat yleensä alustetaan tietotyypeillä ja siinä käytetään DIM (declare in memory) -komentoa esimerkkinä "*DIM muuttuja as Tietotyyppi*". Tietotyyppien tilalle asetetaan esimerkiksi merkkijonoille *String* tai numeroille *Integer*.

Mikäli alustamista ei tehdä, esim. *DIM muuttuja*, Excel oletuksena asettaa muuttujalle tietotyyppiksi *Variant*. Näin muuttuja ei ole rajattu vain yhteen tietotyyppiin, vaan siihen pystyy asettamaan usean eri tietotyypin dataa.

Vertailuoperaattorit ovat vastaavia, kuin muissa ohjelmointikielissä.

Yhtä suuri kuin: $X = Y$

Erisuuri kuin: $X <> Y$

Pienempi ja suurempi kuin: $X < Y, X > Y$

Pienempi ja suurempi tai yhtä suuri kuin: $X <= Y, X >= Y$

Ehtolauseet pääsääntöisesti tapahtuvat If-lausekkeella (kuva 6).

Kuva 6. VBA If-lause.

```
If x = 1 Then
    Suoritettava lause
ElseIf x = 2 Then
    Suoritettava lause
Else
    Suoritettava lause
End If
```

Elsif-komennolla toteutetaan useita ehtoja yhden If lauseen sisälle. Vaihtoehtoisesti on mahdollista käyttää Case-lausetta, jossa suoritetaan tietyn Case-arvon alla oleva funktio (kuva 7).

Kuva 7. VBA Case.

```
Select Case valinta
Case 1
    Suoritettava Lause
Case 2
    Suoritettava Lause
Case N
    Suoritettava Lause
Case Else
    Lause, jos mikään edellisistä ei täyty
End Select
```

Toistorakenne tai silmukka voidaan toteuttaa usealla eri tavalla. Esimerkkinä FOR, Do Loop tai While Wend, FOR soveltuu ennalta tiedetyn määrän toiston tekemiseen, kun taas Do Loop sellaisiin missä toistettava määrä vaihtelee. Kuvassa 8 näkyy FOR-lauseke.

Kuva 8. VBA FOR-lauseke.

```
' FOR-lauseke
For Laskuri = alkuarvo To loppuarvo
    Suoritettava lause
Next Laskuri
```

Do Loop -lauseke on mahdollista tehdä kahdella tapaa, Do ja Do While. Do alkuisen silmukan alla olevat lauseet suoritetaan vähintään kerran, ehdot kerrotaan silmukan

lopussa. Do While -silmukassa katsotaan ehdot ensin, näin ollen silmukan sisään ei välttämättä mennä (kuva 9).

Kuva 9. VBA Do Loop.

```
' Do Loop, vähintään kerran suoritettava
Do
    Suoritettava lause
Loop [{While | Until} ehto]
' Do Loop, ehtotarkistus ensin
Do [{While | Until} ehto]
    Suoritettava lause
Loop
```

Perinteisempi While Wend -silmukka toistetaan niin, että ehdon ollessa tosi, suoritetaan silmukan sisällä oleva funktio. Kun päästään Wend kohtaan, suoritetaan sama uudelleen. Jos ehto ei enää täyty, niin suoritus jatkuu Wend lauseesta eteenpäin (kuva 10).

Kuva 10. VBA While Wend.

```
' While Wend
Dim Laskuri
Laskuri = 0
While Counter < 10
    Counter = Counter + 1
Wend
```

Excel asiakirjoja muokatessa voi olla hyödyllistä yhdistellä merkkijonoja ja tallentaa ne muuttujiin. Tämä onnistuu & ja + merkeillä.

merkkijono = merkkijono1 & merkkijono2

merkkijono = merkkijono1 + merkkijono2

Tästä esimerkkinä otetaan soluista A1 ja B1 merkkijonot, tehdään niille yhdistys ja asetetaan yhdistetty merkkijono C1 soluun. Jos A1 olisi "Testi" ja B1 "nen", tulostuisi C1 soluun Testinen (sivu 15, kuva 11).

Kuva 11. VBA merkkijonojen yhdistäminen.

```
Sub merkkijono_yhdistys()
    Dim ensimmäinen_arvo, toinen_arvo, yhteinen_arvo As Variant
    ensimmäinen_arvo = Range("A1").Value
    toinen_arvo = Range("B1").Value
    yhteinen_arvo = ensimmäinen_arvo & toinen_arvo
    Range("C1").Value = yhteinen_arvo
End Sub
```

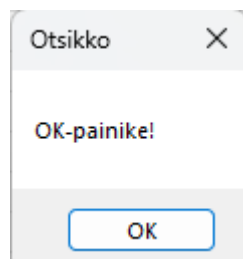
VBA:n kautta pystytään näyttämään viestejä käyttäjälle sanomaikkunalla tai kysyä käyttäjältä arvoa johonkin syöttöikkunalla, esimerkiksi omaa nimeään. Nämä ovat *Msgbox* ja *Inputbox*. Alla on niistä esimerkit.

Msgbox-funktioon (kuva 12) saa esimerkiksi OK merkin (kuva 13). On mahdollista asettaa muitakin painikkeita vaihtamalla MsgBox-funktion numeroarvoa, esimerkissä on 0, eli pelkkä OK.

Kuva 12. VBA MsgBox syntaksi.

```
Sub viesti()
    Dim Arvo As Integer
    Arvo = MsgBox("OK-painike!", 0, "Otsikko")
End Sub
```

Kuva 13. MsgBox sanomaikkuna.

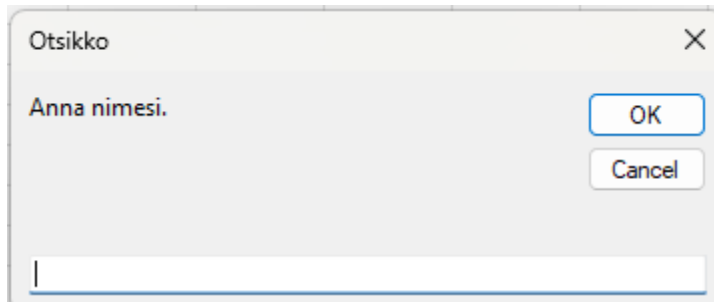


Inputbox-funktiolla (sivu 16, kuva 14) kysytään käyttäjältä tietoja erillisellä ikkunalla (sivu 16, kuva 15). Käyttäjän antama arvo tulee *syöte*-muuttujan arvoksi.

Kuva 14. VBA Inputbox funktio.

```
Sub syottoruutu()
    Dim syote As String
    syote = InputBox("Anna nimesi.", "Otsikko")
End Sub
|
```

Kuva 15. Inputbox syöttöruutu.



Tämän luvun esimerkit ovat peräisin lähteestä (Jyväskylän yliopisto, 2005).

Toiminnallisuuksia on lukuisia ja näiden soveltaminen omiin tarpeisiin onnistuu ketterästi. Ohjelmointia helpottaa huomattavasti se, kun makroa tallentaa ja tehdään muutoksia Excelliin, niin tehdyn operaation koodi generoituu reaaliajassa automaattisesti VBA-ympäristöön tarkasteltavaksi ja muokattavaksi.

3.2 Automatiikan hyödyntäminen Excelissä

Excel oli suuressa osassa yhtä IT-yrityksen projektia, tähän projektiin liittyi paljon aineistojen muokkausta, ja tähän osui sopivasti käyttää makroja sekä VBA:ta. Käydään pelkistetysti läpi, millaisista työtehtävistä oli kyse, joihin pystyttiin toteuttamaan automatiikkaa. Töiden perusidea oli ladata palvelimelta Excel-aineisto, jossa erinäistä tietoa. Nämä aineistot katsottiin läpi ja tehtiin tarvittavat muutokset sekä joissakin tilanteissa tallennettiin tietyn tyyppisenä tiedostona. Tämän jälkeen vietiin ne erilliselle palvelimelle ja ajettiin tiedot tietokantaan.

Tehtävissä mainitaan käsitteet csv ja xlsx. CSV (Comma-Separated Values) on tiedostomuoto, johon on tallennettu taulukkomuotoinen data tekstimuotoiseksi. Verrattuna Excel tiedoston soluihin, csv-tiedostossa solut ovat eroteltu esimerkiksi pilkulla tai

puolipisteellä ja jokainen rivi vastaa omaa datariviään. XLSX on Excel-asiakirjojen oletusarvoinen tiedostomuoto. (Google, 2025)

Alle listattu kolme yleisintä tehtävää, joiden tekeminen manuaalisesti olisi vienyt ylimääräistä aikaa varsinkin aineistojen kokojen ollessa suuria. Todellisuudessa automatisoitavia tehtäviä ja huomioitavia seikkoja oli huomattavasti enemmän, näistä esimerkeistä nuo karsittu pois selkeyden vuoksi:

1. Poistetaan xlsx-asiakirjan otsikkorivi, tarkastetaan viimeinen rivi puolipistemerkkien osalta, eli jos lopussa on tyhjiä ”;” -merkkejä niin nekin rivit poistetaan. Tehdään tallennus csv-tiedostona omaan kansioon.
2. Tarkastetaan xlsx-asiakirjan ensimmäisen solun rivit, poistetaan sellaiset rivit, jotka alkavat tietyllä merkkijonolla. Tämän lisäksi tehdään sama, kuin edellisellä eli tallennetaan csv-tiedostona omaan kansioon.
3. Avataan csv-tiedosto ja tallennetaan xlsx-asiakirjana.

Ensimmäiseen tehtävään luotiin aluksi vain otsikkorivin poisto. Ensin tarkastettiin, että kyseessä on otsikkorivi, eikä datarivi ja kun tarkistus meni läpi, niin poistettiin ensimmäinen rivi ja tallennettiin csv-tiedostona. Koodipuolen näkymää kuvassa 16, tallennus tapahtui niin, että otettiin aktiivisen tiedoston tiedostonnimi ja poistettiin siitä xlsx-tiedostopäätte ja tallennettiin se Exceltesti kansioon csv:nä.

Kuva 16. VBA Poista otsikko koodi.

```

Sub PoistaOtsikko()
    Dim strTiedosto As String
    Dim polku As Variant
    ' Poista otsikkorivi
    If Range("A1").Value = "Otsikko" Then
        Rows(1).Delete
    End If
    strTiedosto = Left(ActiveWorkbook.Name, InStr(1, ActiveWorkbook.Name, ".") - 1) & ".csv"
    polku = "C:\Exceltesti\" & strTiedosto
    ' Tallenna tiedosto .csv-muodossa
    ActiveWorkbook.SaveAs Filename:=polku, FileFormat:=xlCSV, CreateBackup:=False
End Sub

```

Tämän jälkeen huomattiin, että aineistoon saattaa tulla loppuun pelkkiä puolipisteitä. Niinpä koodia muokattiin, jotta nekin saadaan poistettua (sivu 18, kuva 17). Tämä lisättiin saman aliohjelman alle niin kaikki tapahtui samaan aikaan.

Kuva 17. VBA Puolipisteiden poisto.

```
ActiveCell.SpecialCells(xlLastCell).Select
Range(Selection, Cells(1)).Select
Selection.Replace What:=";", Replacement:="", LookAt:=xlPart, _
    SearchOrder:=xlByRows, MatchCase:=False, SearchFormat:=False, _
    ReplaceFormat:=False
```

Toiseen työtehtävään, jossa pitää poistaa A sarakkeelta tietyn merkkijonon omaavia rivejä sovellettiin edellisen ongelman otsikkotarkistusta. Tähän tarvittiin myös silmukkarakennetta, joka käy sarakkeen A arvoja läpi ja tekee poistot. Koodi poistaa ne rivit, joissa A sarakkeella lukee Poisto (kuva 18).

Kuva 18. VBA Poistorivi.

```
Sub PoistaRivitArvollaPOISTO()
    Dim viimeinenRivi As Long
    Dim i As Long

    ' Etsi viimeinen rivi, jossa on dataa
    viimeinenRivi = Cells(Rows.Count, "A").End(xlUp).Row

    ' Käy läpi rivit alhaalta ylöspäin
    For i = viimeinenRivi To 1 Step -1
        If UCase(Cells(i, "A").Value) = "POISTO" Then
            Rows(i).Delete
        End If
    Next i
End Sub
```

Kolmas tehtävä toteutettiin niin, että luetaan kansion sisältö ja muutetaan kaikki kansiossa olevat csv-tiedostot xlsx-tiedostoiksi, sekä poistetaan alkuperäinen csv (sivu 19, kuva 19). Do while silmukalla tehdään toimenpiteet kansion tiedostoille. Tehdyt toimenpiteet näkyvät kuvassa 19 virheällä tekstillä. Ainoa muuttuja koodissa on tiedostopolku, josta luetaan tiedostot.

Kuva 19. VBA tiedostomuutos csv-tyyppisestä xlsx-tyyppiseksi.

```

Do While workfile <> ""
'   Avataan CSV tiedosto
Workbooks.Open Filename:=folderName & workfile
'   otetaan vanhan CSV tiedoston nimi
oldfname = ActiveWorkbook.FullName
'   Muutetaan XLSX
newfname = folderName & Left(ActiveWorkbook.Name, Len(ActiveWorkbook.Name) - 4) & ".xlsx"
ActiveWorkbook.SaveAs Filename:=newfname, FileFormat:=xlOpenXMLWorkbook, CreateBackup:=False
ActiveWorkbook.Close
'   Poistetaan vanha CSV tiedosto
Kill oldfname
Windows(myfile).Activate
workfile = Dir()
Loop

```

Edellä mainittuihin ratkaisuihin ja myös niiden tarpeisiin päästiin tekemisen kautta. Ensin havaittiin tarve, johon arvioidaan automatiikan toteutusta testauksilla. Tämän jälkeen otettiin käyttöön, mikäli oli mahdollista toteuttaa. Muutoksien toteuttaminen Excel VBA:lla kävi nopeasti sen modulaarisuutensa takia, sillä makroja voidaan yhdistää toisiinsa ja toiminnallisuuksia sitoa yhden ajettavan makron alle. Edellisiä toteutuksia ei siis tarvitse lähteä muuttamaan, vaan uusia funktioita lisätään erillisinä lisäosina.

VBA-ohjelmoinnissa haasteena on ohjelmointikielen opetteleminen ja soveltaminen. Jos se ei ole toiminnallisuuksiltaan tuttua, niin ei välttämättä osata hahmottaa automaation mahdollisuuksia. VBA vaatii syvempää osaamista tehokkaiden lopputulosten saamiseen, tässä projektissa oli mukana henkilö, jolla useiden kymmenien vuosien kokemusta VBA-ohjelmoinnista.

4 Automaatiolle vaihtoehtoja

Mikäli luvussa 2 mainittuja ratkaisuja halutaan viedä pidemmälle, vaatii se syvempää perehtymistä Excel-ympäristöihin ja tietenkin rajoitteita on. Exceliä pidemmälle ei voida mennä VBA-automatiikan kanssa, vaikka VBA ja makroja on saatavilla Microsoftin muissakin tuotteissa, kuten Outlook ja Word. Nämä kuitenkin ovat aiheita erikseen, eivätkä kuulu tämän työn aihealueeseen. Työtehtävät Excelin muokkaamisen lisäksi vaativat tiedostojen viemistä palvelimelle ja kommentojen ajamista. Tämän tyyppisiä tehtäviä ei voi VBA:lla toteuttaa.

Seuraavaksi tutkitaan vaihtoehtoja työkaluista, joilla pyritään toteuttamaan sama, ellei enemmänkin. Aiheena pääosin nyt jo tutuksi tulleet RPA- ja pilviautomaatiotyökalut. Nämä työkalut, joita tarkastellaan ovat nimeltään Power Automate, UiPath ja Blue Prism. Seuraavissa luvuissa avataan jokaisen työkalun toiminnallisuutta, sekä mitä kaikkea

käyttöönotto vaatii ja voidaanko niitä käyttää pilvipalveluna vai paikallisesti omalla työasemalla.

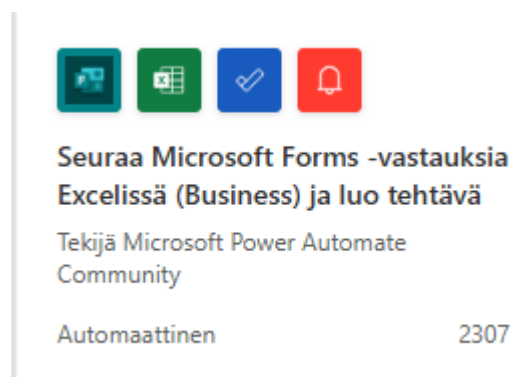
4.1 Power Automate

Power Automate (aiemmalta nimeltään Flow), julkaistu 2016, on Microsoftin pilvipohjainen sovellus, jolla luodaan automaatioprosesseja. Nämä ovat nimeltään työnkulkua eng. Flow, ja ne toimivat hyödyntämällä liittimiä, jotka yhdistävät sovelluksia toisiinsa. Sovellukset eivät rajoitu vain Microsoftin omiin toteutuksiin. Käyttäen näitä liittimiä on mahdollista luoda automaatiikkaa monien eri ympäristöjen välillä. Toteutusmenetelmiä tai liipaisumahdollisuuksia on myös useita (Stork, 2023). Alla listausta Power Automaten eri käyttötavoista:

- Automaattinen pilvityönkulku. Käynnistetään määrättyllä tapahtumalla.
 - Välitön pilvityönkulku. Käynnistetään manuaalisesti tarvittaessa.
 - Ajoitettu pilvityönkulku. Valitset itse, milloin ja miten usein ne suoritetaan.
 - Suunnittele kuvailemalla (beta). Kuvaile haluamaasi työnkulkua ja tekoäly luo sellaisen sinulle.
 - Työpöytätyönkulku. Automatisoi työpöytäympäristön prosessit.
 - Prosessilouhinta. Arvioi ja optimoi olemassa olevia prosesseja ja tehtäviä.
- (Microsoft, n.d.-a)

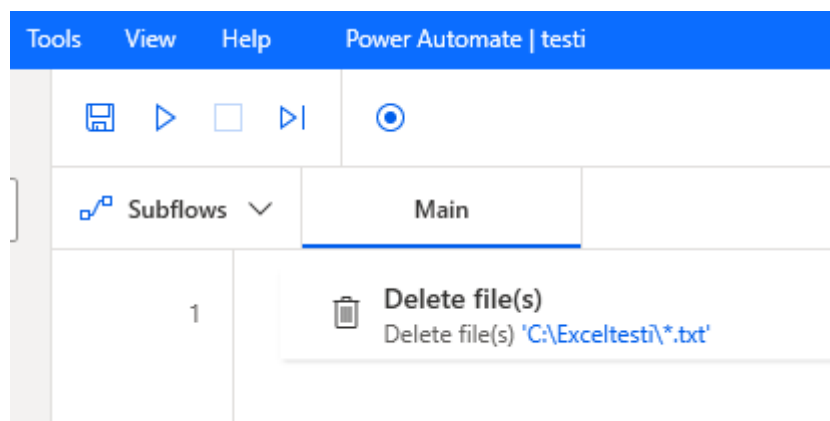
Power Automatella on laaja yhteisö, ja automaatiomalleja voidaan jakaa muidenkin käyttöön. Microsoft myös tehnyt useita valmiita pohjia automaatiolle (kuva 20).

Kuva 20. Power Automate Malli.



Power Automate tarjoaa myös paikallisesti asennettavan RPA-sovelluksen työasemaakohtaiseen automatiikkaan. Tähän ei ole valmiita malleja, sillä työasemat ovat vaihtelevia. Täytyy siis luoda alusta omia tarpeita vastaava prosessi. Tässä on toteutettu automaattinen prosessi, joka poistaa tietyistä kansioista txt-päätteiset tiedostot (kuva 21). Tämän prosessin luomiseen tarvitaan vain yksi Delete files -funktio, johon määritetään polku ja tiedosto tiedostopäätteen mukaan.

Kuva 21. Power Automate RPA.



4.2 UiPath

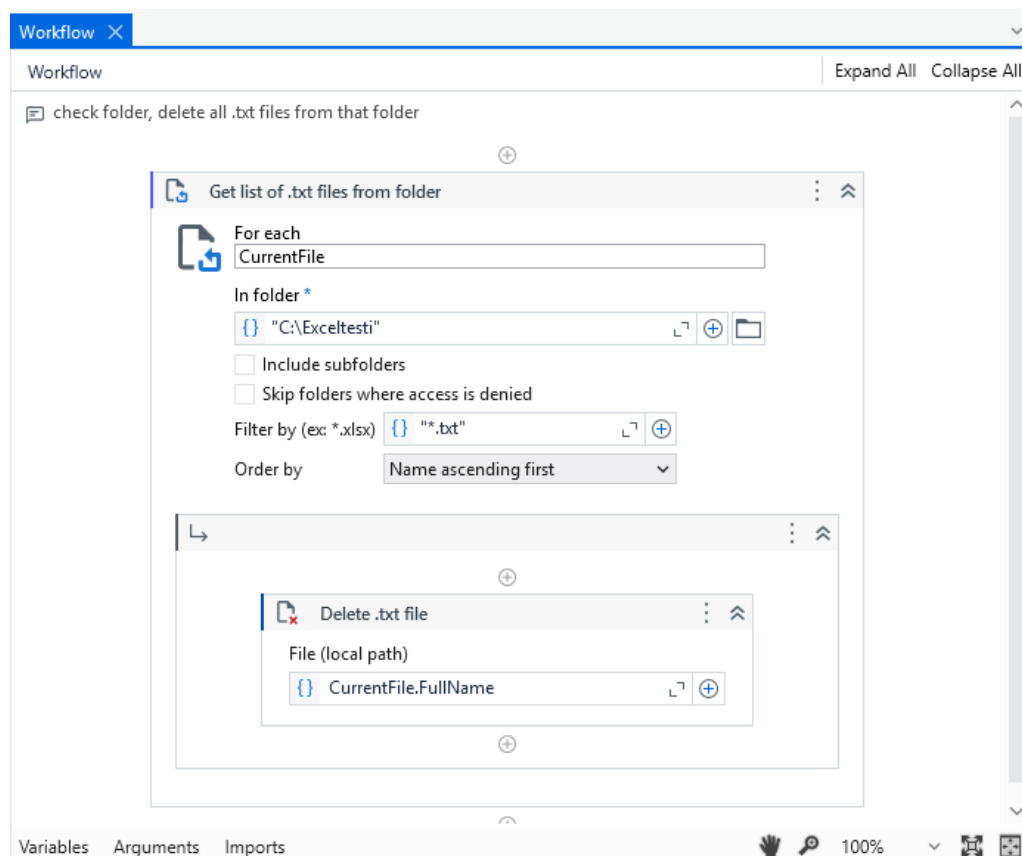
UiPath on täysverinen robottiprosessiautomaatio-ohjelma, joka on saanut alkunsa Romaniassa vuonna 2005 (Gheorghe, 2018). UiPath tarjoaa neljää erilaista versiota:

- Community Cloud, yleisesti pienet tiimit tai yksittäiset henkilöt käyttävät tätä automaation harjoitteluun.
- Studio, tarkoitettu yksilöille, jotka haluavat käyttää UiPathia yritystasolla.
- Enterprise Cloud, pilvipalveluna toimiva RPA-alusta, joka sisältää tuotteet Studio, Robots ja Orchestrator.
- Enterprise Server, tarjoaa samat tuotteet kuin Enterprise Cloud, mutta tämä toimii paikallisena palvelinasennuksena.
(Kappagantula, 2023)

UiPath Automation Cloud vastaa hyvin pitkälle Power Automaten pilvipalveluna toimivaa automaatiota. Tässä työssä tarkastellaan UiPathin Studio versiota.

Kuten Power Automatella, niin myös UiPathilla pääsee käsiksi moniin eri sovelluksiin. Mukaan lukien itse Windows työasemaan, kuten resurssienhallintaan. Automaation toteutus tapahtuu liittämällä moduuleita yhteen ja käyttämällä esimerkiksi operaattoreita, funktioita ja silmukoita. Ulkoasu ja automaatioiden rakenne näyttää vastaavalta, kuin esimerkiksi Scratch-graafinen ohjelmointikieli, jossa vedetään esimerkiksi ehtolausepalikka ja liittää siihen tehtäviä toimintoja. Kuten Power Automateen, niin UiPathiin on myös hyödynnetty tekoälygenerointia, johon kirjoitetaan vapaamuotoisesti, mitä halutaan automatisoida. Kuvassa 22 on UiPathia pyydetty luomaan prosessi, jossa käydään läpi kansio ja poistetaan txt-päätteiset tiedostot. Huomiona, että UiPath latsi tämän yhteydessä kaikki tarvittavat lisäosat. Tekoälyn generoima prosessi verrattuna Power Automateella toteutettuun on turhan monimutkainen, mutta kuitenkin toimiva. Tämä liiallinen monimutkaisuus kannattaa ottaa huomioon, jos käyttää tekoälytoimintoa prosessien luomiseen. Suurissa aineistomäärissä saattaa tulla hidasteita tämän prosessin kanssa, sillä prosessi ottaa kaikista kansion tiedostoista listan talteen, johon tekee vertailut.

Kuva 22. UiPath Workflow-näkymä.



4.3 Blue Prism

Blue Prism on RPA työkalu, lähempänä UiPathia kuin Power Automatea. Blue Prism on luotu vuonna 2001 Englannissa. Yhtiö julkaisi ensimmäisen tuotteen Automate vuonna 2003, tätä kutsutaan RPA-käsitteen syntyperäksi. Myöhemmin vuonna 2022 finanssi- ja terveydenhuollon alan yhtiö SS&C osti Blue Prism Group Plc:n. Blue Prism toimii nykyään virallisesti nimellä SS&C Blue Prism (Law, 2023). Blue Prism ei tarjoa eritasoisia versioita, vaan on käytössä tarjouspohjainen lisensiointi.

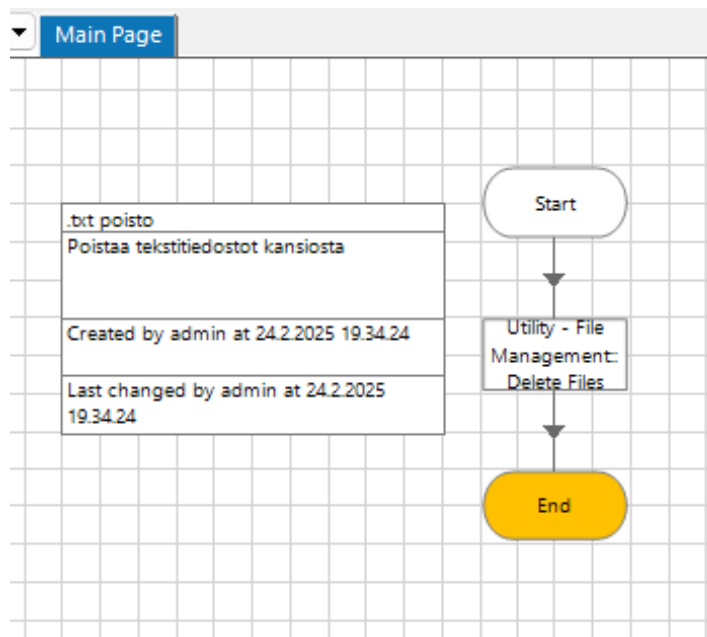
Pääpiirteittäin Blue Prism koostuu kolmesta komponentista:

- Sovelluspalvelin, joka toteuttaa Blue Prism palvelut. Toimii ikään kuin kokoonpanon aivoina. Suorittaa erilaisia funktioita, kuten yhteyden muodostamisen taustatietokantoihin, tehtävien ajoitukset ja lokitukset.
- Suoritusresurssit, jotka tunnetaan paremmin robotteina. Voivat olla joko fyysisiä tai virtuaalikoneita.
- Työpöytäsovellus, joka on paikallisesti asennettu työasemalle. Tämän kautta toteutetaan automaatioprosesseja, ajoittaa töitä tai monitoroida robotin luomia lokeja.

Nämä kolme komponenttia kommunikoivat jatkuvasti keskenään ja muodostavat yhden kokonaisuuden. (Ying, 2018)

Blue Prism ulkoasultaan on selvästi riisutumpi verrattuna edellisiin työkaluihin. Tuoreella asennuksella ei voi toteuttaa vielä prosesseja, vaan täytyy ladata ensin Business object Blue Prismin Digital Exchange sivuilta. Nämä objektit ovat kuin palapelin palasia, joita tuodaan työpöytäsovellukseen. Pystyäkseen toteuttamaan saman prosessin, kuin Power Automatella tai UiPathilla, täytyy ensin ladata File Management -objekti. Tämän jälkeen on mahdollista valita erilaisia toimintoja, kuten tarkista tiedosto, löytyykö tiedosto, poista tiedosto ja niin edelleen. Eri funktioita, kuten Action, Loop, Decision, on mahdollista käyttää ja näiden välille vedetään linkki, jotta ketju alusta loppuun onnistuu. Sivun 24 kuvassa 23 näkyy ulkoasusta kuvaa, jossa toteutettu txt-tiedostojen automatisoitu poisto. Siinä myös hahmottuu prosessin logiikka, järjestyksessä mennään Start – End ja välissä olevat prosessit suoritetaan. Toiminnaltaan samanlainen, kuin Power Automatella toteutettu prosessi, eli suoraan poistetaan kansiota txt-tiedostot.

Kuva 23. Blue Prism -editori.



5 Tarkempaa katsausta työkaluihin

Oikean työkalun valinnassa on tärkeää tiedostaa tarpeet ja vaatimukset automatisoitavalle tehtävälle, jotta työkalulla pystyy toteuttamaan automatiikan. Lähtökohtaisesti ennustettavat skenaariot työtehtävissä ovat hyviä alustoja automatiikalle. Edellä mainitut Power Automate, UiPath ja Blue Prism ovat kaikki hyviä automaatiotyökaluja, mutta valinta näiden kesken voi olla haastavaa. Näiden kolmen työkalun asentamista, käyttöönottoa ja prosessien toteuttamista katsotaan tarkemmin, sekä tehdään kattavampaa vertailua. Tässä luvussa käydään läpi tarkemmin ominaisuuksia järjestyksessä Power Automate, UiPath ja Blue Prism asennuksesta lähtien.

Valitaan lopuksi yksi näistä työkaluista, jolla pyritään ratkomaan samat ongelmat, kuin VBA-menetelmällä. Tutkitaan myös, onko mahdollista viedä jopa pidemmälle automatiikkaa hyödyntämällä robotiikkaa.

5.1 Käyttöönotto

Power Automate on Microsoftin sovellus, joten helpoiten sitä pääsee käyttämään Microsoft 365 -lisenssillä. Power Automate löytyy Microsoft 365 -sivuilta, kun hakee kaikkia sovelluksia. Tämä Microsoft 365 -versio tosin kattaa Power Automatesta vain

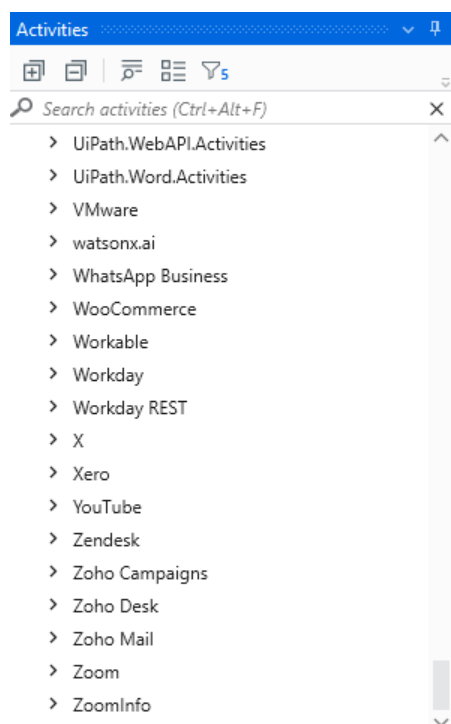
ilmaislisenssin, mutta se kuitenkin tarjoaa käyttäjille sujuvan pääsyn perustoimintoihin ja sopii yksityiskäyttöön. Laajempaan esimerkiksi yrityskäyttöön myydään erikseen Premium-tason lisenssi. (Microsoft, 2024)

Jos automatisoitavat tehtävät pyörivät Microsoftin tuotteiden ympärillä, pystytään toteuttamaan prosesseja täysin pilvipalvelun puolella. Näin ollen erillistä latausta sovelluksesta ei tarvita. Työpöydällä ajettava RPA-sovellus on ulkoasultaan vastaava, kuin UiPath tai Blue Prism. Tämä sovellus on käyttäjäystävällinen ja aloittelijallekin selkeä ohjeineen. RPA-sovelluksessa on kattavasti sovelluksia, joihin pääsee käsiksi. Näistä suurin osa ovat kuitenkin Microsoftin tuotteita, muiden julkaisijoiden tuotteita on harvakseltaan. Useat oppilaitokset kattavat saatavuuden tämän käyttöön, missä käyttöoikeustaso on Power Automate for Office 365.

UiPath tarjoaa eri tason työkaluja, on olemassa ilmainen versio, mutta tämä on toiminnallisuuksiltaan niukka. Yksityishenkilöille perusasioiden automatisointiin ilmainen versio kuitenkin riittää. Saatavilla on 60 päivän ilmainen Pro-version kokeilujakso, jossa huomattavasti laajemmat tuet ja sovellusalueet ilmaiseen verrattuna (UiPath, 2025). Rekisteröitymisen ja version valinnan jälkeen toteutetaan automaatiota joko pilvipalvelussa UiPath Automation Cloud -ympäristössä tai ladata työasemalle RPA-työkalu. Työkalun asentaminen käy helposti ja käyttöliittymä on käyttäjäystävällinen, selvät opasteet ja tekoälytoiminnot sopivat hyvin aloittelevalle käyttäjälle. Tässä työssä käytetään Pro Trial -versiota.

Sovelluksia, mihin UiPath pääsee käsiksi, on huomattavasti enemmän, kuin Power Automatesa. Eivätkä nämä rajoitu vain tietyn palveluntuottajan sovelluksiin tai palveluihin. Esimerkkinä muutama ulkoinen järjestelmä, joihin pääsee käsiksi tietyiltä osin: YouTube, Jira, Facebook, PayPal. Järjestelmät ovat helposti näkyvissä UiPath Studio Activities-näkymässä, sivun 26 kuvassa 24 näkyy listausta osasta näistä järjestelmistä.

Kuva 24. UiPath Aktiviteetit -näkyvä.



Blue Prism on maksullinen työkalu, vaikka ilmaiskokeiluja on niin täysin ilmaista versiota ei ole saatavilla. Tätä työtä varten otettu 30 päivän kokeilujakso, tämä versio sisältää kaikki ominaisuudet (Blue Prism, n.d.). Rekisteröimisen ja ilmaisen kokeilujakson valitsemisen jälkeen ladataan RPA-sovellus omalle työasemalle. Sovellukseen ei kuitenkaan pääse ennen, kuin on saatu sähköpostitse kokeilujakson lisenssin avain, joka pitää asettaa asennuksen yhteydessä. Käyttöliittymältään Blue Prism on jäljessä verrattuna kahden edellisen käyttöliittymään. Ulkoasusta on hankala hahmottaa, missä voidaan luoda prosessi. Ensimmäisellä käynnistyksellä kuitenkin tarjotaan pienimuotoinen opaste työkalun käyttöön. Opasteessa ei kuitenkaan mainita, että pitää ulkoisesta verkkosivustosta ladata moduulit, johon haluaa toteuttaa automaatiota. Toisin, kuin kahdella aiemmalla työkalulla, Blue Prismillä ei ole valmiiksi käyttöliittymällä näkymää kaikista mahdollisista ulkoisista sovelluksista, mihin pääsee käsiksi. Näitä ulkoisia sovelluksia on erittäin laaja määrä tarjolla, eli varmasti löytyy omiin tarpeisiin soveltuva moduuli.

Varsinainen prosessin luominen tapahtuu erillisessä muokkausnäkyvässä, joka muistuttaa ruudukkomaisuudellaan vihkoa, kuten sivun 24 kuvassa 23 nähdään. Tutustumisen ja

perehtymisen jälkeen prosessien luominen on kohtuullisen kätevää. Blue Prism ei ole aloittelevalla tekijällä kovin intuitiivinen, ja käyttöönotossa menee oma aikansa.

5.2 Vertailu

Kukin näistä työkaluista tarjoavat omia ominaisuuksia ja toiminnallisuuksia, jotka soveltuvat erilaisiin automaatiotilanteisiin ja -tarpeisiin. Taulukossa 1 on otettu vertailuarvoiksi helppokäyttöisyys, integraatiomahdollisuudet, sekä yhteisö ja tuki. Tieto perustuu omaan kokemukseen, sekä sovellusten virallisten sivujen antamaan tietoon. Taulukkoon sovellettu suljettua Likert-asteikkoa, jossa numerot 1-3, tarkoittaen 1. täysin eri mieltä, 2. ei samaa eikä eri mieltä ja 3. täysin samaa mieltä. Hinnoittelua ei arvioida tällä asteikolla, vaan käsitellään erikseen taulukon jälkeen.

Helppokäyttöisyyden mittarina on henkilökohtaiset näkemykset käyttöönotossa.

Integraatiomahdollisuudella tarkoitetaan kuinka moneen eri järjestelmään päästään työkalulla käsiksi. Tuki ja yhteisö -kohdassa listataan mahdollisia keskustelufoorumeita, joista saadaan tukea työntekoon tai ongelmiin.

Taulukko 1. Power Automate, UiPath, Blue Prism vertailutaulukko.

	Power Automate	UiPath	Blue Prism
Helppokäyttöinen	3	2	1
Laajat integraatiomahdollisuudet	1	2	3
Yhteisöltä saa neuvoja	3	3	2

Näiden työkalujen hinta koostuu lisensseistä, bottimääristä, sekä eri tasoista, mikäli sellainen on suoraan nähtävissä. Hinnoittelussa mainitaan käsite botti, joka tarkoittaa virtuaalityöntekijää. Botti suorittaa prosesseja.

Power Automate sisältää ilmaisversion, Premium version 14 €/kk per käyttäjä, nämä ovat tarkoitettu yksityiskäyttäjille. Process 140,40 €/kk per botti, Hosted Process 201,30 €/kk per botti, nämä ovat suunnattu yritystoimintaan. (Microsoft, n.d.-b)

UiPath sisältää ilmaisversion, Pro-version hinta alkaa yhdellä kehittäjälisenssillä \$420 /kk. Yritysversiot ovat hinnoiteltu erikseen ja tehdään tarjousperusteisesti. (UiPath, n.d.)

Blue Prism tarjoaa ilmaisen kokeilujakson. Lisenssien hinnat määritellään tarpeen mukaan ja tarjousten perusteella. Suoraa hinta-arviota ei näe sivuilta.

6 Vanhojen ongelmien uudet ratkaisut

Toteutetaan yhdellä näistä RPA-työkaluista samat työtehtävät, mitkä ratkottiin käyttäen Excel- ja VBA-ympäristöjä. Pyritään päästä mahdollisimman lähelle lopputuloksen osalta ja katsotaan mitä kaikkea lopputulokseen pääsy vaatii.

Automatisoitavat tehtävät tehdään pääosin Excel ympäristössä ja hieman otetaan myös yhteyttä oman työaseman tiedostorakenteeseen (vrt. luku 2.2, jossa käyty läpi automatisoitavia työtehtäviä, tästä jätetään pois 2. tehtävä, sillä se on vastaava tehtävän 1 kanssa). Blue Prismin ominaisuudet eivät vastaa automatisoitavia tarpeita, valittavaksi jää UiPath tai Power Automate. Power Automaten RPA-sovellus sopii tähän käyttötarkoitukseen parhaiten toiminnallisuksiensa sekä työn vaatimusten osalta, suurena tekijänä tässä on saumaton integraatio Microsoftin sovellusten kanssa.

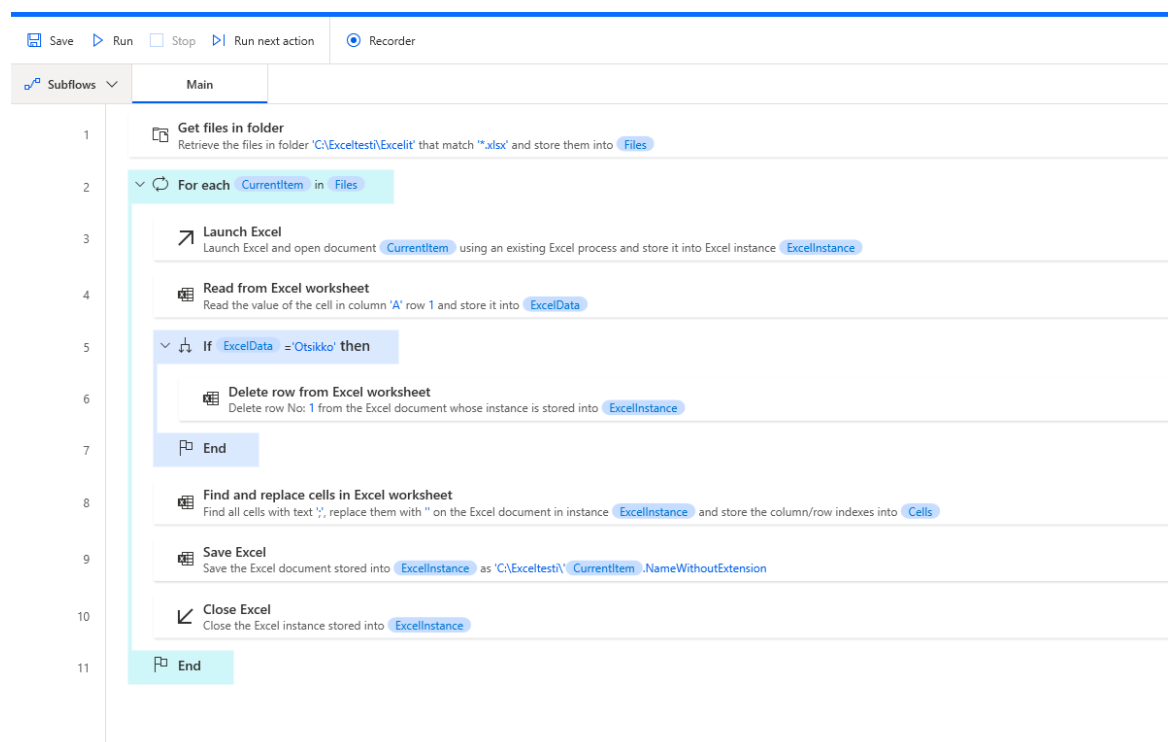
6.1 Toteutus

Ensimmäinen automatisoitava tehtävä on ottaa kansioista kaikki Excel-tiedostot ja poistaa niistä otsikkorivi, sekä kaikki puolipisteet ja tallentaa csv-tiedostona. RPA-automatiikkaa toteuttaessa täytyy ajatella koko prosessia alusta loppuun niin, kuin sen tekisi ihminen. Prosessi siis alkaa sillä, että yksitellen avataan kansioista Excel-tiedostot, tarkistetaan näistä ensimmäinen rivi A1-solusta Otsikko-arvo ja tehdään tarvittaessa poisto, sekä

tyhjäetään kaikki puolipistemerkit. Tämän jälkeen tallennetaan csv-tiedostoksi eri kansioon ja suljetaan tiedosto.

Toteutuksessa ensin tarvitaan silmukka, joka käy läpi kansion xlsx-tiedostot ja tallettaa ne Files-nimiseen kirjastoon. Silmukan sisällä jokaiselle kirjaston tietueelle, eli Excel-tiedostolle (CurrentItem), tehdään edellä mainitut toimenpiteet. Ensin avataan Excel-tiedosto, luetaan A1-solun arvo ja tallennetaan se muuttujan ExcelData nimen taakse. If-ehtolauseella tarkistetaan, onko ExcelData-muuttujalla arvona Otsikko. Mikäli on, niin poistetaan ensimmäinen rivi asiakirjasta. Puolipisteiden poistoon käytetään etsi ja korvaa -toimintoa, haetaan kaikki puolipistemerkit ja korvataan ne tyhjällä. Sitten tehdään tallennus nimellä, tiedostotyyppiä valitaan csv ja asetetaan polku. Tiedostonimen ilman alkuperäistä xlsx-päätettä saa CurrentItem.NameWithoutExtension-arvolla. Tämän jälkeen suljetaan tiedosto. Kuvassa 25 on näkymää prosessieditorista ja tästä luodusta prosessista.

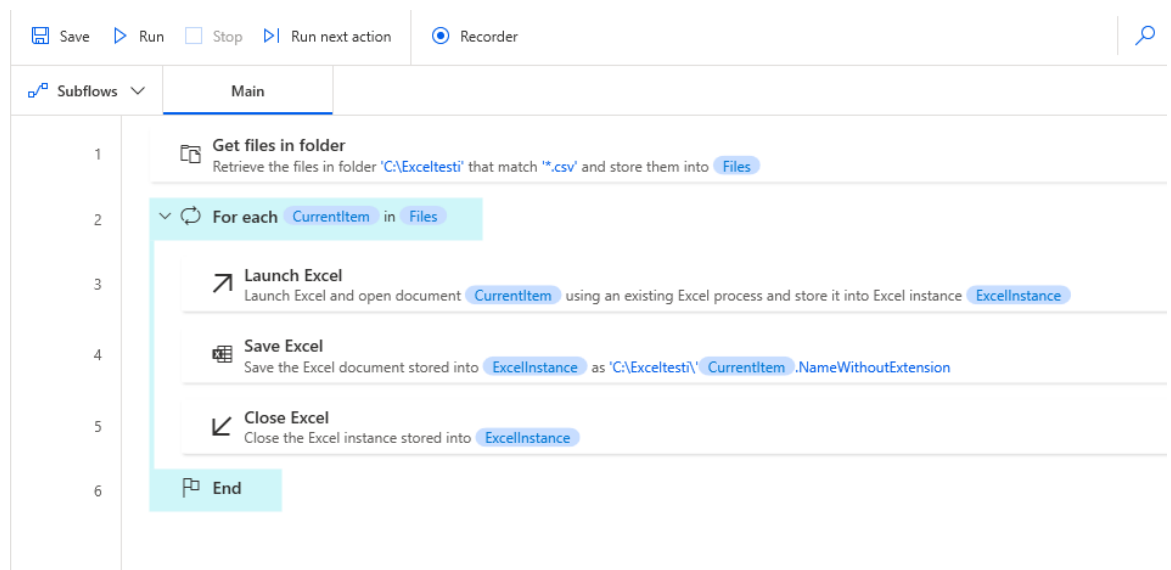
Kuva 25. Power Automate RPA Toteutus.



Prosessi suoriutuu nopeasti; alle sekunti kuluu asiakirjan korjauksessa, ja suurin aika siitä kuluu Excel-tiedoston avauksessa. Työaseman suorituskyvyllä on paljon vaikutusta tällaisen paikallisen RPA-prosessin ajamisella. Muokkaukset eivät tapahdu näkymättömästi, vaan RPA-työtä voi samalla seurata, kun Excel-tiedosto avautuu, muokkautuu, tallentuu ja sulkeutuu.

Toinen toteutettava prosessi on muuttaa csv-tiedosto xlsx-tiedostoksi. Tässä käänteisessä prosessissa ei tarvita tehdä asiakirjalle tarkistuksia. Riittää, että avataan csv-tiedosto ja tallennetaan se xlsx-muotoisena omaan kansioon. Prosessin rakentaminen on yhtä yksinkertainen ja pääpiirteinen rakenne on sama, kuin edellisessä prosessissa. Avataan csv-tiedostot kansioista ja talletetaan ne Files -kirjastoon, jotka käydään yksitellen läpi For each -silmukassa. Silmukan sisällä avataan tiedosto Excelissä, tallennetaan xlsx-tyyppisenä ja suljetaan asiakirja. Kuvassa 26 näkyy prosessi editorissa.

Kuva 26. Power Automate CSV-Exceliksi.

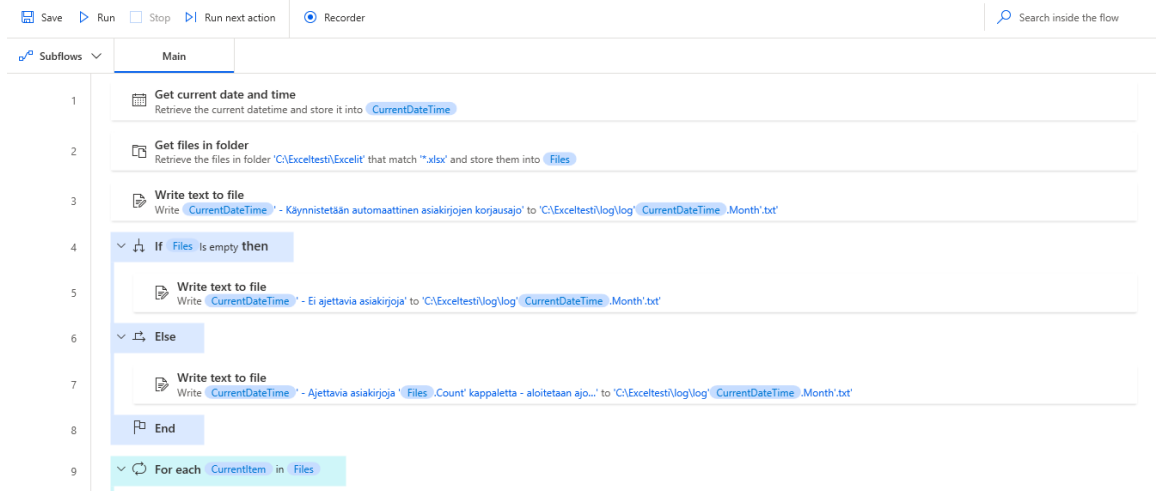


Tuotannossa automaatiota käytettäessä on tärkeää, että tehdystä työstä jää jälki, johon voi palata tarvittaessa. Tästä töiden jäljestä käytetään termiä loki (eng. log). Lokitiedostot ovat hyvinkin tärkeitä esimerkiksi ongelmatilanteiden syiden selvittelyssä. Jos edellä olevat prosessit lakkaavat toimimasta kesken ajon, olisi niiden selvitystyö hankalaa esimerkiksi ilman tietoa, minkä aineiston kohdalla ohjelma kaatuu. (Awati, n.d.)

Power Automatessa on sisäänrakennettu lokityökalu prosesseille, mutta tämä on Premium -lisenssin takana. Kiertotie kuitenkin tälle on ottaa aikaleimoja prosessien keskellä ja kirjoittaa yhteen tekstitiedostoon, mitä ollaan tekemässä ja missä vaiheessa prosessia ollaan. Tämä onnistuu käyttämällä omaa muuttujaa CurrentDateTime, johon tallennetaan nykyinen järjestelmän aikaleima ja kyseistä muuttujaa päivitetään ennen jokaista askelta. Näin saadaan todellisuutta vastaava aikaleima. Lokitusta on toteutettu ensimmäiseen prosessiin.

Prosessin alkaessa otetaan nykyinen aikaleima ja kirjoitetaan Log-nimiseen tekstitiedostoon, että prosessi alkanut. Käydään läpi kansio tiedostoista, josta edetään ehtolauseeseen, missä tarkastetaan, onko ajettavia tiedostoja vai ei. Kansion ollessa tyhjä, lokitiedostoon kirjoitetaan "Ei ajettavia asiakirjoja". Muutoin kirjoitetaan lokille ajettavien asiakirjojen määrä ja korjausajon alkaminen. Kuvassa 27 näkyy tämä osio prosessista.

Kuva 27. Power Automate -prosessin lokitus.



Tekstitiedostoon tallennettaessa tietoa, on tärkeä ottaa pois käytöstä tekstin ylikirjoitus, jotta lisätyt lokitekstit menevät omille riveilleen.

Samaa logiikkaa käytetään silmukan sisäisten tehtävien tallennukseen. Silmukan sisällä otetaan uusi aikaleima, tallennetaan lokitiedostoon avatun asiakirjan tiedostonimi ja siirrytään toimenpiteisiin. Otsikkojen poistossa otetaan uusi aikaleima ja ilmoitetaan lokille, että kyseiseltä asiakirjalta poistettu otsikkorivi. Puolipisteiden poiston osalta lokille kirjoittuu poistettujen puolipisteiden määrä. Silmukan lopussa otetaan vielä aikaleima ja kirjoitetaan lokille onnistuneesta asiakirjan korjauksesta. Tämä tehdään jokaiselle ajettavalle asiakirjalle. Silmukan loppuessa otetaan aikaleima ja kirjoitetaan lokille "Ajo suoritettu". Sivulla 32 kuvassa 28 näkyy lokitiedoston sisältöä kokonaisuudessaan.

Kuva 28. Power Automate -lokin sisältö.

```

10.3.2025 19.12.24 - Käynnistetään automaattinen asiakirjojen korjausajo
10.3.2025 19.12.24 - Ajettavia asiakirjoja 3 kappaletta - aloitetaan ajo...
10.3.2025 19.12.26 - Avattu asiakirja C:\Exceltesti\Excelit\asd.xlsx
10.3.2025 19.12.27 - Poistettu asiakirjalta C:\Exceltesti\Excelit\asd.xlsx otsikkorivi
10.3.2025 19.12.27 - Puolipisteitä poistettu asiakirjalta C:\Exceltesti\Excelit\asd.xlsx 13 kpl
10.3.2025 19.12.28 - Korjattu: C:\Exceltesti\Excelit\asd.xlsx
10.3.2025 19.12.29 - Avattu asiakirja C:\Exceltesti\Excelit\ssssa.xlsx
10.3.2025 19.12.29 - Poistettu asiakirjalta C:\Exceltesti\Excelit\ssssa.xlsx otsikkorivi
10.3.2025 19.12.29 - Puolipisteitä poistettu asiakirjalta C:\Exceltesti\Excelit\ssssa.xlsx 9 kpl
10.3.2025 19.12.30 - Korjattu: C:\Exceltesti\Excelit\ssssa.xlsx
10.3.2025 19.12.31 - Avattu asiakirja C:\Exceltesti\Excelit\ssssdddddssa.xlsx
10.3.2025 19.12.31 - Poistettu asiakirjalta C:\Exceltesti\Excelit\ssssdddddssa.xlsx otsikkorivi
10.3.2025 19.12.31 - Puolipisteitä poistettu asiakirjalta C:\Exceltesti\Excelit\ssssdddddssa.xlsx 9 kpl
10.3.2025 19.12.32 - Korjattu: C:\Exceltesti\Excelit\ssssdddddssa.xlsx
10.3.2025 19.12.32 - Ajo suoritettu

```

6.2 Jatkokehitys

Power Automatella on onnistuttu toteuttamaan vastaavat prosessit. Lopputulos on sama, kuin VBA-menetelmällä. Toteuttaminen kuitenkin kävi helpommin, kuin VBA:lla.

Näitä toteutuksia voitaisiin kehittää entistä pidemmällekin, kuten tiedostojen eteenpäin vieminen ja jaottelu nimen perusteella omiin kansioihin. Power Automatessa on mahdollista tehdä näistä prosesseista ajastettuja ajoja. Eli mikäli on tiedossa tietty kansio mihin voi tulla milloin vain aineistoja, niin nämä saadaan automaattisesti käsiteltyä.

Excelin VBA:ta sekä RPA:ta pystytään toteuttamaan yhdessäkin. Tiedetyt tehtävät suoritettaisiin makrojen avulla, joiden liipaisu tapahtuu RPA:n toimesta tietyssä kohtaa prosessia. Yksityiskohtaisemmat asiat tehtäisiin makroilla ja suuremmat eräajojen aloittamiset sekä asiakirjakohtaiset käsittelyt RPA:lla. Tämä voisi parantaa suorituskykyä, joka on tärkeää isojen aineistomäärien käsittelyissä.

Esimerkkinä otetaan Power Automatella käsittelyyn Excel-asiakirja ja automatisoidusti ajetaan VBA:lla tehty makro. Power Automatessa on sisäänrakennettuna toiminnallisuus laukaista Excel-makroja prosessin keskellä. VBA:lla voi olla hankalaa – vaikkakin mahdollista – ottaa käsittelyyn useita asiakirjoja kansioista. Tämän haasteen voi korvata RPA:lla, vaikka muut asiakirjan muokkaukset tapahtuisivatkin makromaailmassa. Näin ollen nämä kaksi työkalua helpottavat toisiaan, RPA:lla voidaan helpottaa VBA-ohjelmointia ja makroilla vähennetään RPA:n tekemiä askeleita. Tämän takia suorituskyky ja tehokkuus nousisivat. Toisin sanoen jatkokehitystä on mahdollista toteuttaa joustavasti tarpeen mukaan.

7 Pohdintaa

Automaatio helpottaa itseään toistavien tehtävien tekemisessä ja näin ollen jättää aikaa tärkeämmille töille. IT-töissä tulee väistämättä vastaan tehtäviä, jotka olisi parempi olla tekemättä käsin. Näitä havaittaessa olisi hyvä pohtia automaation toteuttamisen mahdollisuutta ja toisiko se helpotusta se omiin tai muiden töihin. On hyvä kuitenkin huomioida, että jokaista työtehtävää ei ole suotavaa tai edes mahdollistakaan automatisoida. Ainoastaan itseään toistavia ja ennustettavia töitä saadaan tehtyä robotiikalla. Nämä erilaiset työkalut ovat työntekijän tukena ja tuottavuuden parantajia.

Tämä opinnäytetyö käsitteli Excel-ympäristössä työskentelyä ja sen automaatiota. Sen myötä hahmottui, että Excelin automatisointi onnistuu monella eri tapaa. Excelin omilla makroilla ja VBA:lla päästään pitkälle, mutta vaatii harjoittelua. Toinen vaihtoehto on hyödyntää automaatiotyökaluja. Työssä tutkittiin kolmea erilaista työkalua, Power Automate, UiPath ja Blue Prism. Tarkoituksena oli toteuttaa yhdellä näistä työkaluista prosessi, joka vastaa alkuperäistä VBA-ympäristössä toteutettuja Excel-makroja.

On myös hyvä huomioida, että työkalujen tutkimusten tulokset eivät välttämättä ole luotettavia perustuen kapeaan testausmäärään. Niitä on testattu materiaalilla, joka ei perustu pitkälliseen aikaväliin tai monipuoliseen sisältöön. Saadut tulokset sopivat vain tämän työn kontekstiin, eikä vastaa yleistä sovellettavuutta. Näitä työkalujen prosesseja ei olla oikeissa työympäristöissä toteutettu, joten ei voida verrata yksi yhteen todellisten työprojektien kanssa.

Näistä kolmesta Power Automate osoittautui parhaaksi valinnaksi Excelin automatisointiin. Power Automatella päästiin melkein jokaiseen Excelin ominaisuuteen käsiksi, sekä työaseman hakemistorakenteisiin kiinni pääseminen oli tärkeä osa automaatioprosessien luomista. Power Automate oli suurilta osin helppokäyttöisempi ja aloittelijaystävällisempi, kuin VBA-ohjelmointi. Suorituskyvyiltäänkin Power Automate pärjää suurissa aineistomäärissä.

UiPath ja Blue Prism olivat hieman ulkopuolella tämän automatisoitavan ongelman aiheesta, vaikka niilläkin olisi teoriassa mahdollista tehdä samat prosessit kuin Power Automatella. Kuitenkaan suorituskyvyiltään eivät pääse samoihin tasoihin Excel-asiakirjojen muokkauksissa verrattuna Power Automateen tai VBA:han. Nämä soveltuvat eri ympäristöihin paremmin.

Power Automatella saavutettu lopputulos vastasi odotuksia täysin. Lokitustoiminnolla saatiin myös laajennettua toteutusta pidemmälle. Laajennusmahdollisuuksia on mittavasti ja näitäkin työkaluja koko ajan kehitetään toiminnallisuuksiltaan. Opinnäytetyössä esiteltyjen työkalujen lisäksi löytyy lukuisia muita vaihtoehtoja. Automatiikkatyökalut kehittyvät vauhdilla ja on todennäköistä, että tulevaisuudessa tulee vielä monipuolisempia ja älykkäämpiä ratkaisuja. Varsinkin tekoälyjen nopea kehitys tukee automaatiota, tätä osapuolta varmasti tullaan paremmin hyödyntämään automaatiotyökaluissa. RPA:n, tekoälyn ja koneoppimisen yhdistelmää kutsutaan älykkääksi automaatioksi, tämä on automaation seuraava askel.

Lähteet

- Awati, R. (n.d.) Log file. *TechTarget*. <https://www.techtarget.com/whatis/definition/log-log-file>
- Badkar, A. (17.9.2024). Key Advantages and Disadvantages of Automation. *Simplilearn*. <https://www.simplilearn.com/advantages-and-disadvantages-of-automation-article>
- Baker, P. (2024). *Generative AI*. <https://learning.oreilly.com/library/view/generative-ai-for/9781394270743/>. For Dummies
- Blue Prism. (n.d.). SS&C Blue Prism Free Trial. Haettu 1.3.2025 osoitteesta <https://www.blueprism.com/free-trial/>
- Bommadevara, N., Jansen, S., Klak, L., Subherwal, M. (8.8.2019). Flip the ratio: Taking IT from bottleneck to battle ready. <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/flip-the-ratio-taking-it-from-bottleneck-to-battle-ready#/>
- CGI. (16.5.2024). CGI OMNI360 – Tekoälytoiminnallisuudet [video]. YouTube. <https://www.youtube.com/watch?v=buH14hentE8>
- Christian P. Janssen, Stella F. Donker, Duncan P. Brumby, Andrew L. Kun. (2019). History and future of human-automation interaction, *International Journal of Human-Computer Studies*, (ss. 99-107).
- Gheorghe, G. (9.4.2018). The story of UiPath – How did it become Romania’s first unicorn? *Business Review*. <https://business-review.eu/news/the-story-of-uipath-how-it-became-romanias-first-unicorn-164248>
- Google. (2025). Gemini 2.0 Flash. [Laaja kielimalli]. Haettu 9.3.2025 osoitteesta <https://gemini.google.com/>
- IBM. (6.7.2023-a) What is IT automation? <https://www.ibm.com/think/topics/it-automation>
- IBM. (6.7.2023-b) IT automation tools. <https://www.ibm.com/think/topics/it-automation>
- IBM. (n.d.) What is cloud automation? <https://www.ibm.com/think/topics/cloud-automation>
- Jovanović, S. Z., Đurić, J. S., & Šibalija, T. V. (2019). *Robotic process automation: overview and opportunities*. *International Journal Advanced Quality*, (ss. 34–39).
- Jyväskylän yliopisto. (3.2.2005). VBA-perusteet erittäin lyhyesti <https://appro.mit.jyu.fi/doc/tiedonhallinta/vba/>
- Kappagantula, S. (2.6.2023). What is UiPath? – A Quick Guide To The Top RPA Tool. *Edureka*. <https://www.edureka.co/blog/what-is-uipath/>
- Kusleika, D. (2022). Excel VBA Programming. <https://learning.oreilly.com/library/view/excel-vba-programming/9781119843078/>. For Dummies
- Law, M. (16.8.2023). SS&C Blue Prism: A leader in the world of RPA. *Technology Magazine*. <https://technologymagazine.com/articles/ss-c-blue-prism-a-leader-in-the-world-of-rpa>

- Lee, I. (24.7.2024). 9 benefits of an automation-first mindset. *Red Hat*.
<https://www.redhat.com/en/blog/9-benefits-automation-first-mindset>
- Microsoft. (30.8.2023). Data type summary <https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/data-type-summary>
- Microsoft. (13.12.2024). Types of Power Automate licenses <https://learn.microsoft.com/en-us/power-platform/admin/power-automate-licensing/types?tabs=power-automate-free%2Cpower-automate-process%2Cconnector-types>
- Microsoft. (n.d.-a) Power Automate. Haettu 15.2.2025 osoitteesta <https://www.microsoft.com/fi-FI/power-platform/products/power-automate>
- Microsoft. (n.d.-b). Power Automaten hinnoittelu. Haettu 11.4.2025 osoitteesta <https://www.microsoft.com/fi-fi/power-platform/products/power-automate/pricing>
- Montecino, V. (2020) History of Computing. <https://mason.gmu.edu/~montecin/computer-hist-web.htm>
- Save My Leads. (13.7.2024). Power Automate Vs UiPath Vs Blue Prism. *Savemyleads*.
<https://savemyleads.com/blog/other/power-automate-vs-uipath-vs-blue-prism>
- Şimşek, H. (17.12.2024). Top 60+ IT Automation Use Cases in 2025. <https://research.aimultiple.com/it-automation-use-cases/>
- Stork, P.P, (2023) Learning Microsoft Power Automate. <https://learning.oreilly.com/library/view/learning-microsoft-power/9781098136352/>. O'Reilly Media, Inc.
- Suomen Koodikoulu. (2019). *Johdatus tekoälyyn*. (ss. 3–10)
- UiPath (19.2.2025) Automation Cloud admin guide. Haettu 1.3.2025 osoitteesta <https://docs.uipath.com/automation-cloud/automation-cloud/latest/admin-guide/about-licensing#free>
- UiPath (n.d.) Plans and Pricing. Haettu 11.4.2025 osoitteesta <https://www.uipath.com/pricing>
- Veenendaal, A. (n.d.) Common Robotic Process Automation (RPA) Use Cases.
<https://www.blueprism.com/automation-journey/robotic-process-automation-rpa-use-cases/>
- Ying, L.M, (2018). Robotic Process Automation with Blue Prism Quick Start Guide
<https://learning.oreilly.com/library/view/robotic-process-automation/9781789610444/>. Packt Publishing