



Sami Tenkanen

Tilaustenkäsittelyn prosessin kehitys ohjelmistorobotiikan avulla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tuotantotalouden tutkinto-ohjelma

Opinnäytetyö

5.5.2022

Tiivistelmä

Tekijä:	Sami Tenkanen
Otsikko:	Tilaustenkäsittelyn prosessin kehitys ohjelmistorobotiikan avulla
Sivumäärä:	74 sivua + 1 liite
Aika:	5.5.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tuotantotalouden tutkinto-ohjelma
Ammatillinen pääaine:	ICT-Liiketoiminnan johtaminen
Ohjaajat:	Jarno Liikanen, Kehityspäällikkö, Hakonen Solutions Oy Nina Hellman, Lehtori

Insinöörityön tarkoituksena oli implementoida ohjelmistorobotti logistiikka-alan yrityksen käyttöön. Ohjelmistorobotin avulla oli tarkoitus kehittää ja automatisoida yrityksen tilaustenkäsittelyprosessia.

Yrityksen tilaustenkäsittelyprosessissa tietoa täytyi siirtää yrityksen kumppanien ja yrityksen oman järjestelmän välillä. Järjestelmien välille ei ollut mahdollista suorittaa tavanomaista integraatiota tietojen siirtämiseksi.

Ohjelmistorobotin käyttöönottoa varten prosessin nykytila analysoitiin. Analyysin aikana havaittiin prosessissa haasteita ja ongelmia, jotka oli ratkaistava ennen robotin käyttöönottoa. Prosessin kehitystä ja ohjelmistorobotin käyttöönottoa varten selvitettiin parhaita käytänteitä ja työkaluja. Prosessin kehityksen osalta tutustuttiin varsinkin Lean-ajattelumalliin. Myös Agile-metodologiaan ja sen soveltumiseen ohjelmistorobotin kehittämiseen tutkittiin.

Robotin käyttöönoton jälkeen selvitettiin projektin vaikutukset tilaustenkäsittelyprosessiin ja selvitettiin projektin ja ohjelmistorobotin käyttöönoton kannattavuutta. Lisäksi selvitettiin tilaustenkäsittelyprosessiin osallistuvien työntekijöiden mielipiteitä projektista ja automatisaatiosta.

Avainsanat: Ohjelmistorobotiikka, RPA, prosessin kehitys

Abstract

Author: Sami Tenkanen
Title: Development of Order Entry Process With RPA
Number of Pages: 74 pages + 1 appendices
Date: 5 May 2022

Degree: Bachelor of Engineering
Degree Programme: Industrial Management and Engineering
Professional Major: IT Business
Supervisors: Jarno Liikanen, Development manager, Hakonen Solutions Oy
Nina Hellman, Senior Lecturer

The purpose of this thesis was to implement an RPA solution for Hakonen Solutions Oy. The purpose of the RPA implementation was to develop and automate an order entry process for the company.

The process included moving information between the company's and its partners' systems. There was no possibility to create a traditional software integration between the systems.

The current state of the process was first analyzed for the implementation of the RPA solution. During the analysis some problems and challenges were identified in the process that had to be addressed before implementing the RPA Solution. Information about best practices in RPA and process development was gathered to support the project. Lean thinking was especially explored for the sake of process development. Also, Agile methodologies and their usability in RPA development were studied.

The impact of the project on the automated process, and the profitability of the project and RPA solution were analyzed after the RPA deployment. Furthermore, a survey was sent to the employees related to the automated process in which they were asked about their opinions about the project and automation in the company, and the answers were analyzed.

Keywords: RPA, Process development

Sisällys

1	Johdanto	1
2	Menetelmät ja materiaali	4
2.1	Tutkimussuunnitelma	4
2.2	Tiedonkeruu	6
3	Nykytila-analyysi	8
3.1	Nykytilan yleiskuva	8
3.2	Järjestelmät	10
3.3	Analyysi nykytilasta	12
3.4	Nykytila automatisaation näkökulmasta	13
3.5	Nykytilan mittarit	18
3.5.1	Prosessien käsittelyajat	19
3.5.2	Prosessien vasteajat	21
3.5.3	Tilauksien virhemäärät	24
3.6	Yhteenveto nykytilasta	25
4	Parhaat käytänteet	27
4.1	Ohjelmistorobotiikan parhaat käytänteet	27
4.1.1	Ohjelmistorobotiikka lyhyesti	28
4.1.2	Ohjelmistorobotiikan suunnittelu	29
4.1.3	Ohjelmistorobotin kehitys	31
4.1.4	Ohjelmistorobotin tuotantoon vienti ja ylläpito	33
4.2	Prosessien kehitys	34
4.2.1	Lean IT	35
4.2.2	Lean-työkalut	39
4.3	Agile-sovelluskehitys	40
4.4	Yhteenveto parhaista käytänteistä	44
5	Ohjelmistorobotin käyttöönotto ja kehitys	47
5.1	Prosessien optimointi	47
5.1.1	Työvaiheiden standardisointi	47
5.1.2	Automatisaatioon liittyvien haasteiden ratkaisu	49
5.1.3	Yhteenveto prosessinkehityksestä	51

5.2	Ohjelmistorobotin käyttöönotto ja kehitys	52
5.2.1	Ohjelmistorobotin käyttöönotto	52
5.2.2	Ohjelmistorobotin kehitys	54
6	Tulosten arviointi ja jatkokehitys	64
6.1	Projektin vaikutus mittareihin	64
6.2	Projektin kannattavuus ja ROI	68
6.3	Työntekijöiden kokemukset	69
6.4	Ohjelmistorobotin tulevaisuus yrityksen toiminnassa	70
7	Yhteenveto	72
	Lähteet	74
	Liitteet	
	Liite 1: Työntekijöiden kokemukset projektin lopuksi	

Lyhenteet

ERP-Järjestelmä: *Enterprise Resource Planning*. Toiminnanohjausjärjestelmä, jonka kautta yritys hallitsee eri liiketoimintatietoja, kuten tilauksia, varastoa ja tuotantoa.

Sherpa: Opinnäytetyön kohdeyrityksen käyttämä toiminnanohjausjärjestelmä.

ROI: *Return of Investment*. Sijoitetun pääoman tuottoaste. Ilmoittaa kuinka paljon johonkin kohteeseen sijoitettu pääoma on tuottanut.

RPA: *Robotic Process Automation*. Ohjelmistorobotiikka.

ROM: *Robotic Operating Model*. Blue Prismin kehittämä viitekehys, joka käsittelee ohjelmistorobotiikan hyödyntämistä liiketoiminnassa.

PDCA: *Plan Do Check Act*. Muun muassa Lean-ajattelussa hyödynnetty kehityssykli.

1 Johdanto

Digitalisaation myötä datan käsittely on nopeutunut ja helpottunut huomattavasti. Sähköisten järjestelmien ja tietokantojen kehityksestä huolimatta toimistoihin on jäänyt paljon manuaalista sekä rutiininomaista työtä. Paljon dataa joudutaan syöttämään, siirtelemään ja käsittelemään manuaalisesti. Yrityksien tahtotilana on minimoida tällainen rutiininomainen työ, jotta työntekijät pystyisivät keskittymään kehittävään ajatteluun vaativaan työhön. Työtehtäviä on pyritty automatisoimaan erilaisilla ohjelmistoilla, sovelluksilla ja järjestelmäintegraatioilla. Ohjelmistojen ja integraatioiden kehitys on usein paitsi kallista sekä aikaa vievää. Tästä syystä uuden ohjelmiston kehittäminen yksittäistä työtehtävää varten, tai integraation rakentaminen pieneen järjestelmään ei usein ole kannattavaa. Tällaisissa tilanteissa vaihtoehtona automaatiolle voisi olla esimerkiksi manuaalista työtä jäljittelevä ohjelmistorobotti.

Tämän opinnäytetyön tarkoituksena on implementoida ohjelmistorobotti yrityksen käyttöön, suunnitella ja kehittää prosessi ohjelmistorobotin suoritettavaksi sekä pohtia robotin tulevaisuutta yrityksen liiketoimintaprosessien automatisoinnissa.

Hakonen Solutions Oy on palvelevan logistiikan perheyritys. Yritys on perustettu vuonna 1918, mistä lähtien yritykselle on kertynyt laajasti kokemusta erilaisten kuljetuspalveluiden toteuttamisesta. Opinnäytetyön tekemisen aikaan yrityksen ydinliiketoiminta perustuu huonekalujen toimituksiin sekä kodinkoneiden myyntiin, toimituksiin ja huoltoihin.

Yrityksen asiakaskunta koostuu pääosin huonekaluja tai kodinkoneita myyvistä yrityksistä sekä taloyhtiöistä ja vuokranantajista. Jokaisella yrityksellä on yleensä omat järjestelmänsä myyntiä, tilauksia, toimituksia tai huoltopyyntöjä varten. Tavallisesti joko asiakkaat tai loppuasiakkaat tekevät tilauksen asiakkaan järjestelmään, josta tilaus ja sen tiedot täytyy saada siirrettyä kuljetusyrityksen omaan järjestelmään. Joissain tapauksissa näiden vaiheiden välissä voi

olla vielä lisäksi kolmannen osapuolen hallinnoima järjestelmä. Mikäli tilausvolyymi on suuri, tavallisesti järjestelmien välille tehdään integraatio, jotta tilaukset ja niiden tiedot saadaan siirtymään automaattisesti järjestelmien välillä. Aina integraation tekeminen ei kuitenkaan ole mahdollista eikä kannattavaa. Asiakkaan käyttämä järjestelmä ei välttämättä tue integraatioon tarvittavaa tekniikkaa, tai tilausvolyymi on niin pieni verrattuna integraation kehittämisen aiheuttamiin kustannuksiin, ettei asiakas tai kuljetusyritys halua investoida integraatioon. Tällaisissa tapauksissa tilauksien tiedot on siirrettävä manuaalisesti asiakkaan järjestelmistä kuljetusyrityksen omaan järjestelmään. Lisäksi tilaukseen liittyvät päivitykset on tehtävä molempiin järjestelmiin erikseen. Tilauksien käsittelystä on vastuussa sopimuksesta riippuen asiakkaan tai kuljetusyrityksen edustaja. Tässä työssä keskitytään kuljetusyrityksen vastuulla olevaan tilauksenkäsitelyyn.

Vaikka yksittäinen järjestelmäintegraatio ei olisikaan laskennallisesti kannattavaa, asiakaskunnan ja eri järjestelmien määrän kasvaessa tilausten käsittelyn aiheuttama työmäärä ja haasteet kasvavat. Jokainen uusi järjestelmä vaatii työntekijän kouluttamisen kyseisen järjestelmän käyttämiseen. Eri asiakkaiden ja järjestelmien välillä voi myös olla eroavaisuuksia siinä, missä muodossa ja mitä tietoja tilauksilla tulee siirtää. Tämä aiheuttaa suurempaa riskiä virheille tilauksenkäsitelyssä. Myös tilausten syötössä datan yhteneväisyys voi kärsiä, sillä toisaalta eri järjestelmät voivat esittää samaa tietoa keskenään eri lailla. Myös eri tilausten käsittelijät voivat herkästi käsitellä tilauksia eri tavoin, mikäli menetelmiä ei ole tarkasti määritetty.

Työn päämääränä on implementoida ohjelmistorobotti suorittamaan tilausten käsittely työssä määriteltyjen järjestelmien välille. Lopputulema on itsenäisesti työskentelevä ohjelmistorobotti, joka hakee tilaukset asiakkaiden järjestelmistä ja syöttää niiden tiedot kuljetusyrityksen omaan järjestelmään. Lisäksi robotti käy tarvittaessa saattamassa ajan tasalle tilauksiin liittyviä päivityksiä asiakkaiden järjestelmiin.

Työ keskittyy kolmesta eri asiakasjärjestelmästä tulevien tilausten käsittelyyn. Eri järjestelmistä tulevien tilausten käsittely jaetaan kolmeen erilliseen prosessiin, jotka tullaan siirtämään ohjelmistorobotin käsiteltäviksi. Työn lopussa arvioidaan työn lopputulos ja sen vaikutukset. Lisäksi pohditaan ohjelmistorobotin tulevaisuutta yrityksen liiketoiminnassa.

2 Menetelmät ja materiaali

Tässä luvussa kuvataan opinnäytetyön tutkimussuunnitelma, käytetyt menetelmät sekä tiedonkeruuaineistot. Luvussa käydään myös lyhyesti läpi, miksi kyseiset menetelmät työlle valittiin.

Ohjelmistorobotti pyrkii jäljittelemään oikean työntekijän työskentelyä. Robotti käyttää järjestelmiä pitkälti kuten oikea työntekijä klikkaamalla näytöllä näkyviä painikkeita ja kirjoittamalla niihin tekstiä. Robottia varten tarvittiin siis tarkka kuvaus nykytilasta ja yksityiskohtainen tieto prosessin vaiheista.

2.1 Tutkimussuunnitelma

Työ koostuu viidestä vaiheesta, jotka on kuvattu ja visualisoitu lyhyesti kuvassa 1.



Kuva 1 Tutkimussuunnitelma

Ensimmäisessä vaiheessa kuvattiin liiketoiminnan haaste ja määritettiin työn tavoite. Seuraavassa vaiheessa selvitettiin prosessien nykytila ja määritettiin senhetkiset prosessit. Tavoitteena nykytila-analyysille oli saada tarkka kuvaus ja ymmärrys prosessien vaiheista, sekä niiden soveltuvuudesta automatisoitaviksi. Nykytila-analyysin aikana kehitettiin mittarit nykytilan ja projektin lopputuloksen mittaamiseksi.

Kolmannessa vaiheessa selvitettiin robotin implementointiin sekä prosessien kehitykseen liittyviä parhaita käytänteitä. Vaiheen pohjalta syntyi työn kirjallinen viitekehys.

Neljännessä vaiheessa keskityttiin prosessin kehittämiseen automaatiota varten sekä robotin käyttöönottoon, kehitykseen ja tuotantoon viemiseen. Viimeisessä vaiheessa arvioitiin projektin lopputulos ja selvitettiin ohjelmistorobotilla toteutetun automatisaation hyödyt. Samalla tehtiin myös ehdotus robotin jatkokehitysmahdollisuuksista yrityksen liiketoiminnassa.

2.2 Tiedonkeruu

Nykytila-analyysiä varten oli kerättävä aineistoa yrityksen henkilökunnalta. Aineistoa kerättiin vapaamuotoisen kysymysten sekä tapaamisten avulla. Ohjelmistorobotin rakentamista varten oli tärkeää saada yksityiskohtainen kuva tilausten käsittelyn prosesseista ja työvaiheista. Tätä varten tilausten käsittelijöitä pyydettiin tapaamisissa esittelemään työvaiheet kuten perehdyttäisivät uuden henkilön työtehtäviin. Aineistoa kerättiin myös palautekyselyn kautta projektin lopuksi. Kerätyt aineistot on esitetty taulukossa 1.

Taulukko 1 Työn aikana kerätyt aineistot

Osallistujat	Menetelmä	Aihe	Pvm	Dokumentoitu
Aineisto 1, Nykytila-analyysi				

Palvelupäällikkö	Tapaaminen paikan päällä	Nykytilan analysointi, Prosessien kokonaiskuva ja osalliset, Kumppanien järjestelmät	21.2.2022	MS Word
Tilaustenkäsittelijä 1	Tapaaminen paikan päällä	Nykytilan analysointi, Tilausten käsittely ja järjestelmät	23.2.2022	MS Word
Tilaustenkäsittelijä 2	MS Teams	Nykytilan analysointi, Tilausten käsittely Kodinkoneasennus ja järjestelmät	25.2.2022	MS Word
Laskuttaja, tilaustenkäsittelijä 3	MS Teams	Nykytilan analysointi, Tilausten käsittely Huolto, Tilausten kuittaus	1.3.2022	MS Word
Aineisto 2, Palaute				
Tilaustenkäsittelijät 1–3, ja laskuttaja	Ms Teams, sähköposti	Palautteen kerääminen	19.4.2022	MS Word

3 Nykytila-analyysi

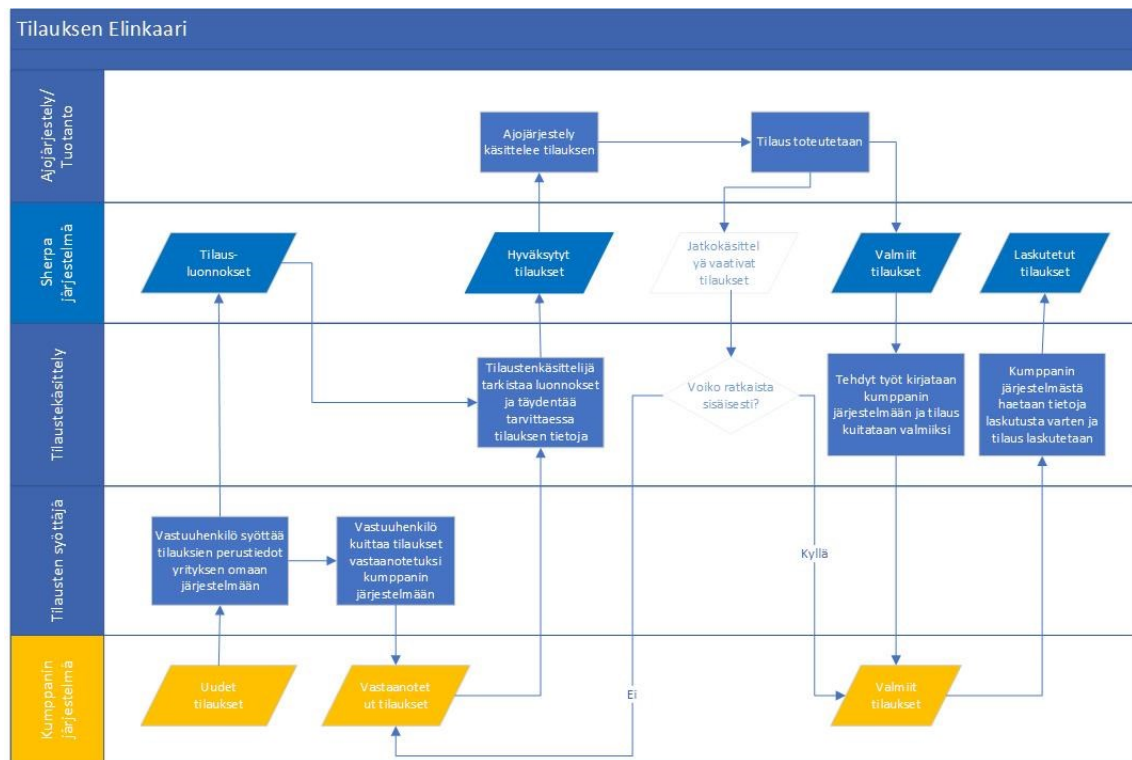
Tämän osion tarkoituksena on selvittää automatisoitavien prosessien nykytila, ja kuvata prosessit niiden nyky muodossa. Lisäksi osiossa luodaan mittarit nykytilaa ja automatisaation onnistumisen tarkastelua varten.

Nykytila-analyysia varten järjestettiin tapaamisia prosessien osalta merkityksellisten henkilöiden kanssa. Aluksi järjestettiin tapaaminen palvelupäällikön kanssa, jossa selvitettiin tilauksen elinkaaren sekä prosessien yleiskuva. Tapaamisessa selvitettiin myös prosessiin osallistuvat työntekijät tulevia tapaamisia varten. Tämän jälkeen järjestettiin tapaamiset prosessiin osallistuvien työntekijöiden kanssa. Tapaamisissa pyydettiin työntekijöitä kuvaamaan prosessien työtehtävät kuin he perehdyttäisivät uutta työntekijää kyseisiin työtehtäviin. Lisäksi työntekijöiltä tiedusteltiin heidän näkemystänsä prosessien haasteista.

3.1 Nykytilan yleiskuva

Hakonen Solutions Oy on jaettu liiketoimintojen mukaisiin liiketoiminta-alueisiin, jotka edelleen jakautuvat läheisesti yhteistyössä toimiviin osastoihin. Tarkastelussa olevat prosessit kuuluivat yrityksen Elinkaaripalvelut liiketoiminta-alueeseen. Liiketoiminta-alueen alla prosesseihin osallistui kaksi osastoa Huolto sekä Kodinkoneasennus.

Nykytilan kartoitusta varten järjestettyjen tapaamisten pohjalta tehtiin kuvaus tilauksen elinkaaresta nykytilassa. Tilauksen elinkaari on esitetty kuvassa 2. Elinkaareen tietovirrat olivat pääpiirteittäin samanlaiset riippumatta siitä, minkä kumppanin järjestelmästä tilaus oli luotu tai mikä osasto tilauksen käsittelyn suoritti.



Kuva 2 Nykytila-analyysin pohjalta kuvattu tilauksen elinkaari

Kuvan mukaisesti tilauksen elinkaari alkaa, kun tilaaja luo tilauksen kumppanin järjestelmään. Seuraavaksi tilauksen tiedot syötetään yrityksen omaan ERP-järjestelmään (Sherpa). Uusien tilausten haku ja syöttäminen suoritettiin massajona niin, että kaikki uudet tilaukset syötettiin samanaikaisesti Sherpaan. Käytännössä kumppanin järjestelmästä ladattiin tilauksien data yhtenä .csv-tiedostona, jossa oli tilauksien tärkeimmät tiedot. Saatu tiedosto siirrettiin Excelillä luotuun työkaluun, joka muunsi datan yrityksen Sherpa-järjestelmän hyväksymään muotoon. Samalla data myös tarkistettiin silmämääräisesti räikeiltä virheiltiltä sekä luokiteltiin tyypin ja käsittelevän osaston mukaan. Työkalun avulla saatiin yksi Excel-tiedosto, joka ladattiin suoraan yrityksen Sherpa-järjestelmään. Lopuksi tilauksen syöttäjä merkkasi tilaukset vastaanotetuiksi kumppanin järjestelmiin. Jokaisesta kolmesta kumppanin järjestelmästä saatavassa .csv-tiedostossa data oli keskenään erilaisissa muodoissa. Näin ollen jokaiselle järjestelmälle vaadittiin oma Excel-työkalu kyseisestä työvaihetta varten. Työvaihe suoritettiin yleensä 6 kertaa päivässä, eli kahdesti jokaisen järjestelmän osalta.

Seuraavassa työvaiheessa tilausten käsittelijä tarkasti luonnoksen. Tarkastuksessa katsottiin, että tilaajan sekä loppuasiakkaan tiedot löytyivät ja olivat oikein, ja tilauksessa oli riittävä työnkuvaus tilauksen suorittamista varten. Mikäli tarvittavaa tietoa puuttui, jouduttiin se hakemaan kumppanin järjestelmästä. Seuraavaksi tilaukselle merkattiin myytävät tuotteet, mikäli niitä sisältyi tilaukselle. Lisäksi tilaukselle tehtiin vielä luokituksia ajojärjestelyä ja erilaisia analyyssejä varten. Työvaiheen lopputulemana tilauksen tila muutettiin tilausluonnoksesta hyväksytyksi tilaukseksi. Yhden tilauksen käsittelyyn kuluva aika vaihteli käsittelijän, tilauksen tyyppin sekä korjausta vaativien tietueiden mukaan. Tapaa- misten aikana tehtyjen mittausten perusteella. Yhden tilauksen käsittelyajaksi voitiin arvioida keskimäärin 5 minuuttia.

Hyväksytyt tilaukset siirtyivät ajojärjestelyn käsiteltäviksi ja sitä kautta tuotantoon. Kun tilaus saavutettiin valmiiksi, käy tilauksen käsittelijä kirjaamassa tehdyt työt ja mahdollisesti myydyt tavarat kumppanin järjestelmään ja kuittaamassa tilauksen valmiiksi. Lopuksi tilaus laskutetaan. Yhden kumppanin järjestelmän osalta tilaus siirtyi kuittauksen jälkeen vielä asiatarkestettavaksi kumppanille. Asiatarkastuksen jälkeen tilaukselle muodostui ostotilausnumero, joka laskuttajan täytyi hakea kumppanin järjestelmästä ja merkitä laskutettavalle tilaukselle ennen laskutusta. Viimeiseen työvaiheeseen kului noin 1 minuutti tilausta kohden, pois lukien ostotilausnumeron hakeminen kumppanin järjestelmästä. Ostotilausten hakeminen tehtiin ensimmäisen työvaiheen kaltaisella massa- ajolla kerran päivässä. Ajoon kului noin 10 minuuttia.

Yhteensä kaikista kolmesta järjestelmästä tulee sesongin mukaan noin 1800–3000 tilausta kuukaudessa. Manuaalinen tilausten käsittely vaati siis hiljaisina kausina hieman vajaan yhden henkilötyöpäivän, ja kiireisinä kausina lähes 2 henkilötyöpäivää kuukaudessa.

3.2 Järjestelmät

Tilauksia haettiin kaikkiaan kolmesta erillisestä järjestelmästä. Kaikilla kolmella järjestelmällä oli selainpohjainen käyttöliittymä, johon tilausten käsittelijöillä oli

pääsy. Kaksi järjestelmää oli yksittäisten kumppanien omia järjestelmiä, jotka kattoivat vain kyseisen kumppanin tilauksia. Nämä kaksi kumppania ja järjestelmää kattoivat noin 75 prosenttia tilausmäärästä. Kolmas järjestelmä oli käytössä useammalla pienellä asiakkaalla, ja sisälsi näin ollen tilauksia monelta eri asiakkaalta.

Kaikki kumppanijärjestelmät olivat toimintaperiaatteeltaan hyvin samankaltaisia. Kaikissa järjestelmissä oli mahdollista saada listaus tilauksista avoimista, keskeneneräisistä ja valmiista toimituksista. Samoin myös jokaisessa järjestelmässä oli tilaus avattava tarkasteluun, jotta kaikki halutut tietueet saatiin näkyviin. Järjestelmien välillä oli kuitenkin eroa siinä, miten tietueet olivat nimettyjä ja missä muodossa data esitettiin. Yhdessä järjestelmässä oli joitakin tilauksen tietoja haettava erinäisten linkkien kautta.

Tilaukset syötettiin yrityksen omaan ERP-Järjestelmään (Sherpaan). Sherpa-järjestelmä kattaa yrityksessä koko tilauksen elinkaareen tilausten luonnista aina laskutukseen. Järjestelmän kautta tapahtuu myös ajojärjestely, henkilöstöresursointi ja varastonhallinta. Kuljettajia ja asentajia varten järjestelmä toimii myös mobiililaitteella, jolla kuljettajat näkevät omat työlistauksensa. Samalla kuljettajat voivat kuitata tilaukset toteutuneiksi ja tehdä erilaisia merkintöjä tilauksille kuten työselosteen.

Tilaus

Liitteet (0) | Veroluottot (0) | Tavarat (0) | Reikit (0) | Runkoreitit (0) | Projektit (0) | Poikkeamat (0) | Palautteet (0)

Tilauuspäivämäärä: 5.3.2022 15:33

Suunniteltu toimituspäivämäärä: 4.3.2022 00:00

Ajorajustelijan valitsema:

Suunniteltu toimituspäivä sovittu asiakkaan kanssa:

Määräaika: 9.3.2022 00:00

Maksava asiakas: Tukite 29 (Takonen Solutions Oy) Avaa

Toiminto*: Toimitus

Toiminto lukittu*:

Toimitusosoite: Tukite 29 A.S. Teet Avaa
Mobiilinumero: +358405444997

Nouto-osoite: Tukite 29 A.S. FI-01740 Vantaa Avaa

Myyjä: 01 Tukite (Hak) Avaa

Numero: TESTI HA60071327

Tilaustyyppi (laskutus): KDA - Kodinkoneasennus Avaa

Kustannuspaikka: 110114 - Kodinkonejalat ja -asennus

Kustannuspaikka lukittu:

Toimitustyyppi: Kodinkoneasennus vaativa

Tausta-ajot:

Laatu: Hyvä

Tila*: Luonnos

Laskutuslaji: Keskeneneräinen

Tilaja: Sami Tenkanen

Sisäiset huomiot:

Työn tiedot: Vahditään uusi llesi vanhan tilalle, ja vahditään samalla jääkäjin loppu

Vakiotilauksesta luotu:

Muokattu: 05.03.2022 15:41

Muokkaaja: Tenkanen, Sami

Luontiaika: 05.03.2022 15:38

Luoja: Tenkanen, Sami

Kuva 3 Tilauksenmuokkausnäkyminen Sherpa-järjestelmässä

Kuvassa 3 nähdään, miltä tilauksen tarkastelu ja muokkaus näyttää Sherpa-järjestelmässä. Käytännössä kaikki yksittäisen tilauksen käsittelyt sen koko elinkaaren läpi voidaan suorittaa kyseiseltä sivulta. Jotkin nähtävissä olevista kentistä ovat vapaita tekstikenttiä, kun toiset toimivat pääosin hakutyökaluina, joihin kirjoittaessa lopullinen arvo valitaan alavetovalikosta. Valittuja tietueita pääsee muokkaamaan avaamalla ne, jolloin myös tietueen sisältämä data tulee näkyviin. Laskutuksen kannalta tärkeät asiat, kuten veloitukset, toimitetut tavarat sekä tehdyt työt tulevat tilauksen edetessä nähtäville samaan näkymään.

3.3 Analyysi nykytilasta

Prosessiin osallistuvien työntekijöiden kanssa pidetyissä tapaamisissa havaittiin, ettei käsittelijöiden kesken ollut selkeää yhtenäistä mallia tilausten käsittelyyn. Kaksi kolmesta järjestelmästä oli tullut käyttöön viimeisen kuuden kuukauden aikana. Samalla prosessien omistajat sekä suuri osa tilausten käsittelijöistä olivat vaihtuneet, eikä prosesseilla ollut enää selkeää omistajaa. Yhtäaikaisesti myös tilausmäärät olivat kasvaneet. Tämä oli aiheuttanut sen, että pidempään töissä olleet tilaustenkäsittelijät olivat jääneet noudattamaan edellisten prosessien omistajien luomia ohjeistuksia, kun taas uudet työntekijät olivat saaneet eriäviä ohjeistuksia. Käsittelijöiden saamat ohjeistukset eivät koskeneet kaikkia tilauksen käsittelyn vaiheita vaan kattoivat tilauksen kannalta tärkeimmät asiat. Lisäksi ohjeistusta ei aina päivitetty järjestelmään tulleiden muutosten ja päivitysten mukana. Puutteet ohjeistuksessa aiheuttivat sen, että käsittelijät olivat alkaneet luomaan ajansaatossa jokainen hieman omia tapoja työvaiheiden suorittamiseen.

Käytännössä samoja tietoja merkattiin järjestelmissä eri paikkoihin. Joissain tapauksissa tietoja merkattiin varmuuden vuoksi useampaan eri paikkaan, mikäli täyttää varmuutta tiedon oikeasta sijainnista ei ollut. Kaikkien tietojen tarkoitusta tai hyödyllisyyttä ei osattu selittää tai ei ymmärretty. Tietueiden oikeasta esitysmuodosta oli myös eri käsityksiä, jonka seurauksena samoja tietueita, kuten

osoitteita oli useita kopioita. Kopioiden joukossa saattoi olla vain pieniä eroavaisuuksia tai jopa täysin virheellistä tietoa, joka saattoi epähuomiossa päätyä virheellisesti käyttöön.

Tilausten syötössä käytetyt Excel-työkalut olivat, manuaaliseen syöttämiseen verrattuna, tehokas tapa siirtää tilaukset kumppanin järjestelmästä Sherpaan. Työkalut olivat kuitenkin monimutkaisia ja altistivat helposti virheille. Muutamalla virheellä työkalun käsittelyssä saattoi työn joutua aloittamaan alusta. Kumppanien järjestelmistä saatava .csv-tiedoston data oli usein puutteellista ja saattoi sisältää virheitä. Työkalun käsittelijän oli usein vaikea havaita näitä virheitä suuren Excel-taulukon keskeltä, ja näin ollen paljon puutteellista ja virheellistä tietoa päätyi tilauksille. Virheitä ja niiden määriä on selvitetty tarkemmin luvussa 3.5.2.

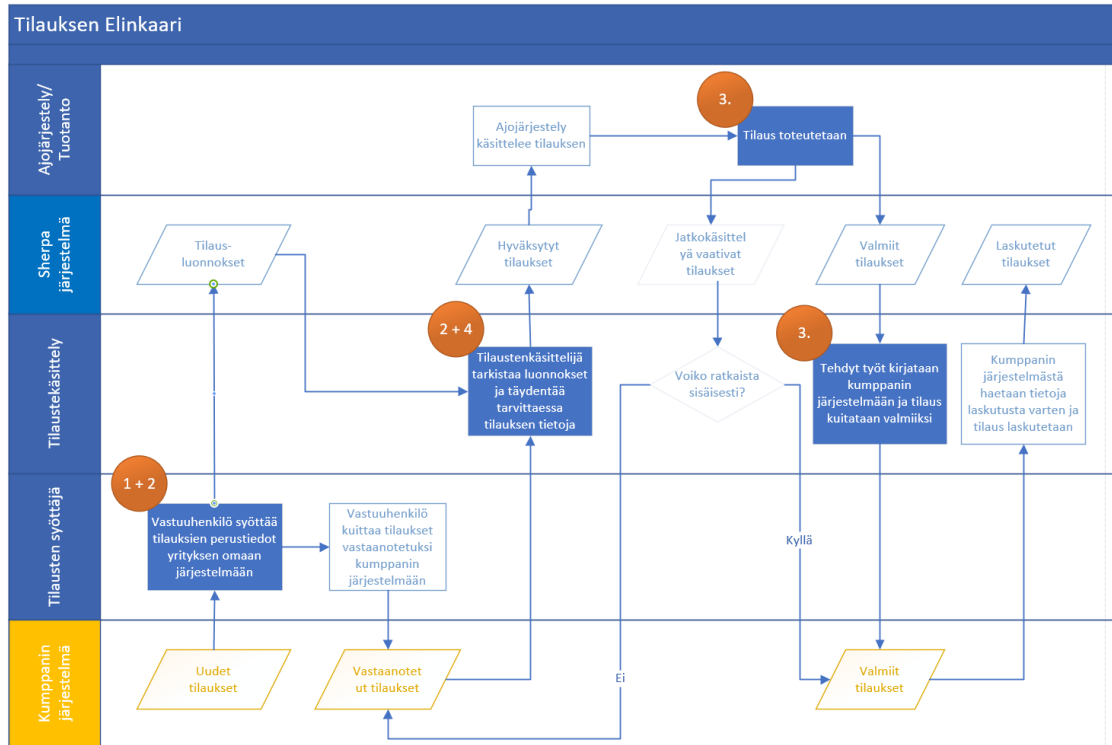
Valmiiden tilausten kuittaus ja laskutus oli hyvin suoraviivainen ja selkeä prosessi. Ongelmia prosessiin aiheutti kuitenkin työselosteen puuttuminen tilauksilta, jolloin käsittelijän oli selvitettävä itse tehdyt työt ja kirjattava ne järjestelmään, yksinkertaisen leikkaa ja liimaa -työvaiheen sijasta.

Työntekijöiden kanssa tehdyissä tapaamisissa nousi esille toive, että varsinkin asiakastiedot tulisi saada järjestelmään oikeassa muodossa. Excel-työkalulla tilauksia tuodessa asiakas tieto usein puuttui oikeasta kentästä, tai työkalu ei ollut osannut valita oikeaa tietuetta. Asiakastiedon puuttuessa oikea asiakas täytyi selvittää, ja valita usein monen duplikaatti tietueen joukosta. Toisena toiveena oli, että tilaukselle tuotava työnkuva tulisi olla mahdollisimman kattava. Tällä hetkellä tilaukselle tuotiin vain lyhyt kuvaus ja lisätietoja jouduttiin kaivamaan kumppanin järjestelmästä.

3.4 Nykytila automatisaation näkökulmasta

Ohjelmistorobotin kannalta on tärkeää, että datan käsittely on loogista ja mahdollisimman standardisoitua. Tiedot olivat kumppanien järjestelmissä pääosin oikeissa kentissä ja sitä kautta suoraan siirrettävissä Sherpa-järjestelmään. Itse

tietueista löytyvä data ei kuitenkaan ollut usein standardimuotoista. Osoitetiedot olivat monesti eri muodoissa, ja tilauksien kuvaukset saattoivat vaihdella sisältämänsä tiedon osalta suuresti. Automatisaation sekä prosessin kannalta voitiin tunnistaa neljä ongelmakohtaa.



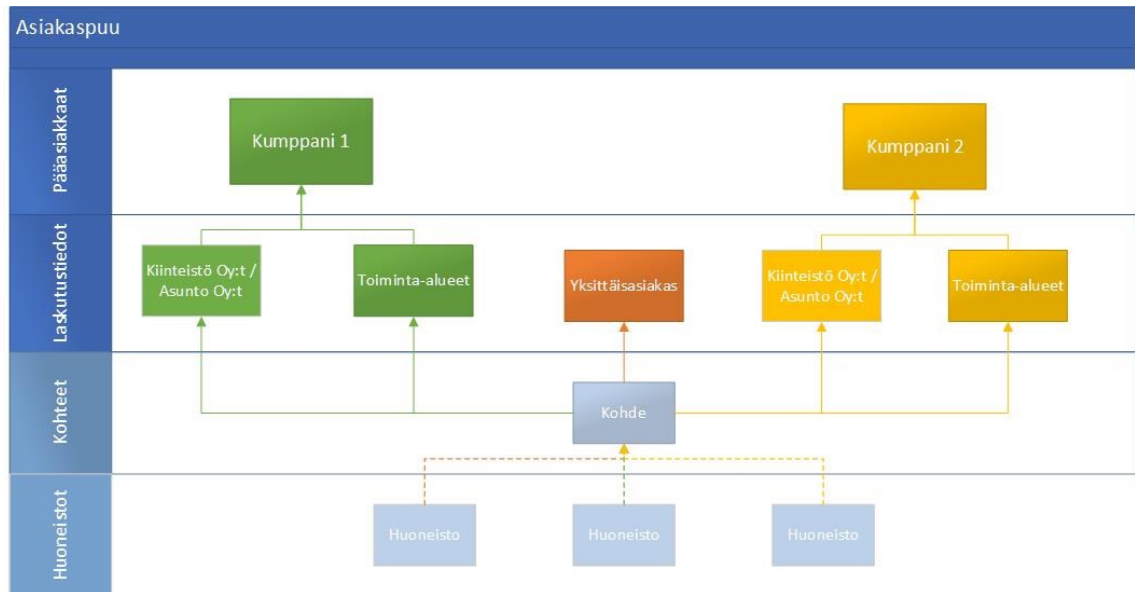
Kuva 4 Havaitut ongelmakohtat ja niihin vaikuttavat prosessit.

Ongelmakohtat ja niiden sijainti tilauksen elinkaaren eri vaiheissa on esitetty kuvassa 4. Tarkemmat kuvaukset ongelmista on esitetty alla.

1. Laskutusasiakkaan valinta tilaukselle

Ensimmäinen ongelmakohta liittyi eri asiakastietojen valintaan. Lähtökohtaisesti kumppanin järjestelmässä ei ollut asiakastietoja suoraan saatavilla, vaan ne poimittiin saatavilla olevasta toimitusosoitetiedosta. Esimerkiksi laskutusasiakkaaksi tilaukselle merkittiin toimitettavan tilauksen kohde. Kohde oli tavallisesti tilauksen toimitusosoite ilman rappu- ja huoneistotietoa. Oikean kohteen valitsemisen teki ongelmaksi monimutkainen asiakaspuurakenne. Nykytila-analyysin

pohjalta kuvattu asiakaspuurakenne yksittäisen kohteen näkökulmasta on esitetty alla.



Kuva 5 Nykytilan mukainen asiakaspuu

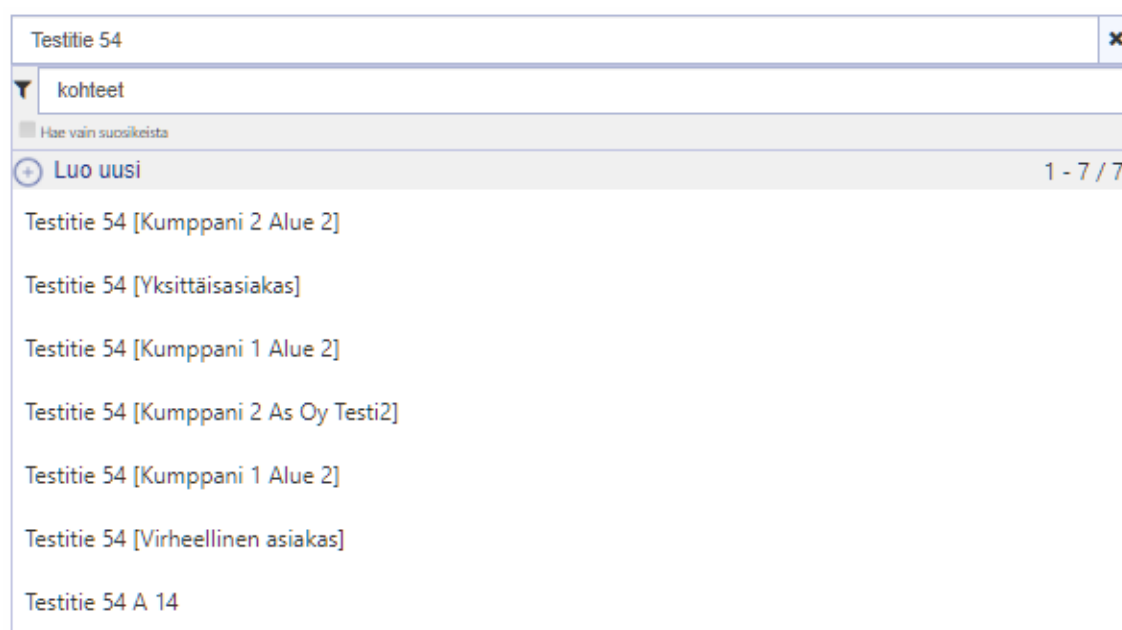
Kuvan 5 mukaisesti asiakaspuu koostuu neljästä kerroksesta. Näistä alin kerros, eli huoneistot, merkataan tavallisesti toimitusasiakkaaksi. Huoneistotieto saadaan kumppanin järjestelmästä poistamalla osoitetiedosta postinumero sekä toimipaikka. Kohde taas oli se rakennus, tai rakennuksen osa, johon huoneisto kuului. Jotkin käsittelijät merkkasivat huoneiston kohteen emoasiakkaaksi, ja toiset jättivät tämän merkinnän tekemättä.

Asiakaspuun toiseksi ylin kerros oli se yritys tai yrityksen laskutusalue, mihin lasku kuului osoittaa. Kohteella ei itsessään ollut laskutustietoja vaan ne saatiin tämän emoasiakkaan kautta. Ylin kerros asiakaspuussa edustaa pääasiakasta, eli kumppania tai konsernia, jonka alaisuuteen tilaus ja laskutusasiakas kuuluivat.

Yksittäiselle kohteelle saattoi tulla tilauksia usealta eri kumppanilta ja laskutusasiakkaalta. Kohteen jokin huoneisto saattoi olla eri kumppanin omistuksessa, kuin toiset kohteen sisältämät huoneistot. Toisaalta tilauksen luonne vaikutti myös laskuttavaan asiakkaaseen. Jotkin korjaustoimenpiteet kuuluivat asunto

osakeyhtiön vastuulle, kun toiset saattoivat kuulua suoraan kumppanin vastuulle.

Jokaiselle eri laskutuskombinaatiolle oli kohteesta tehtävä kopio, jossa emoasiakkaaksi merkattiin oikeat laskutustiedot omaava asiakas. Suurimpien kumppanien osalta nämä kohteet ja kombinaatiot olivat pääosin tiedossa ja valmiiksi merkattuina järjestelmässä. Uusia kohteita kuitenkin syntyi satunnaisesti, jolloin tilauksenkäsittelijän oli kytkettävä se asiakaspuuhun. Ajan saatossa kohteista oli myös syntynyt duplikaattitietoja, joissa asiakaspuukombinaatio oli identtinen toiseen kohteeseen nähden, mutta kirjoitusasu oli hieman erilainen. Samalla joukossa oli myös virheellisiä tai vanhentuneita kohdetietoja, jotka näyttivät lähes oikeilta, mutta sisälsivät esimerkiksi väärät laskutustiedot. Seuraavassa kuvassa on esitetty esimerkkikuva laskutusasiakkaan valintanäkymästä nykytilassa.



Kuva 6 Laskutusasiakkaan valinta Sherpa-järjestelmässä

Kuvassa 6 kohteen nimi on laitettu hakukenttään suodattimeksi, jolloin järjestelmä tarjoaa kaikki kohteen laskutuskombinaatiot. Laskutus- tai emoasiakas

näkyvyydessä hakasulkujen sisällä. Mikäli kuvan tapauksessa lasku tilauksesta kuuluisi osoittaa Kumppani 1 Alue 2:een, joutuisi tilauksenkäsittelijä päättämään kahden identtiseltä vaikuttavan kohteen välillä. Kokeneille käsittelijöille oli kehittynyt vähitellen vaiston omainen taito valita oikea kohde luettelosta. Jos käsittelijä ei ollut täysin varma valinnasta joutui hän käydä tarkastamassa oikean kohteen asiakastietojen kautta.

2. Tilaukselle tehtävät luokittelut

Toinen ongelmakohta liittyi kahteen tilaukselle tehtävään luokitteluun. Ensimmäinen luokittelu tilaukselle tehtiin tilauksen syötön aikana käytettävässä Excel-työkalussa. Tällöin tilaukselle valittiin tilaustyyppi, jolla määritettiin, mille osastolle tilaus kuului ja kuinka kiireellinen tilaus on. Tilaustyyppi osallistui myös tilaukselle käytettävän hinnaston valintaan. Lisäksi tilaustyyppiä tarvittiin yrityksen sisäisessä laskennassa. Luokittelu tehtiin tilauksen kuvauksen perusteella, joka saatiin käytettävään Excel-työkaluun usein hyvin vaillinaisena. Näin ollen tilaustyyppin valinta tilauksen syötön aikana perustui usein enemmän vaistoon kuin tietoon.

Toinen luokittelu tilaukselle tehtiin tilauksen käsittelyssä, jossa tilaukselle valittiin vikakoodi. Vikakoodin kautta järjestelmä määrittäi tilaukselle vaikeustason ajojärjestelyn helpottamiseksi. Koodia oli myös tarkoitus käyttää datanluokittelu-tarkoituksissa tulevaisuudessa. Myös vikakoodin valinta tehtiin tilauksen kuvauksen perusteella. Tilauksen käsittelijöillä ei siis usein ollut tarkkaa tietoa viasta, vaan koodi pyrittiin arvaamaan mahdollisimman todenmukaisesti saatavilla olevan tiedon perusteella.

3. Puuttuvat työselosteet

Kolmas havaittu ongelmakohta havaittiin tilauksen kuittausprosessissa. Ennen laskutusta ja tilauksen kuittaamista kumppanin järjestelmään, täytyi tilauksen käsittelijän viedä tilauksen työseloste kumppanin järjestelmään. Työseloste sisälsi kaikki tilauksella tehdyt työt tai havaitut tarvittavat jatkotoimenpiteet. Selosteen kirjoittaminen oli tilausta toteuttavan huoltomiehen tai asentajan vastuulla.

Mikäli työseloste oli kirjattu tilaukselle, käytiin se kopiaimassa sellaisenaan kumppanin järjestelmään. Mikäli työseloste kuitenkin puuttui tilaukselta, täytyi tilauksen käsittelijän itse kirjata se tilaukselle tai pyytää se ajojärjestelystä.

4. Laitteiden lisääminen tilaukselle

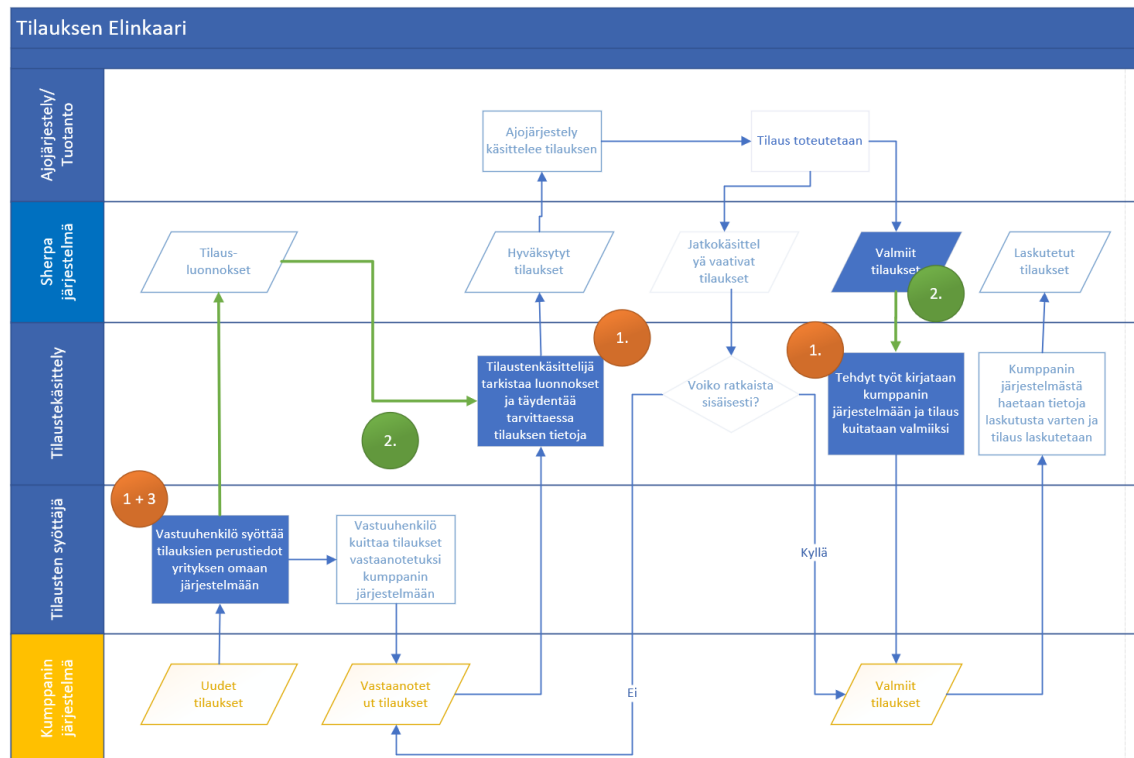
Neljäs ongelmakohta liittyi kodinkoneasennusosaston tilauksiin. Kyseisen osaston tilaukset sisälsivät aina laitteen toimituksen. Laitetta ei ollut tarkkaan määritetty tilaukselle kumppanin järjestelmästä. Tilauksella saattoi lukea ainoastaan ”Liesi” tai ”Jääkaappi”. Tilauksen käsittelijän täytyi lisätä oikea laite tilaukselle käyttäen laitteen laitekoodia. Oikean laitteen valinta perustui pitkälti kokemukseen ja asiantuntemukseen. Käsittelijän täytyi tuntea kyseisen kohteen ja kumppanin käyttämät laitteet ja valita oikea laitekoodin sen perusteella. Yritykseltä löytyi kyllä erinäisiä ohjeita ja työkaluja oikeiden laitteiden valintaan, mutta käytännössä kaikki käsittelijät olivat opetelleet suurimman osan koodeista ulkoa.

3.5 Nykytilan mittarit

Nykytilan ja projektin arvioimista varten toteutettiin mittareita kuvamaan tilauksen elinkaareen ja varsinkin tilauksen käsittelyn prosessien suorituskykyä. Tilausten käsittelyn työvaiheiden suorituskykyä ei yrityksessä mitattu eikä valmiita mittareita siis ollut. Palvelutasosopimuksissa oli tilauksille määritettyinä määräajat, jotka olivat asiakas- ja tilaustyyppikohtaisia. Pääosalla tilauksista oli kolmen päivän määräaika tilauksesta, mutta esimerkiksi rikkinäisten jääkaappien huollot ja vaihdot olivat yhden päivän määräajalla. Yhdessä kumppanin järjestelmässä löytyi erikseen tilaajan asettama määräaika tilaukselle. Muiden järjestelmien tilauksien osalta määräaika asetettiin tilauksen käsittelijän toimesta.

Määräaikojen toteutumista mitattiin vertaamalla tilauksen toteutuspäivämäärää tilaukselle merkittyn määräaikaan. Mittaustapa koettiin kuitenkin ongelmalliseksi, sillä yli puolessa tilauksista lähetettiin asukkaalle viesti, jonka kautta asukas itse varasi toimitusajan tilaukselle. Määräajan toteutumiseen siis vaikuttivat pitkälti asukkaiden omat valinnat.

Nykytilaa varten valittiin ja toteutettiin kolme mittauskohdetta: prosessien käsittelyaika, prosessien vasteaika sekä tilausten virhemäärät.



Kuva 7 Mittauskohteiden sijainti tilauksen elinkaareissa

Kuvassa 7 on esitetty mittauskohteiden sijainti tilauksen elinkaareen eri vaiheissa. Mittauskohteet 1 ja 3 kohdistuvat yksittäisiin prosesseihin ja niiden suorituskykyyn. Mittauskohteella 2 tarkastellaan prosessien välistä vasteaikaa. Tarkemmat kuvaukset mittauskohteista ja niihin toteutetuista mittareista on esitetty alla. Alaluvun numerointi vastaa kuvassa esitettyä numeroa.

3.5.1 Prosessien käsittelyajat

Prosessien käsittelyajat mitattiin nykytilan analyysiä varten tilauksen elinkaaren kolmesta eri vaiheesta: tilausten syötöstä, tilausten käsittelystä, sekä tilausten kuitauksesta. Tilausten syötön osalta valmista dataa käytetystä ajasta ei ollut saatavilla. Kahta eri tilaustensyöttäjää pyydettiin mittamaan kellosta työvaihee-

seen kulunut aika. Mittaukset pyydettiin molemmilta henkilöiltä kahdelta eri käsittelykerralta. Kaikilla käsittelykerroilla ajettiin tilaukset kaikista kolmesta järjestelmästä peräkkäin. Ajankäytössä ei ilmennyt eroja järjestelmien välillä. Toisella käsittelijällä kului noin 30 minuuttia/ajo, kun toisella käsittelijällä kului 60 minuuttia/ajo. Ajot suoritettiin kahdesti päivässä. Tilausten syöttäjät sopivat keskenään, kuka ajon kulloinkin suorittaa. Tilausten syöttöön voitiin siis arvioida kuluvan noin 60–120 minuuttia päivässä.

Tilausten käsittelyaikaa ei ollut tarkkaa mitattu aikaisemmin, eikä suoraa dataa ollut saatavilla. Käsittelijöiden kanssa pidetyissä tapaamisissa havaittiin, että tilauksen käsittelyajat voivat vaihdella suuresti eri tilauksien välillä. Keskiarvon mittaamiseksi tarvittiin siis suurempi otanta, eikä sekuntikelolla ajan mittaaminen ollut tämän takia mahdollista. Mittaustavaksi valittiin tilauksien käsittelyaikojen välisten intervallien mittaukset. Tilausten käsittelyt tehtiin sykleissä, eli yksi käsittelijä keskittyi muutaman tunnin ajan käsittelemään tilauksia tekemättä välissä muita työtehtäviä. Jokaiselle tilaukselle jäi aikaleima sekä käsittelijän allekirjoitus, kun tilaus oli käsitelty. Käsiteltyjen tilauksien data viimeiseltä neljältä viikolta vietiin PowerBI-sovellukseen, jossa data järjestettiin käsittelijän ja käsittelyajan mukaan. Käsittely syklit oli helppo tunnistaa ja yksittäiset syklien ulkopuolella käsitellyt tilaukset poistettiin datasta. Lopuksi tilauksien käsittelyleimojen väliset ajat laskettiin ja taulukoitiin.

Pääasiakas (groups)	Huolto	Kodinkoneasennus	Muut	Total
Järjestelmä 1	4,71	4,21	4,38	4,42
Järjestelmä 2	5,63	4,68	6,62	5,13
Järjestelmä 3	6,26	4,07	4,10	5,38
Total	5,41	4,42	5,75	4,88

Kuva 8 Tilausten keskiarvoiset käsittelyajat (minuuttia/tilaus)

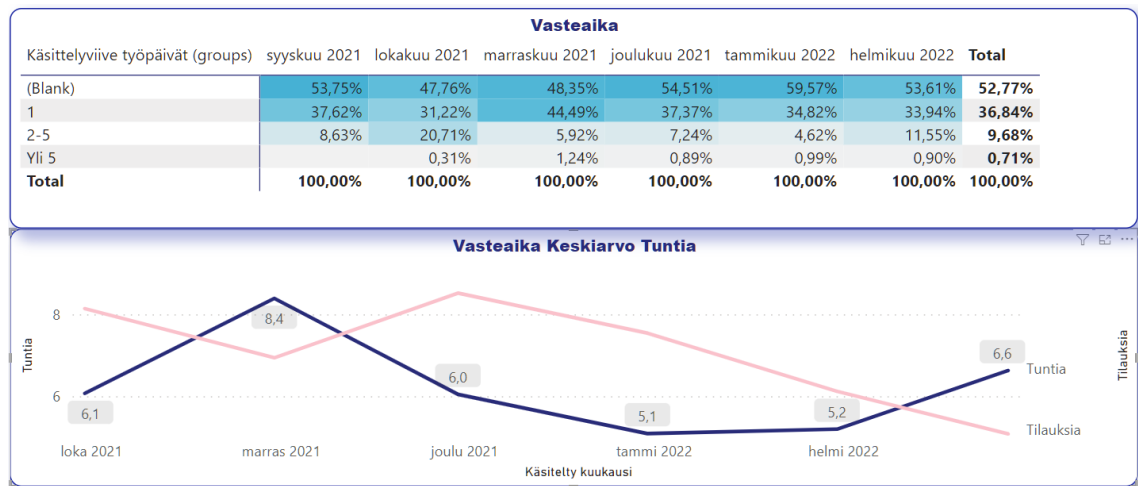
Kuvassa 8 on esitetty lopputuloksena saadut tilauksien keskiarvoiset käsittelyajat. Keskiarvot on jaoteltu käsittelevän osaston sekä tilauksen alkuperän mukaan. Muut-kategorian tilauksien käsittelyyn osallistui molemmat osastot.

Tilausten kuittauksista kumppanin järjestelmään ei jäänyt aikaleimoja Sherpa-järjestelmään, joten intervallien mittaaminen ei ollut tässä tapauksessa mahdollista. Mikäli työseloste löytyi tilaukselta, oli tilausten kuittaminen hyvin suoraviivainen prosessi, jossa vaihtelua käsittelyajoissa tilausten välillä ei ollut odotettavissa. Kuittausajat mitattiin sekuntikellolla 20 tilaukselle. Järjestelmästä riippuen kuittamiseen kului noin 1 minuutti/tilaus. Mikäli työseloste puuttui, mutta se oli käsittelijän itse lisättävissä, kasvoi käsittelyaika noin yhdellä minuutilla/tilaus. Mikäli työseloste täyty kysyä ajojärjestelystä seisahtui tilauksen käsittely, kunnes vastaus saatiin. Tällaisista tapauksista ei aikojen mittaaminen ollut mahdollista.

3.5.2 Prosessien vasteajat

Prosessien vasteaikoja mitattiin nykytilan analyysiä varten tilauksen elinkaaren kahdesta vaiheesta. Vasteaikoja varten tehtiin mittarit ja seuranta PowerBI-sovelluksen avulla.

Tilauks käsittelyn vasteaikaa varten mitattiin aika tilauksen luomisen ja käsitelty aikaleimojen välille. Näin saatuja vasteaikoja voitiin mitata ainoastaan Huolto-osaston tilauksista sekä kiireellisistä kodinkoneasennustilauksista. Tämä johtui siitä, että monet kodinkonetilaukset käsiteltiin tarkoituksella mahdollisimman myöhään, jottei laitteita sitoutuisi tilauksille tarpeettomasti. Huollon tilaukset taas tuli käsitellä mahdollisimman nopeasti. Saadut tulokset huolto osaston vasteajoista on esitetty kuvassa 9.



Kuva 9 Käsittelyn vasteajat

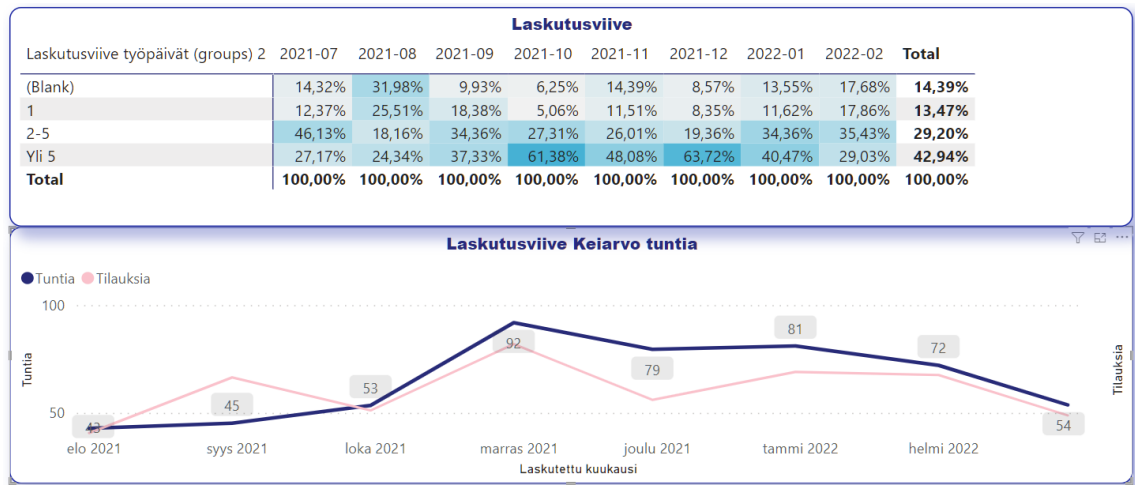
Kuvan ylemmässä puoliskossa tilaukset on kategorisoitu luokkiin. Luokat kertovat, kuinka monta työpäivää tilauksen luomisesta sen käsittelyyn on kestänyt. ”(Blank)” Luokka edustaa nollaa päivää, eli saman päivän aikana käsiteltyä tilausta. Jotta tilauksen kolmen päivän määräaika täytyisi varmuudella, tulisi tilauksen olla käsitelty viimeistään seuraavana päivänä tilauksen luomisesta.

Alempi viivakavio näyttää vasteaikojen keskiarvon kehityksen. Tässä kuvassa viiveestä on poistettu kaikki työajan ulkopuoliset tunnit ennen keskiarvon laskemista. Keskiarvo siis osoittaa, kuinka monta työtuntia keskimäärin kestää tilauksen luonnista sen käsittelyyn. Taustalla oleva vaaleanpunainen viiva osoittaa käsiteltyjen tilauksien määrän kehityksen.

Vasteaikoja tarkasteltiin myös erikseen kiireellisille yhden päivän määräajan omaaville tilauksille sekä kaikille kolmelle järjestelmälle erikseen. Tilauksen kiireellisyys ei näyttänyt juurikaan vaikuttavan tilauksen käsittelyn vasteaikoihin. Järjestelmien välillä oli kuitenkin nähtävissä selkeää eroa vasteajoissa.

Tilauksen kuittausten osalta käytettiin samanlaista mittaustapaa kuin niiden käsittelystä. Ajanlasku tilauksen laskutusviiveestä alkoi, kun tilaus oli toimitettu, tarkastettu ja merkitty laskutusvalmiiksi. Kuittausaikaleimana käytettiin tilauksen laskutusajankohtaa. Käytännössä useampi tilaus käytiin ensin kuittaamassa

kumppanin järjestelmiin, jonka jälkeen kuitatut tilaukset laskutettiin kerralla. Saadut tulokset on esitetty seuraavassa kuvassa.



Kuva 10 Laskutusviive

Laskutuksen osalta ei ollut kumppanien kanssa sovittua määräaika. Yrityksen oma ohjeistus ja tavoite oli, että tilaukset tulisi olla laskutettuna seuraavan päivän sisällä siitä, kun tilaus on tarkistettu ja asetettu laskutusvalmiiksi. Kuvasta 10 nähdään, että laskutusviiveen mukaiset tavoitteet eivät useinkaan toteutuneet. Laskutusviiveen osalta vaikutti myös olennaisesti, täytyikö tilaukselle hakea hankintatilausnumero kumppanin järjestelmästä ennen laskutusta.



Kuva 11 Laskutusviive kategorioittain nykytilassa

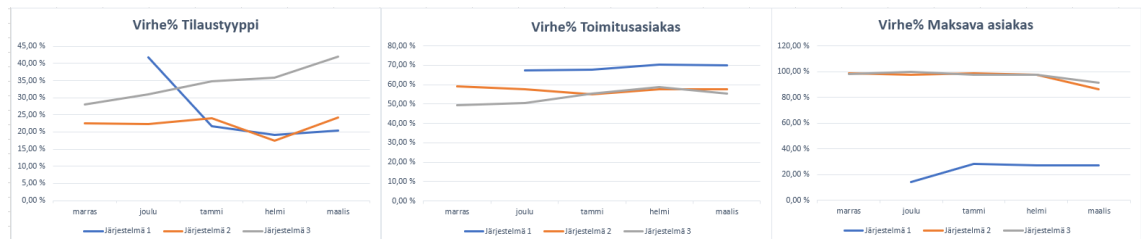
Kuvassa 11 on jaettu tilaukset kahteen kategoriaan. Punainen arvo edustaa tilauksia, jossa hankintatilausnumero on täytynyt hakea kumppanin järjestelmästä, sininen viiva taas edustaa kaikkia muita tilauksia. Graafista huomataan,

että laskutusviive on noin kaksin-kolminkertainen tilauksilla, joilla hankintatilausnumero oli haettava.

3.5.3 Tilauksien virhemäärät

Viimeisenä mitattiin tilauksen syötön virhemääriä. Tulokset saatiin ensin yhdistämällä kaikki järjestelmään syötetyt Excel-tiedostot yhteen taulukkoon. Sen jälkeen haettiin Excel-tiedoston sisältämien tilauksien tiedot yrityksen Sherpa-järjestelmästä. Mikäli järjestelmästä saatu tietue oli eri kuin Excelissä oleva, tulkittiin tämä virheenä. Virheitä mitattiin kolmesta eri tietueesta.

Virhekategorioiden päätettiin niiden merkittävyyden ja mitattavuuden mukaan.

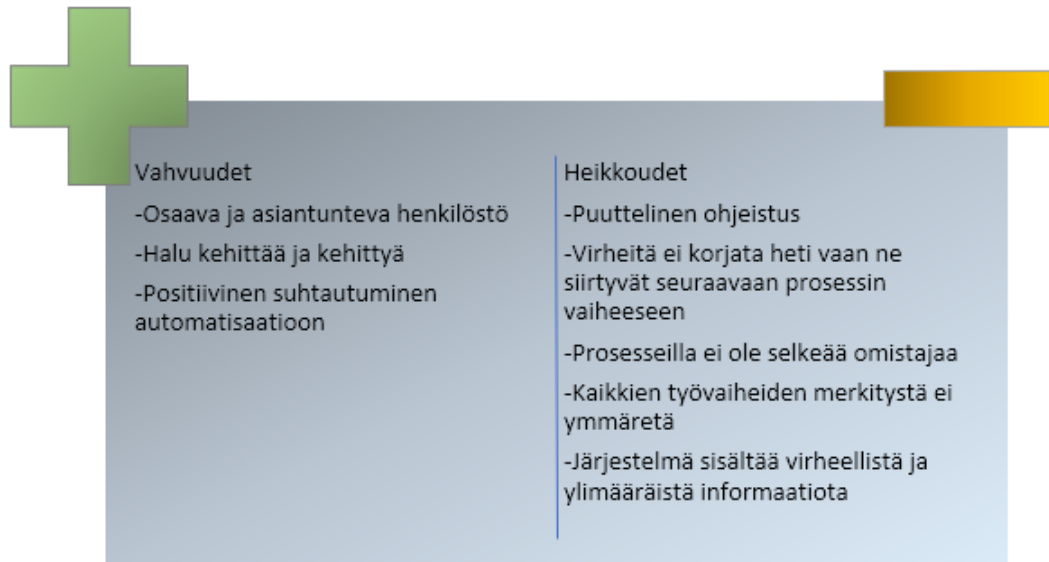


Kuva 12 Tilauksien syötön virhemäärät nykytilassa

Kuvassa 12 on esitetty virheellisten tilausten ilmaantuvuus prosentteissa virhetyyppin mukaan. Tilastoista on nähtävissä, että lähdejärjestelmien välillä oli suuriakin eroja virheiden määrässä. Esimerkiksi järjestelmistä 1 ja 2 tulleissa tilauksissa melkein kaikista tilauksista puuttui maksava asiakas tieto tai se oli virheellinen. Tarkastelussa olevien virheiden lisäksi mobiilinumeron puuttui kaikista tilauksista. Myös työselosteissa oli henkilöstön mukaan paljon virheitä ja puutteita. Näiden virheiden määriä ei kuitenkaan voinut mitata jälkikäteen, sillä työseloste kenttään saatettiin tehdä erinäisiä merkintöjä tilauksen elinkaaren eri vaiheissa.

3.6 Yhteenveto nykytilasta

Tässä luvussa on esitetty lyhyt yhteenveto nykytilasta ja siitä tehdyistä havainnoista.



Kuva 13 Nykytilan vahvuudet ja heikkoudet

Kuvassa 13 on esitetty nykytilan vahvuudet ja heikkoudet. Vahvuuksina voidaan mainita kokenut ja asiantunteva henkilöstö. Ilman kokenutta henkilöstöä, joka muun muassa pystyi tunnistamaan virheellisiä tietueita nopeasti, olisi käsittely prosessissa kestänyt todennäköisesti huomattavasti pidempää. Työntekijöillä oli myös selkeä halu kehittää prosesseja, joka tuli esille tapaamisten aikana. Prosessien automatisointiin suhtauduttiin mielenkiinnolla ja positiivisesti. Automatisaatio nähtiin asiana, joka vapauttaisi aikaa muille työtehtäville ja itsensä kehittämiselle.

Nykytilassa oli myös havaittavissa selkeitä heikkouksia. Prosesseilla ei ollut selkeää omistajaa eikä ohjeistusta. Järjestelmään tehtyjä päivityksiä ei myöskään huomioitu ohjeistuksessa. Samoja työvaiheita saatettiin tehdä uudestaan tilauksen elinkaareen eri vaiheissa, ja virheet usein siirtyivät elinkaareen mukana

eteenpäin. Prosessien suorituskykyä ei juurikaan mitattu, ja olemassa olevat mittarit eivät kuvastaneet prosessien suorituskykyä luotettavasti. Järjestelmässä oli paljon ylimääräistä, vanhentunutta ja virheellistä tietoa, joka vaikeutti työtehtävien suorittamista ja altisti virheille. Automatisaation kannalta ongelmallisia olivat jotkin prosessissa tehtävät harkinnanvaraiset luokittelut, sekä suuri virheellisten tietueiden määrä.

Mittauskohde	Tarkennettu mittauskohde	Tulos keskimäärin
Käytetty työaika	Tilausten syöttö	60-120 min/pvä
	Tilausten käsittely	5 min/tilaus
	Tilausten kuittaus	1 min/tilaus
Vasteajat	Käsittelyn vasteaika ka.	5-7 tuntia
	Tilauksia käsitelty tavoiteajassa	85 %
	Kuittauksen vasteaika ka.	50-80 tuntia
	Tilauksia kuitattu tavoiteajassa	30 %
Virhemäärät	Tilaustyyppi	20-40 %
	Toimitusasiakas	50 - 70 %
	Laskutusasiakas	80 - 100 % (yhdessä järjestelmässä vain 25%)

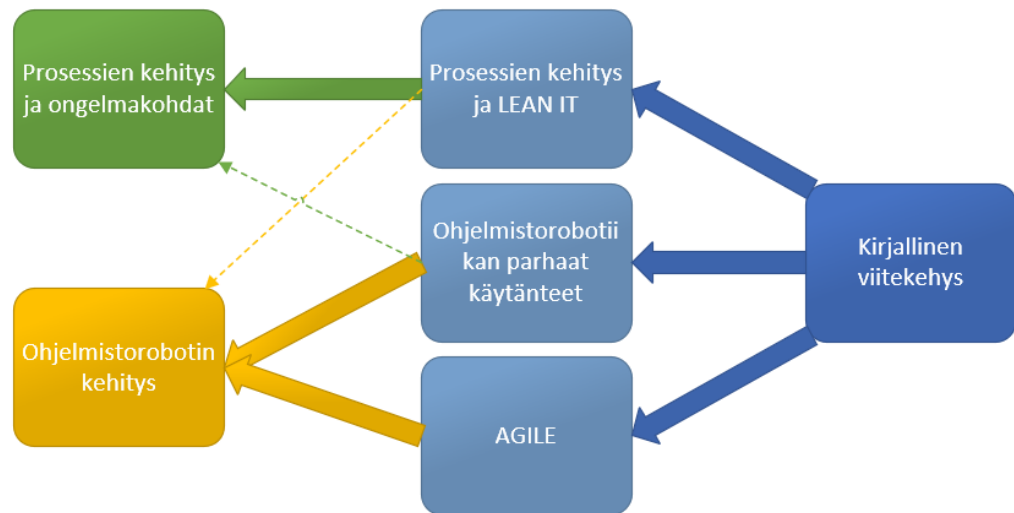
Kuva 14 Yhteenveto nykytilan mittaustuloksista

Kuvassa 14 on esitetty yhteenveto nykytila-analyysin aikana toteutettujen mittauksien tuloksista. Tulokset edustavat kaikkien lähdejärjestelmien suurin piirtein keskiarvoa.

Nykytila-analyysin aikana toteutettujen mittarien avulla määritettiin tilausten käsittelyajaksi noin 5 minuuttia tilausta kohden. Käsittelevä osasto sekä tilauksen lähdejärjestelmä vaikuttivat hieman käsittelyaikoihin. Myös tilaustensyötön virhemäärä vaihteli tilauksen lähde järjestelmästä ja virheen tyypistä riippuen. Suurin virhemäärä tapahtui laskutusasiakkaan valinnassa, jossa virheitä oli melkein 100 % kahdesta järjestelmästä tulevista tilauksista. Toimitusasiakkaan kohdalla virheitä havaittiin noin 50–70 % tilauksista, ja tilaustyyppin kohdalla noin 20–40 % tilauksista.

4 Parhaat käytänteet

Tässä luvussa esitetään kirjallinen viitekehys prosessin kehitystoimenpiteille sekä ohjelmistorobotin käyttöönotossa ja sen ohjelmoinnissa käytettävät parhaat käytänteet.



Kuva 15 Kirjallinen viitekehys

Kuvassa 15 on esitetty kirjallisen viitekehysten sisältö ja sen osien suhde opinäytetyön eri vaiheisiin.

4.1 Ohjelmistorobotiikan parhaat käytänteet

Tässä luvussa käsitellään ohjelmistorobotiikkaan ja automatisaatioon liittyvää kirjallisuutta ja sieltä esiinnousseita parhaita käytäntöjä ohjelmistorobotiikkaan liittyen. Parhaita käytäntöjä tarkastellaan kolmesta eri ohjelmistorobotiikan kehitysvaiheesta: suunnittelusta, kehityksestä ja ylläpidosta.

4.1.1 Ohjelmistorobotiikka lyhyesti

Ennen ohjelmistorobotin käyttöönottoa on hyvä ymmärtää, mitä ohjelmistorobotiikka on ja ei ole sekä mitkä sen vahvuudet ja heikkoudet ovat. The Robotic Process Automation Handbook määrittelee ohjelmistorobotiikan ohjelmistona, joka matkii ihmisen toimintaa tietokoneella. Englannin kielellä vakiintunut nimitys Robotic Process automation (RPA) voikin olla hieman harhaanjohtava, sillä ohjelmistorobotiikka ei suoranaisesti automatisoi prosesseja vaan työntekijöiden suorittamia yksittäisiä työtehtäviä.

Ohjelmistorobotti koulutetaan työtehtävään tavanomaisesti piirtämällä erilaisia prosessikaavioita. Kaavioiden osaset koostuvat palasista, jotka kertovat robotille, mitä tehdä missäkin vaiheessa esimerkiksi klikkaa tästä tai lue ruudulla näkyvä arvo ja tee päätös sen perusteella. Ohjelmistorobotin kouluttaminen vaatii erittäin vähän, jos ollenkaan osaamista ohjelmointikielistä.

Ohjelmistorobotilla on useita vahvuuksia muihin ohjelmistokielistä ja sovelluksilla tapahtuviin automatisaatioihin verrattuna. Yksinkertaisissa automatisaatioissa robotin kehitys ja tuotantoon vienti on usein huomattavasti nopeampaa ja vähäriskisempää.

Robotti ei tavallisesti vaadi muutoksia muihin sovelluksiin. Tavanomaisen automaation suorittaminen sovelluksen koodia muuttamalla vaatii usein tarkkaavaisuutta, sillä pienen muutoksen tekeminen sovellukseen voi vaikuttaa sovelluksen muihin toimintoihin. Robotin kehityksessä näitä vuorovaikutuksia ei tarvitse miettiä, sillä robotti ei vaikuta itse sovellusten toimintaan. Tämä vuorovaikutusten puuttuminen mahdollistaa myös robotin kytkemisen pois päältä heti esimerkiksi virheiden sattuessa ilman, että tarvitsee miettiä, vaikuttaako poiskytkeminen johonkin muuhun toimintoon. Näin ollen robotti voidaan viedä tarvittaessa nopeastikin tuotantoon ilman pitkällistä testausta. Robotin avulla on siis mahdollista automatisoida nopeasti monia pieniä työtehtäviä, joka muuten olisi kannattamatonta tai vaikeaa. Monet pienet automaatiot kertyessään voivat aiheuttaa merkittäviä säästöjä yritykselle.

Robotin avulla on myös mahdollista automatisoida työtehtäviä, jotka tapahtuvat järjestelmissä, joihin automaation rakentaminen ei muuten olisi mahdollista. Tällaisia voisi olla jokin ulkopuolisen tahon ylläpitämä järjestelmä, tai jokin yrityksen käyttämä vanha sovellus, jonka tuki on loppunut. Lisäksi robotin kehitys ei usein vaadi yrityksen IT-osaston osallistumista, joka mahdollistaa IT-osaston keskittymisen muihin tärkeämpiin tehtäviin. Ohjelmistorobotiikalla on myös havaittu olevan positiivinen vaikutus henkilöstön sitoutumiseen ja tyytyväisyyteen, kun henkilöstön aikaa on vapautunut rutiininomaisista työtehtävistä muiden mielenkiintoisempien työtehtävien suorittamiseen.

Yksi robotiikan hyödyistä on myös datan laadun paraneminen. Robotti voidaan kouluttaa syöttämään ja käsittelemään data aina samalla tavalla. Samalla poistuu inhimillisten virheiden mahdollisuus. Tiedon parempi laatu edesauttaa tulevien automatisaatioiden rakentamista, vähentää virheiden siirtymistä seuraaviin työvaiheisiin, ja on tärkeässä roolissa mahdollisen analytiikan ja tekoälyn hyödyntämisessä.

Ohjelmistorobotiikalla on myös heikkoutensa. Robotista maksetaan tavallisesti kausittaista lisenssimaksua. Lisenssimaksun lisäksi robotti tarvitsee palvelimia ja laitteita toimiakseen. Robotin käyttöönoton jälkeen robotin vaihtaminen toisen palveluntarjoajaan robottiin on hankalaa, ja vaatisi usein robotin kehittämisen täysin alusta. Robotti vaatii myös osaavan henkilöstön kehittämään ja ylläpitämään sitä. Robotti myös vaatii jatkuvaa seuranta ja ylläpitoa. Pienet muutokset robotin käyttämiin järjestelmiin voivat aiheuttaa virheen robotissa. Päivitykset yrityksen johonkin järjestelmään vaativat siis usein myös päivityksen robotille. Robotin kehittäminen vaatii myös sen kehittäjältä yksityiskohtaisen tiedon jokaisesta työvaiheesta, jonka robotin kuuluu suorittaa. (Taulli, 2020.)

4.1.2 Ohjelmistorobotiikan suunnittelu

Edellisessä kappaleessa todettiin, että ohjelmistorobotiikan avulla on mahdollista automatisoida työtehtäviä nopeasti. Robotti ei kuitenkaan itsessään korjaa

huonoa tai virheellistä prosessia. Samoin kuin muutokset robotin käyttämiin järjestelmiin, vaatii robotin kehityksen jälkeiset prosessien muutokset usein myös muutoksia itse robotin toimintaan. Robotin on tästä syystä vaikea toimia jatkuvasti muuttuvassa ympäristössä. Ennen automatisointia on siis tärkeää, että työvaiheet ovat standardoituja. Lisäksi ennen robotin kehitystä on suositeltavaa tarkastella koko prosessia ja varmistaa, että prosessin toiminta on mahdollisimman tehokasta jo ennen työtehtävien automatisointia. (Taulli, 2020.)

Tehokkaan prosessin olemassaolon tärkeyttä ennen ohjelmistorobotin implementointia tukee myös RPA:sta tehty kirjallisuustutkimus. Tutkimuksen mukaan kirjallisuus tukee ajatusta siitä, että prosessi tulisi optimoida tai jopa uudelleen rakentaa ennen automatisointia. Myös päinvastaisia ajatuksia, joissa suositellaan automatisoimaan ohjelmistorobotiikan avulla juuri tehottomat ja ei-optimoidut työvaiheet, on olemassa. (Syed ym. 2019: 7.)

Prosessien kehityksessä, ennen ohjelmistorobotin kehitystä sekä kehityksen aikana voidaan hyödyntää erilaisia prosessi metodologioita. Yhdessä RPA:n kanssa on havaittu toimivaksi ainakin Lean, Six sigma ja Lean Six sigma metodologiat. (Taulli, 2020.) Näistä Lean tarjoaa Six sigmaan nopeamman ja kevyemmän ratkaisun prosessin kehitykseen. Leanin avulla voidaan myös tehdä helpommin lyhyitä tarkennettuja kehitys toimenpiteitä, kun taas Six sigma vaatii usein enemmän aikaa ja koulutusta tuloksien saamiseksi. (Antony, 2011.)

Itse robotin kehityksessä ja ylläpidossa käytetyimmät mallit ovat Lean ja Leanin pohjalta johdettu Agile-malli (Fantina ym. 2021). Taulli suosittelee kirjassaan ensin Lean-mallin omaksumista ennen RPA:n implementointia. Kun Lean-malli on omaksuttu, voidaan tarvittaessa kehitystä jatkaa Six sigman avulla. (Taulli, 2020.)

Kun prosessi on optimoitu, voidaan automaation kannattavuutta mitata. Kannattavuutta voidaan mitata laskemalla automaatiolle ROI (Return of Investment). ROI voidaan laskea arvioimalla automatisoitaviin työtehtäviin kuluvat henkilötyötunnit ja laskemalla niiden kustannus yritykselle. Näin saadaan automaation

säästö, josta on vielä poistettava automaatioon kuluneet kustannukset, jolloin saadaan automaation tuotto. (Taulli, 2020.)

RPA:lla on kuitenkin usein muitakin hyötyjä, jotka eivät näy suoraan rahallisena säästönä, mutta jotka tulisi ottaa huomioon automatisaation kannattavuutta määritettäessä. Näitä ovat muun muassa prosessin ja automatisoitavan työtehtävän sisällä tapahtuva virheiden määrä. Robotti voi vähentää virheiden määrää huomattavasti. Virheiden vähentyminen voi näkyä jossakin toisessa prosessissa tai työvaiheessa vähentyneenä työaikana. Pienempi virheiden määrä voi myös parantaa asiakastyytyväisyyttä tai mahdollistaa parempaa analytiikkaa tai tekoälyn käyttöä. Robotti voi myös pienentää prosessin läpimeno- ja jonotusaikoja ja tuoda näin lisäarvoa asiakkaalle. Myös työntekijöiden tyytyväisyys tulisi ottaa kannattavuuden arvioinnissa huomioon. Pienemmät automatisaatiot eivät välttämättä vähennä merkittävästi käytettyä työaikaa, mutta voivat parantaa työntekijöiden tyytyväisyyttä, joka edelleen voi parantaa työntekijöiden tehokkuutta, pienentää vaihtuvuutta tai lisätä innovointia. (Taulli, 2020.)

Lopuksi automaatiolle kannattaa määrittää tavoitteet tai päämäärät. Nämä auttavat mittaamaan automaation onnistumista projektin lopuksi sekä ohjaamaan robotin kehitystä haluttuun suuntaan. Päämäärät voivat olla esimerkiksi työtehtäviin käytetyn työajan vähentyminen, nopeammat jonotusajat tai virhemäärien väheneminen. (Taulli, 2020.)

4.1.3 Ohjelmistorobotin kehitys

Aikaisemmissa kappaleissa on todettu, että ohjelmistorobotin avulla tehty automaatio voidaan tehdä oikein suoritettuna nopeallakin aikataululla. Nopean kehityksen tueksi on olemassa erilaisia parhaita käytäntöjä. Näiden parhaiden käytäntöjen hyödyntäminen nopeuttaa robotin kehitystä ja samalla varmistaa, että robotti täyttää vaaditut kriteerit ennen kuin se viedään tuotantoon. Suurin osa ohjelmistorobottien tuottajista tarjoaa käyttäjilleen oman viitekehityksen robotin kehityksen ja sen elinkaaren ylläpidon tukemiseksi. Blue Prismillä tämä viitekehitys on nimeltään ROM (Robotic Operating Model).

ROM koostuu 7 tukipilarista, joista opinnäytetyön kannalta keskeisimpänä on **Delivery Methodology**. Delivery Methodology tarjoaa käyttäjälleen suoritteita ohjelmistorobotin kehityksen tueksi. Suoritteet voivat olla erilaisia dokumentteja tai parhaita käytäntöjä, joiden tarkoituksena on tukea automatisaatioprosessia. Suoritteiden avulla pyritään varmistamaan, että automatisaation vaiheet tehdään tehokkaasti ja erilaiset riskit otetaan huomioon. Suoritteiden hyödyntäminen muodostaa myös yhtenäisen toimintatavan robottien kehittäjien välille mahdollistaen robotin ja automatisaation jouhevan ylläpidon ja jatkokehityksen henkilöstön vaihtuessa. Jokaisen suoritteen täyttäminen jokaisessa automaatioprojektissa ei ole pakollista, vaan suoritteita voidaan käyttää valikoidusti työkaluina tarpeen mukaan.

Kehitysvaiheessa kannattaa hyödyntää jo olemassa olevia resursseja. Monilla ohjelmistorobottien tarjoajilla on omat ”bottikaupat”, joista voi ladata valmiita prosessipohjia tai toiminnallisuuksia robotille. Prosessipohjissa on valmiiksi otettu erilaiset parhaat käytännöt ja poikkeuksien hallinta huomioon, jolloin robotin kehittäjän tarvitsee yksinkertaistettuna vain lisätä omat työvaiheet prosessikaavioon. Valmiit toiminnallisuudet taas tarjoavat esimerkiksi nopean ratkaisun jonkin tietyn sovelluksen käyttämiseen. Tällainen voisi olla esimerkiksi Excel-toiminnallisuus, josta löytyvät kaikki Excel-automaatiota varten tarvittavat toiminnot valmiina. Tätä käyttäessä kehittäjän tarvitsee vain valita tarvitsemansa toiminto, eikä robottia tarvitse erikseen ohjeistaa siitä, kuinka Exceliä käytetään. (Taulli, 2020.)

Kun robotti kuitenkin joudutaan kouluttamaan käyttämään jotakin järjestelmää tai sovellusta, kannattaa tämä jakaa moneen pieneen palaseen tai aktiviteettiin. Sama koskee myös robotille tehtäviä prosesseja. Osiin jakaminen mahdollistaa aktiviteettien ja prosessien uusiokäyttöä. Esimerkkinä on kaksi erillistä työtehtävää, jotka käyttävät samaa järjestelmää. Toisessa tehtävässä syötetään tilauksia ja toisessa niitä taas haetaan. Molemmissa tehtävissä on kuitenkin ensin kirjaututtava sisään järjestelmään. Mikäli järjestelmään sisäänkirjautuminen on jaettuna omaksi aktiviteetiksi, voidaan tätä samaa aktiviteettiä käyttää molemmissa työtehtävissä. (Taulli, 2020.)

Robottia kehittäessä on myös tärkeää noudattaa selkeyttä ja luettavuutta. Robotille on ominaista, että työvaiheita joudutaan päivittämään järjestelmien päivityessä tai prosessin muuttuessa. Robotille tehtyjen prosessien selkeys nopeuttaa tarpeellisten muutoskohtien löytämistä ja robotin päivittämistä. Selkeys myös helpottaa useamman kehittäjän työskentelyn samojen prosessien parissa. Myös uuden prosessin kehitys on nopeampaa, mikäli siinä voidaan käyttää aiemmin luotuja selkeästi tehtyjä prosesseja tai aktiviteetteja. (Taulli, 2020.)

Skaalautuvuus on yksi ohjelmistorobotiikan suurimmista vahvuuksista. Tämän takia robotin kehityksessä tulisi myös aina ottaa huomioon volyymien nousu ja robotin jatkokehitys. Robotin tulisi olla kehitetty niin, että yksittäisen työtehtävän automaatiota voidaan helposti laajentaa tulevaisuudessa. (Taulli, 2020.)

4.1.4 Ohjelmistorobotin tuotantoon vienti ja ylläpito

Robotin tuotantoon viennin voi tehdä monella tavoin. Mikäli automaatio laajuudeltaan suuri tai liittyy kriittisiin liiketoimintoihin, voi tarkempi parhaiden käytäntöjen seuraaminen tulla kysymykseen. Parhaiden käytäntöjen mukaan robotti ensin kehitetään ja testataan, jonka jälkeen se vielä viedään testattavaksi itse loppukäyttäjille. Mikäli loppukäyttäjät hyväksyvät ratkaisun voidaan robotti tuoda tuotantoon. Eri testausvaiheessa on hyvä käyttää apunaan erilaisia tarkastuslistoja, joiden avulla voidaan varmistua robotin täyttävän kaikki vaaditut kriteerit. (Fantina ym. 2021.) Blue Prism tarjoaa erilaisia tarkastuslistoja käyttäjilleen, joita tuotantoon viennissä voidaan hyödyntää. Nämä tarkastuslistat ovat osa Blue Prismin delivery metodologyn suoritteita. Kaikissa tapauksissa erillistä testausympäristöä ei välttämättä ole kuitenkaan saatavilla vaan robotti on rakennettava suoraan tuotanto ympäristöön.

Tuotantoon vienti kannattaa suorittaa asteittain aloittamalla esimerkiksi jostain yksittäisestä työvaiheesta tai osastosta. Kun robotti on todettu toimivaksi, voidaan toimintoja laajentaa. Tuotantoon viennissä on myös hyvä rajata vastuualueet työntekijöiden ja robotin kesken. Muiden työntekijöiden on tärkeää ymmärtää, mitä tehtäviä robotti tekee ja ei tee sekä mitä sen toiminta vaatii muilta

työntekijöiltä. Vastuu on myös jaettava robotin suorittamista työtehtävistä siltä varalta, että robotti ei toimi. Tällöin vastuutetun henkilön on suoritettava robotille osoitetut tehtävät. Tuotantoon vienti tulisi tapahtua yrityksen CoE:n (Centre of Excellence) kautta, mikäli yrityksellä tällainen on. (Taulli, 2020.)

Tuotantoon viennin jälkeen robottia on seurattava ja ylläpidettävä. Tavallisesti ohjelmistorobotit sisältävät työkalut robotin toiminnan ja suorituskyvyn seuraamiseen. Monitorointi vaiheessa tulisi palata suunnittelu vaiheessa määritettyihin tavoitteisiin ja seurata niiden toteutumista. Mikäli selkeitä mitattavia tavoitteita ei suunnitteluvaiheessa ole määritetty, tulisi robotille tässä vaiheessa viimeistään määrittää raja-arvot, jotka tulisi täyttyä. (Taulli, 2020.)

Ensin katsotaan, että robotti täyttää suunnittelussa määritetyt kriteerit. Mikäli kriteerit eivät täyty, tulisi juurisyyt etsiä ja robottiin tehdä mahdolliset tarvittavat päivitykset. Mikäli suunnitellut kriteerit täyttyvät, jätetään robotti työskentelemään. Robotin suorituskykyä tulisi kuitenkin säännöllisesti palata tarkastelemaan, ja selvittää tarvitseeko muutoksia tehdä. (Taulli, 2020.) Robotin voi myös ohjelmoida ilmoittamaan itse esimerkiksi sähköpostilla, mikäli suorituskyky laskee jonkin raja-arvon alle (Fantina ym. 2021).

Onnistuneen tuotantoon viennin jälkeen tulisi alkaa selvittämään robotin jatkokehitys mahdollisuuksia tai muita automatisaation kohteita. Robotti voi työskennellä 24 tuntia vuorokaudessa, ja parhaan ROI:n takaamiseksi kannattaa robotin koko työskentelyaika pyrkiä hyödyntämään. Jatkokehitys kohteita ei kuitenkaan kannata suoraan rajata pois ROI:n perusteella. Mikäli suuren ROI:n omaavia kehityskohteita ei ole, on todennäköisesti parempi työllistää vähemmän tuottavalla automaatiolla kuin jättää robotin työaikaa käyttämättä. (Taulli, 2020.)

4.2 Prosessien kehitys

Tässä luvussa selvitetään malleja ja parhaita käytäntöjä prosessien kehitykseen. Ohjelmistorobotiikan parhaista käytännöistä voitiin havaita, että proses-

sien kehitys ennen robotin implementointia on suositeltavaa. Samalla voitiin todeta myös, että ohjelmistorobotiikan on havaittu toimivan hyvin yhdessä Lean ajattelumallin kanssa. Prosessien kehityksessä keskitymme siis Lean ajattelumalliin, ja sen tarjoamiin työkaluihin. Koska ohjelmistorobotiikka ja automatisoitava kohde liittyvät keskeisesti IT-ympäristöön, on työssä keskitytty varsinkin Lean IT -teokseen.

4.2.1 Lean IT

Lean IT laajentaa Lean-johtamisfilosofian IT-ympäristöön. Laajennuksessa esitetään, kuinka Leanin mukaista ajattelumallia voidaan hyödyntää IT-yritysten sekä IT:tä hyödyntävien yritysten liiketoiminnassa. Filosofian päämääränä on asiakaskeskeinen jatkuvasti kehittyvä organisaatio, jossa jokainen työntekijä osallistuu ongelmien ratkaisuun ja yrityksen kehittämiseen. Päämäärän saavuttamiseksi Lean IT esittää erilaisia parhaita käytäntöjä ja työkaluja.

Liiketoiminnan kehitys tapahtuu Lean-ajattelussa poistamalla hukkaa arvoketjuista ja keskittymällä luomaan arvoa asiakkaalle. Arvoketjut muodostuvat kaikista niistä yrityksen toiminnoista ja aktiviteeteista, jotka osallistuvat tuotteen, palvelun tai informaation tuottamiseen asiakkaalle. Arvoketjussa olevat aktiviteetit voidaan jakaa kolmeen kategoriaan: Arvoa tuottavaan, arvoa tuottamattomaan, ja pakolliseen mutta arvoa tuottamattomaan. Arvo määritellään asiana, josta asiakas on valmis maksamaan.

Arvoa tuottamattomia aktiviteettejä kutsutaan hukaksi. Tällaiset aktiviteetit pitäisi minimoida, ja mikäli mahdollista, poistaa kokonaan. Hukka voidaan jakaa kolmeen kategoriaan: epätasapainoon, ylikuormitukseen ja hukkaan. Epätasapainolla tarkoitetaan jonkin suureen vaihtuvuutta missä tahansa toiminnossa, joka ei tuo arvoa asiakkaalle. Tällaisia voi olla esimerkiksi tuotannon volyyymi ja laatu vaihtelut. Ylikuormitus tarkoittaa sanansamukaisesti työntekijöiden tai välineiden liian suurta työkuormaa. Viimeinen kategoria eli hukka kattaa kaikki arvoketjun aktiviteetit, jotka eivät ole pakollisia eivätkä tuota lisäarvoa asiakkaalle. Tavallisesti hukka jaetaan edelleen seitsemään alakategoriaan. Kategorioiden

avulla voidaan helpottaa hukan tunnistamista arvoketjusta. Seuraavassa kuvassa on esitetty Lean IT:n mukaisesti nämä seitsemän kategorian ja kuinka niiden ilmeneminen eroaa tuotanto-, toimisto- ja IT-ympäristöissä.

Waste	Manufacturing	Office	IT
Inventory	Excess raw material, work in process, and finished goods	Work accumulating in physical and virtual inboxes, excess physical files	Excessive information causing searching and version control problems, excessive backlog and work in progress
Overproduction	Producing more or sooner than the customer needs	Producing documents before they are needed, doing a large batch of paperwork that the downstream worker can't handle all at once	Excessive email, reports, system alerts, etc., that are not read or acted upon
Delays	Waiting for materials from the upstream workcenter	Waiting for unnecessary signature or approval or monthly closing processes, time wasted in unproductive meetings	System downtime, unnecessary workflow steps
Transportation	Moving materials from one location to another	Physical or virtual transportation of documents	Handoff of information across organization boundaries and multiple systems, security barriers to information flow
Over processing	Doing more work than the customer wants (e.g., unnecessary product features)	Producing reports that nobody reads, overuse of the email "reply all" button, expediting	Redundant data, unnecessary transactions, unused reports, software features that users don't need
Motion	Bending, lifting, twisting, walking, packing/unpacking, searching for stuff	Walking, copying, filing, searching for materials and information	Searching for information, reentering data, excessive keystrokes, frequently shifting priorities
Defects	Defects that require correction or cause scrap	Work passed on that is incomplete, incorrect, or requires clarification	Incorrect, untimely, and confusing information that causes bad decisions

FIGURE 3.3
Comparing waste in manufacturing, office, and IT.

Kuva 16. Hukan muodot eri toimintaympäristöissä (Bell ja Orzen, 2010)

Kuvassa 16 olevassa taulukosta voidaan nähdä, että vaikka hukan kategoriat pysyvät samana, ilmenevät ne eri tavoin eri liiketoimintaympäristöissä. Kuvassa näkyvät ilmenemismuodot ovat kuitenkin vain esimerkkejä, joiden tarkoituksena on helpottaa hukkien tunnistamista arvoketjuista.

Lean-ajattelussa keskeistä on jatkuva kehitys, johon osallistuu koko organisaatio. Jatkuvaa kehitystä ylläpidetään muun muassa erilaisten Kaizen-tapahtumien avulla. Kaizen-tapahtumat voivat olla esimerkiksi lyhyitä säännöllisiä tapaamisia tai yksittäisiä nopeita kehitysprojekteja. Kaizen voidaan jakaa kahteen luokkaan: järjestelmä ja prosessi Kaizen.

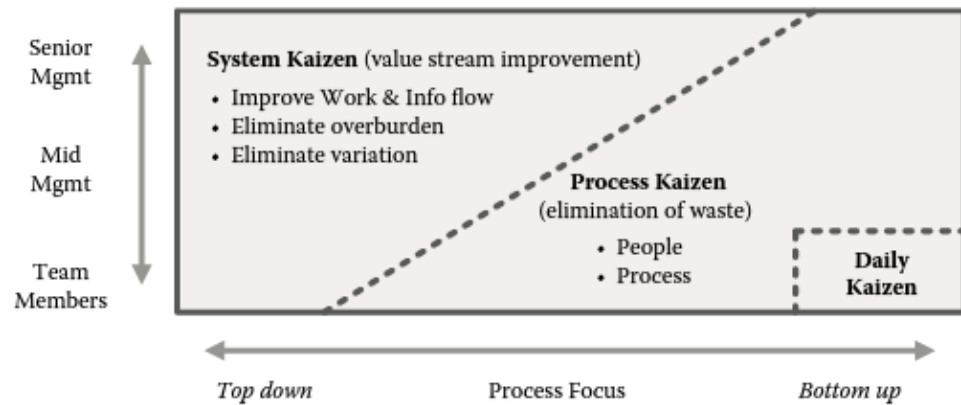


FIGURE 2.9
Kaizen types

Kuva 17 Kaizen-tyypit (Bell ja Orzen, 2010)

Kuvan 17 mukaisesti järjestelmä Kaizen keskittyy työ- ja informaatiovirran kehitykseen ja sisältää epätasapainon ja ylikuormituksen poistamisen. Päävastuu järjestelmä Kaizenista on johdolla ja päälliköillä. Itse hukan poistaminen tapahtuu prosessi Kaizenin avulla. Prosessi Kaizen kuuluisi olla pääasiassa prosessiin osallistuvien työntekijöiden vastuulla. Kuvassa merkattu päivittäinen Kaizen on osa prosessi Kaizenia. Päivittäisessä Kaizenissa suoritetaan jokin kehitystoimi hyvin nopealla aikataululla yhden päivän aikana.

Jotta jatkuva kehitys ja jatkuvat Kaizen-aktiviteetit olisivat mahdollisia, täytyy yrityksessä olla oikeanlainen kulttuuri. Lean-kulttuuriin kuuluu vahvasti työntekijöihin luottaminen ja heidän kunnioittamisensa. Lean-ajattelussa henkilöstöä pidetään yrityksen suurimpana voimavarana. Ilmapiirin tulisi kannustaa kehitykseen ja ottaa kaikkien ihmisten kehitysehdotukset huomioon. Kehityksen tulisi tapahtua siellä missä työ tehdään, ja kehitysideoiden virrata näin ollen alhaalta ylöspäin. Tällöin johdon tehtäväksi jää oikeanlaisen kulttuurin ja ilmapiirin ylläpitäminen ja suunnannäyttäminen. Kehityksen tulisi keskittyä ongelma prosesseihin ei ongelma ihmisiin. Henkilöstöä tulisi kannustaa ensin yksinkertaistamaan ja sitten mahdollisuuksien mukaan automatisoimaan.

Tavallisesti kehitys tapahtuu niin kutsutuissa PDCA (Plan, Do, Check, Act) -sykleissä. Plan vaiheessa tunnistetaan kehityskohde, Do-kohdassa kokeillaan muutosta kehityskohteeseen, Check-vaiheessa tarkastetaan muutoksien vaikutus ja viimeisenä Act-kohdassa tehdään muutokset Check-vaiheessa tehtyjen havaintojen pohjalta. Jos havaitut tulokset olivat positiivisia, otetaan muutokset käyttöön ja palataan Plan-vaiheeseen pyrkien löytämään uusi kehityskohde. Mikäli tulokset eivät vastanneet odotuksia palataan Plan-vaiheeseen tekemättä muutoksia, ja pyrkien löytämään uusi tapa ongelman ratkaisuun.

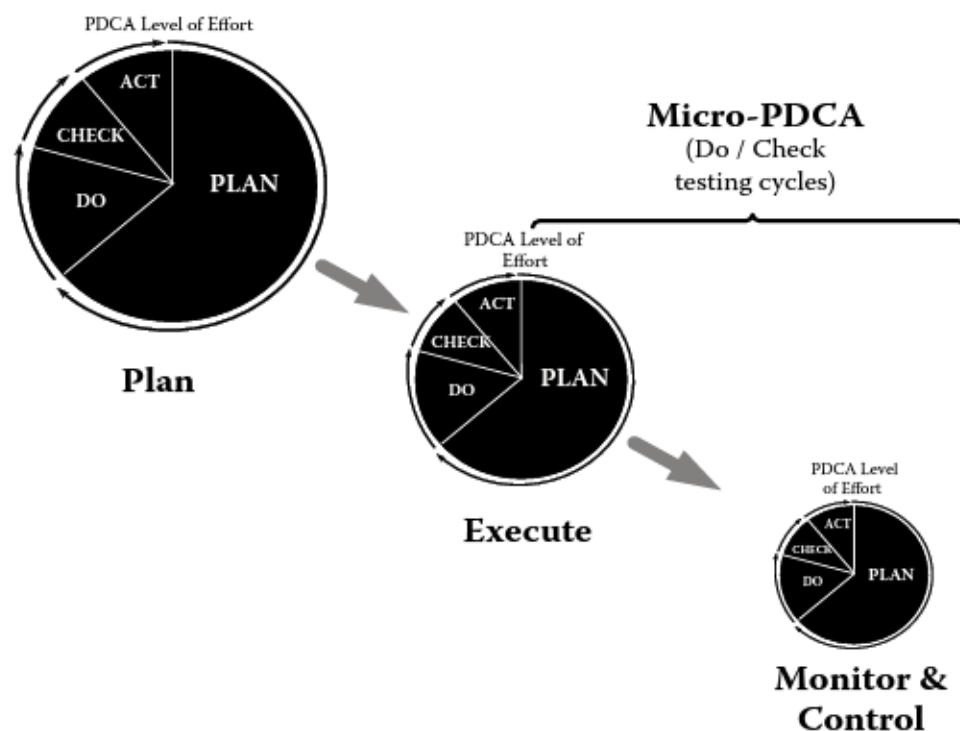


FIGURE 9.10
PDCA applied at three phases of the project life cycle.

Kuva 18 PDCA-Syklit (Bell ja Orzen, 2010)

PDCA-syklit ovat jatkuvan kehityksen ydin. Usein yksi PDCA-sykli voi sisältää useita pienempiä PDCA-syklejä kuvan 18 mukaisesti. Monet pienet PDCA-syklit muodostavat esimerkiksi Agile-mallin ytimen, jossa monia nopeita kehityssyklejä (iteraatiota) suoritetaan peränjälkeen. Tämä mahdollistaa suunnitelmien

muuttamisen nopeasti, jos esimerkiksi asiakkaan tarpeet muuttuu yllättäen. (Bell ja Orzen, 2010.)

4.2.2 Lean-työkalut

Lean-organisaation luomiseen, ylläpitämiseen ja jatkuvan kehityksen tukemiseen on olemassa erilaisia Lean-työkaluja. Työkalut vaihtelevat ohjenuoran omaisista metodeista erilaisiin dokumentteihin ja muistisääntöihin. Yksiä tunnetuimpia Lean-työkaluja ovat: A3-ajattelu, Gemba-kävelyt, arvovirtakuvaus, vakioidu työ, ja 5S-menetelmä. Näiden esimerkkien lisäksi erilaisia työkaluja on huomattava määrä. Jotkin työkalut ovat yleispäteviä ja sopivat moneen eri tilanteeseen, kun toiset työkalut ovat tarkoitettu johonkin tiettyyn kehitysaktiviteettiin. Lean IT suositteliekin ensin omaksumaan muutaman työkalun käytön ja tarvittaessa lisäämään työkaluvalikoimaa.

Lean IT nostaa tärkeimmäksi Lean työkaluksi A3-ajattelun. A3-ajattelu perustuu ajatukseen, että yhden ongelman ratkaisuun vaadittavat kaikki elementit tulisi mahtua yhdelle A3-sivun kokoiselle paperille. Tämän paperin tulisi sisältää ongelman kuvaus ja sen osat. Paperia täyttäessä analysoidaan ongelman juurisyy, tehdään suunnitelma ongelman korjauksesta ja hypoteesit korjauksien vaikutuksesta. A3-ajattelun tarkoituksena ei kuitenkaan ole vain täyttää paperia vaan ohjata ajattelu oikeisiin paikkoihin kuten ongelman juuri syyhyn. Kun A3-ajattelu on omaksuttu organisaatiossa ei paperia välttämättä edes tarvita, vaan A3-ajattelu tulee työntekijöille luonnostaan.

Arvovirtakuvauksella tarkoitetaan yksittäisten arvoketjujen pilkkomista palasiin ja visualisoimalla palaset prosessikuvauksen tyyppisenä ketjuna. Arvoketjun visualisointi auttaa kaikkia arvoketjuun osallistuvia tahoja ymmärtämään kokonaiskuvaa paremmin, ja tunnistamaan arvoa tuottamattomat aktiviteetit muiden joukosta. Tavallisesti arvovirtakuvaukseen merkitään jokaisen aktiviteetin käyttämä työ- ja jonotusaika sekä laatu- tai virheprosentti.

Vakioidulla työllä pyritään määrittämään yhtenäinen työskentely tapa jollekin työtehtävälle tai prosessille kaikkien osallisten kesken. Ensin työskentely tavat määritetään ja sitten dokumentoidaan. Kaikkien osallisten tulisi sitoutua noudattamaan tätä määritettyä työskentelytapaa. Jokaisen kehityssyklin jälkeen dokumentoitu vakiotyö tulisi tarkastaa ja tarvittaessa päivittää. Vakiotyön tarkoituksena on vähentää variaatiota ja sen myötä hukkaa arvoketjusta. Vakiotyö myös mahdollistaa nopean kehityksen ja ketteryyden. Kun työ on vakioitu, pystytään epäkohdat ja hyvät käytännöt tunnistamaan.

5S-menetelmä on organisointi metodi, joka koostuu viidestä vaiheesta: sortteeraus, systematisointi, siivous, standardisointi ja seuranta. Sortteerauksessa kaikki turha pyritään tunnistamaan ja poistamaan, systematisointi vaiheessa selkeytetään järjestystä tai toimintamalleja, joiden mukaan sortteerauksesta jäljelle jääneet kohteet organisoidaan. Siivouksella tarkoitetaan jatkuvaa asioiden järjestämistä systematisointi vaiheessa määritettyjen tapojen mukaisesti. Standardisoinnissa määritetään yhteiset parhaat käytännöt, joilla järjestystä pidetään yllä. Viimeisenä seurannalla sanan mukaisesti seurataan, että standardoituja tapoja noudetaan ja järjestys pysyy yllä. 5S-menetelmän hyödyt eivät rajoitu asioiden organisointiin, vaan sen avulla voidaan myös tunnistaa hukkia, kehityskohteita ja epäjärjestyksen alla piileviä ongelmia. (Bell ja Orzen, 2010.)

4.3 Agile-sovelluskehitys

Tässä luvussa tutustutaan lyhyesti Agile-sovelluskehitykseen ja sen sisältämiin metodeihin.

Agile on Leanin periaatteita hyödyntävä metodologia sekä yksi sovelluskehityksessä käytetyimmistä metodologioista (Laakkonen, 2014: 9-11). Agile-metodologioiden käyttämisellä on havaittu olevan positiivinen vaikutus projektien onnistumiseen, tehokkuuteen ja sidosryhmien tyytyväisyyteen (Serrador ja Pinto, 2015).

Vaikka Agile-metodologioita ei ole laajasti omaksuttu parhaisiin käytäntöihin RPA:n kehityksessä, on Agilen hyödyntämisestä RPA:n kehityksessä ja ylläpidossa mahdollisesti hyötyä. Agilen hyödyntäminen RPA:n kehityksessä ja ylläpidossa voi joissain tapauksissa olla jopa keskeisessä asemassa onnistuneessa RPA-projektissa. (Philipp, 2021.)

Agile-käsite on syntynyt vuonna 2001, kun 17 Agilen puolestapuhujaa julkaisivat Agile-manifestin. Manifestissa määritettiin Agilen arvot, joissa arvotetaan enemmän:

- yksilöitä ja vuorovaikutusta, kuin prosesseja ja työkaluja
- toimivaa sovellusta, kuin kokonaisvaltaista dokumentaatiota
- yhteistyötä asiakkaiden kanssa, kuin sopimus neuvotteluita
- muutokseen reagoimista, kuin suunnitelman noudattamista.

Agile-termi sisältää monia erilaisia sovelluskehitystekniikoita kuten scrum ja extreme programming. Monet näistä tekniikoista pohjautuvat erilaisiin Lean-työkaluihin ja metodeihin. (Bell ja Orzen, 2010.)

Agile-metodeilla pyritään ensisijaisesti nopeuttamaan arvon luontia asiakkaalle, ja poistamaan turhaa työtä eli hukkaa sovelluskehityksestä. Tavanomaisen sovelluskehityksen haasteina on havaittu olevan varsinkin ylituotanto ja virheiden korjaamiseen käytetty suuri aika. (Bell ja Orzen, 2010.)

Ylituotantoa tapahtuu esimerkiksi, kun suunnitteluvaiheessa ylimitoitetaan sovelluksen tarpeet ja sen sisältämät eri ominaisuudet. Asiakas ei välttämättä aina itse ymmärrä, mitä hän sovellukselta oikeastaan tarvitsee. Ylituotantoa syntyy myös, kun asiakkaan tarpeet tai prosessit muuttuvat sovelluskehityksen aikana. (Bell ja Orzen, 2010.)

Tavanomaisessa vesiputousmallissa testaus, jossa virheet voidaan havaita, tapahtuu aivan kehityksen lopuksi ennen tuotantoon vientiä. Kun virheitä ei korjata kehityksen edistyessä, siirtyvät ne työvaiheesta toiseen ja hidastavat kehitystä. Suuresta kokonaisuudesta virheiden korjaaminen on hidasta, sillä yksittäisen virheen korjaaminen voi aiheuttaa lisää virheitä toisaalla. (Bell ja Orzen, 2010.)

Agile-mallilla voidaan vähentää ylituotannon ja virheiden vaikutusta työmäärään suorittamalla kehitys lyhyissä iteraatioissa. Jokainen iteraatio sisältää PDCA-syklin mukaiset vaiheet. Jokaisen syklin päätteeksi saadaan toimiva sovellus, joka jatko kehittyy jokaisen seuraavan syklin aikana. Kehitys kannattaa aloittaa asiakkaalle eniten arvoa tuottavista ominaisuuksista, jonka jälkeen muita ominaisuuksia voidaan lisätä seuraavissa kehitysiteraatioissa. (Bell ja Orzen, 2010.)

Lyhyet iteraatiot mahdollistavat suunnitelmien muuttamisen kesken kehityksen. Mikäli asiakkaan tarpeet muuttuvat, voidaan uudet tarpeet ottaa huomioon jo seuraavassa iteraatiossa. Iteraatioiden ansioista ylituotannon mahdollisuus pienenee, kun asiakas voi ensimmäisten iteraatioiden aikana havaita saaneensa jo kaiken tarpeellisen. Kun virheet korjataan jokaisen iteraation päätteeksi, on myös virheiden korjaus nopeaa, eivätkä virheet pääse siirtymään kehityksessä eteenpäin ja leviämään sovelluksen eri osiin. (Bell ja Orzen, 2010.)

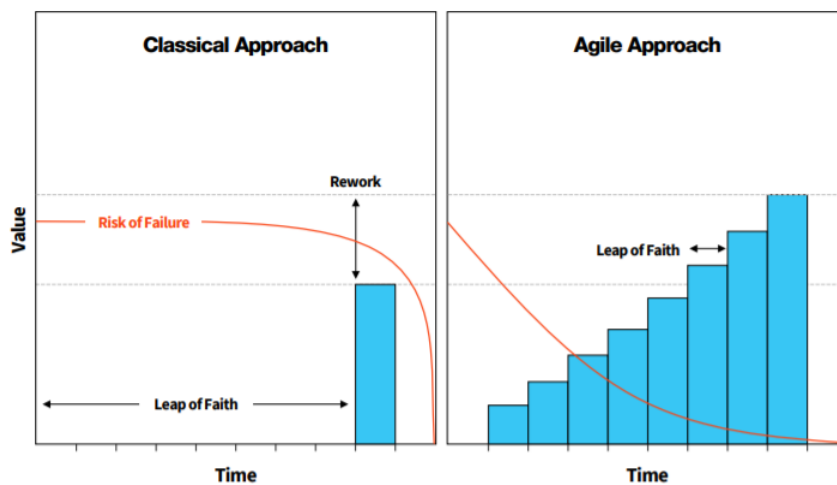
Kommunikaatio kehittäjien kesken sekä sidosryhmien välillä on tärkeää Agile-kehityksessä. Yhteydenotot pitäisi kuitenkin pitää lyhyinä ja välittöminä ja voivat tapahtua esimerkiksi jonkin pikaviestimen kautta. Kehittäjien tulisi myös käydä säännöllisesti paikan päällä asiakkaiden luona, missä itse työ tapahtuu. Näin voidaan varmistua, että asiakkaan tarpeista on hyvä ymmärrys läpi koko sovelluskehityksen. (Bell ja Orzen, 2010.)

Arvon luonti asiakkaalle on keskeisessä asemassa Agile-sovelluskehityksessä. Jotta arvon luonti asiakkaalle olisi mahdollista, täytyy ensin ymmärtää, kuka asiakas on. Sovelluskehityksessä ei usein ole itsestään selvää, ketkä projektin

asiakkaat ovat. Projektilla voikin olla useita asiakkaita kuten sovelluksen käyttäjä, sovelluksen käyttäjän asiakkaat, tai sovelluksen käyttäjän jokin muu sidosryhmä. (Bell ja Orzen, 2010.)

Arvoa luodaan asiakkaalle vastaamalla asiakkaan tarpeisiin. Myös asiakkaan tarpeiden määrittämisessä tulee olla tarkkana. Asiakkaalla voi usein olla vahvoja mielipiteitä tarpeistaan ilman selkeää ymmärrystä kokonaiskuvasta tai uutta sovellusta hyödyntävästä prosessista. Asiakkaan käsitys perustuu usein vanhaan prosessiin, joka hyödyntää vanhaa tekniikkaa. Asiakaskunnan edustajilla voi myös olla keskenään erilainen näkemys siitä, millainen lopputulos tulisi olla. (Bell ja Orzen, 2010.)

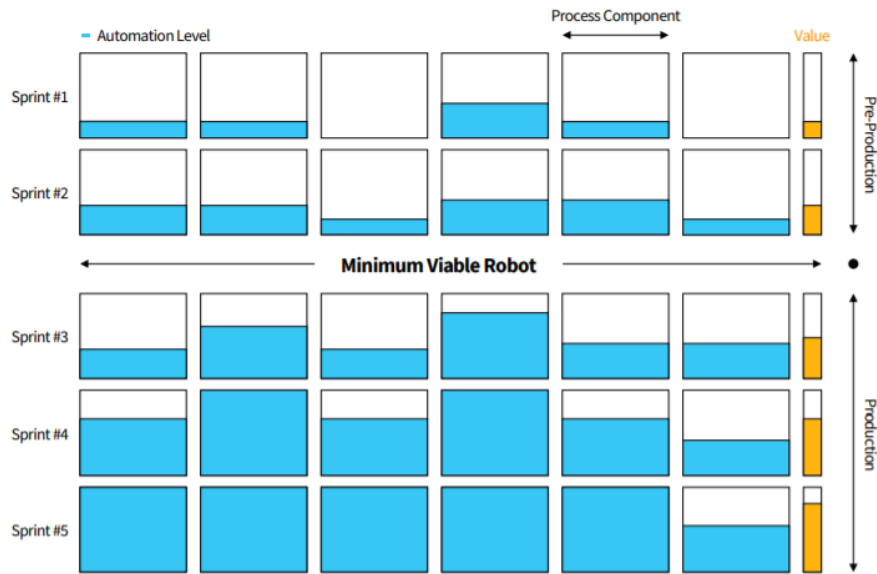
Myös ohjelmistorobotin kehityksessä voidaan Agile-metodien avulla pienentää kehitysprojektin riskejä ja nopeuttaa arvon luontia asiakkaalle. (Philipp, 2021.)



Kuva 19 Tavanomaisen ja Agile-mallin erot (Philipp, 2021)

Kuvassa 19 nähdään tavanomaisen RPA-kehityksen ja Agile-mallin erot asiakkaalle muodostetun arvon ja projektin riskien kehityksestä projektin edetessä. Agile-mallin avulla riskit pienenevät nopeasti iteraatioiden mukana ja arvoa saadaan luotua asiakkaalle jo ensimmäisistä iteraatioista lähtien. (Philipp, 2021.)

Agile-mallin hyödyt tulevat esiin, kun ohjelmistorobotilla tehtävät automaatiot pilkotaan komponentteihin.



Kuva 20 Esimerkki ohjelmistorobotin kehityksestä sprinteissä (Philipp, 2021)

Kuvassa 20 nähdään esimerkki, kuinka useamman komponentin kehitys edistyy iteraatioissa ja kuinka asiakkaalle luotu arvo kasvaa iteraatioiden edetessä. Esimerkissä pyritään tuomaan robotti nopeasti käyttöön eri prosessin osiin. Näin on mahdollista saada prosessin osallisille nopeasti käsitys automaation vaikutuksista ja parantaa heidän käsitystään omista tarpeistaan. Näin on mahdollista saada arvokasta palautetta heti projektin alussa ja näin tarkennettua projektin tarpeita ja kriteereitä ennen jatkokehitystä. (Philipp, 2021.)

4.4 Yhteenveto parhaista käytänteistä

Ohjelmistorobotin kehityksessä on tärkeää ymmärtää automatisoitavat työtehtävät tarkasti. Automatisoitavan prosessin kehittämistä ja optimointia pidetään yleisesti tärkeänä osana ennen ohjelmistorobotin avulla tapahtuvaa automaatiota. Prosessin kehityksessä voidaan soveltaa useita eri ajattelumalleja ja viitekehyksiä, kuten Lean-ajattelua. Erityisen tärkeää robotin kannalta on, että automatisoitava prosessi olisi mahdollisimman vakioitu ennen automatisointia. Myös

hukan poistosta ennen automaatioita voi olla hyötyä, vaikka päinvastaisia mielipiteitä on myös olemassa.

Ohjelmistorobotin kehitykseen liittyvät parhaat käytännöt perustuvat pitkälti sovelluskehityksessä hyödynnettyyn vesiputousmalliin. Sovelluskehityksessä Agile-metodeilla on kuitenkin havaittu parempia tuloksia kuin tavanomaisella vesiputousmallilla. Myös RPA:n kehityksessä on havaittu positiivisia kokemuksia Agile-mallin käytöstä.

Ohjelmistorobotiikan yksi sen suurimpina pidetyistä vahvuuksista on sen skaalautuvuus. Jatkuva kehitys on keskeisessä roolissa ohjelmistorobotiikan skaalautumisen mahdollistamisessa. Jatkuvan kehityksen malli on suositeltavaa omaksua myös muualla organisaatiossa kuin vain ohjelmistorobotin kehityksessä. Jatkuva kehitys on myös Lean-ajattelun keskiössä, ja Lean tarjoaakin kattavasti työkaluja jatkuvan kehityksen ylläpitämiseen organisaatiossa.



Kuva 21 Kirjallisen viitekehityksen kytkeytyminen opinnäytetyöhön

Kuvassa 21 on esitetty teoreettisen viitekehyksen osat ja niiden kytkeytyminen tämän projektin eri vaiheisiin sekä nykytila-analyysin aikana havaittuihin haasteisiin. Ohjelmistorobotiikan parhaita käytäntöjä hyödynnetään ohjelmistorobotin kaikissa elinkaaren vaiheissa. Parhaat käytännöt myös selvittävät prosessin kehityksen tarpeellisuutta ja metodeja ohjelmistorobottia varten. Lean IT sekä sen sisältämät työkalut tukevat prosessin kehitystä, ja nykytila-analyysin aikana prosessissa havaittujen haasteiden ratkaisemista. Lean-ajattelua voidaan myös hyödyntää ohjelmistorobotin jatkokehityksessä. Agile-metodeilla pyritään tukemaan ohjelmistorobotin kehitystä aikaisemmin selvitettyjen parhaiden käytäntöjen lisäksi.

5 Ohjelmistorobotin käyttöönotto ja kehitys

Edellisessä luvussa selvitimme ohjelmistorobotiikkaan ja prosessin kehitykseen liittyviä parhaita käytäntöjä. Tässä esitetään ohjelmistorobotin käyttöönoton ja kehittämisen sisältävät työvaiheet. Lukuun sisältyy myös automatisoitavien prosessien optimointi ennen ohjelmistorobotin käyttöönottoa.

5.1 Prosessien optimointi

Nykytila-analyysin aikana havaittiin prosesseissa 4 automatisaation kannalta ongelmallista osaa. Prosesseissa havaittiin myös muita heikkouksia automaatioon liittyvien haasteiden lisäksi. Osa näistä heikkouksista saadaan ratkaistua itse automatisaation avulla, kun toiset on ratkaistava esimerkiksi Lean-menetelmiä hyödyntäen.

5.1.1 Työvaiheiden standardisointi

Nykytila-analyysissä havaittiin, etteivät työvaiheet ole standardisoituja ja työnteo-
kijöillä on erilaisia ohjeistuksia työvaiheiden suorittamisesta. Jotta robotin kehitys olisi mahdollista, täytyy oikeat työtavat määrittää.

Työvaiheiden standardisointiin voidaan hyödyntää Leanin mukaista vakiotyötä. Työn vakioimisessa voidaan Leanin vakiotyön mukaisten periaatteiden lisäksi hyödyntää 5S-menetelmää.

Koska prosesseilla ei ollut selkeää omistajaa, suoritettiin työtehtävien standardointi yhdessä yrityksen kehityspäällikön kanssa. Ensin nykytila-analyysin aikana havaitut työvaiheet listattiin, jonka jälkeen ne käytiin kehityspäällikön kanssa läpi 5S-menetelmää hyödyntäen.

5S-menetelmän mukaisesti ensin työvaiheille suoritettiin sortteeraus, jossa turhat työvaiheet tunnistettiin ja poistettiin. Tämä tehtiin kyseenalaistamalla jokainen työvaihe ja sen aiheuttama hyöty prosessille. Mikäli kehityspäällikkö ei ollut

varma työvaiheen tarkoituksesta, selvitettiin se jonkun työntekijän tai vastuuhenkilön kanssa sähköpostin välityksellä. Menetelmän avulla voitiin esimerkiksi todeta, ettei tilauksen käsittelyn aikana tehtyä vikakoodi luokitusta hyödynnetty myöhemmin arvoketjussa. Vikakoodin määrittäminen todettiin näin olevan ylituotantoa ja poistettiin prosessista.

Sortteerauksen jälkeen työvaiheet systematisoitiin. Systematisoinnissa määritettiin, kuinka jäljelle jääneet työvaiheet tulisi suorittaa. Vaiheen aikana määritettiin, mihin mikäkin tieto tilauksen käsittelyssä tulisi merkata ja mitä tietojen tulisi pitää sisällään. Mikäli tietue oli tärkeä ajajärjestelyn kannalta, kysyttiin asiasta vielä kuljetusesimiehen mielipide asiaan. Tämän jälkeen määritetyt työskentelytavat kirjattiin ja ohjeistettiin työntekijöille.

Päivittäisen siivouksen eli työskentely tapojen noudattaminen jäi työntekijöiden vastuulle. Oikeiden työskentelytapojen noudattamista pyrittiin helpottamaan tekemällä pieniä muutoksia järjestelmään. Järjestelmä muutoksissa tuotiin muutamia tietueita paremmin näkyviin Sherpan tilausnäkyymään, jotta tietojen syöttäminen olisi helpompaa ja ne tulisivat oikeaan paikkaan. Suurempia muutoksia ei tässä vaiheessa järjestelmään tehty, koska tietojen syöttäminen oli tarkoitus automatisoida ohjelmistorobotilla.

Viimeisen eli seurantavaiheen implementointi oli tässä vaiheessa hankalaa, sillä prosesseilla ei vielä ollut selkeää omistajaa. Kehityspäällikölle kuitenkin ehdotettiin määrittämään prosesseille omistaja, joka ottaisi seurannan vastuulleen. Seuranta varten voitaisiin esimerkiksi perustaa PowerBI-raportti, joka seuraa väärään sijaintiin merkittyjen tietojen esiintyvyyttä. Ohjeistuksen ylläpidossa ja yhtenäisenä pitämisessä on kuitenkin vielä tärkeämpää hyvä kommunikaatio työntekijöiden kesken. Myös jatkuvan kehityksen malli, jossa sovittuja työtehtäviä päivitetään kehityksen mukana, voi auttaa yhtenäisten työskentelytapojen ylläpitämisessä. Tällöin määritetyt työtapa- ja käytännöt käydään työntekijöiden kanssa läpi jokaisessa kehityssyklissä.

Prosessin omistajalla olisi myös tärkeä vastuu informoida prosessiin ulkopuolelta tulevista muutoksista työntekijöitä ja tehdä tarvittavat muutokset ohjeistukseen. Tällaisia voisi olla muun muassa järjestelmään tulevat muutokset, jotka vaikuttavat prosessin sisältämien työtehtävien suorittamiseen.

5.1.2 Automatisaatioon liittyvien haasteiden ratkaisu

Nykytila-analyysissä havaittiin neljä haastetta automatisaation kannalta. Näistä ensimmäinen liittyi oikean laskutusasiakkaan valintaan. Laskutusasiakkaan valinnan kannalta ongelmallista oli suuri määrä duplikaattitietoja ja virheellisten tietueiden esiintyvyys muiden tietueiden joukossa. Haasteita lisäsi automatisaation kannalta se, että laskutusasiakas oli toisinaan osattava valita useamman eri vaihtoehdon kesken. Haastetta on kuvattu tarkemmin nykytila-analyysissä.

Ongelman ratkaisemiseksi asiakaspuu siivottiin ja organisoitiin 5S-menetelmällä. Sortteeraus aloitettiin kartoittamalla jokainen kohde, johon ei liittynyt yhtäkään tilausta, ja ne poistettiin. Oletuksena oli, että jokainen tilaus, jossa on havaittu virheellinen kohde tietue, on korjattu. Näin saatiin poistettua 45 prosenttia kaikista järjestelmässä olevista kohteista. Seuraavaksi järjestelmästä haettiin kaikki kohteet, joihin ei liittynyt tilausta viimeisen kahden vuoden ajalta. Näistä tietueista poistettiin kohde kategorisointi, jolloin ne eivät tule näkyviin valitessa laskutusasiakasta tilaukselle. Tietue olisi kuitenkin helposti otettavissa tarvittaessa takaisin käyttöön lisäämällä kohde kategorisointi tietueelle takaisin. Sortteerauksen seurauksena kohteiden määrä oli enää noin 40 prosenttia alkuperäisestä.

Systematisoinnin ei koettu olevan tässä vaiheessa järkevää. Tietueet olivat kategorisoitavissa selkeästi asiakastyypin ja pääasiakkaan mukaan, joka koettiin tässä vaiheessa riittäväksi. Asiakaspuurakenteeseen ei haluttu tehdä muutoksia tässä vaiheessa.

Seuraavaksi työntekijät ohjeistettiin aina virheellisen tietueen havaitessaan poistamaan tietueelta kohde kategorisointi ja poistamaan tietue, mikäli siihen ei liity yhtäkään tilausta. Tämä myös kirjattiin aikaisemmin dokumentoituun tilausten käsittelyn vakiotyöhön. Sortteerauksen mukainen tilauksettomien kohteiden poistaminen järjestelmästä määritettiin tehtäväksi yhden kuukauden välein. Myös tämän avulla voitiin seurantaa suorittaa. Myös robotin tekemät virhemäärät asiakkaan valinnassa voidaan tulevaisuudessa ottaa osaksi seurantaa.

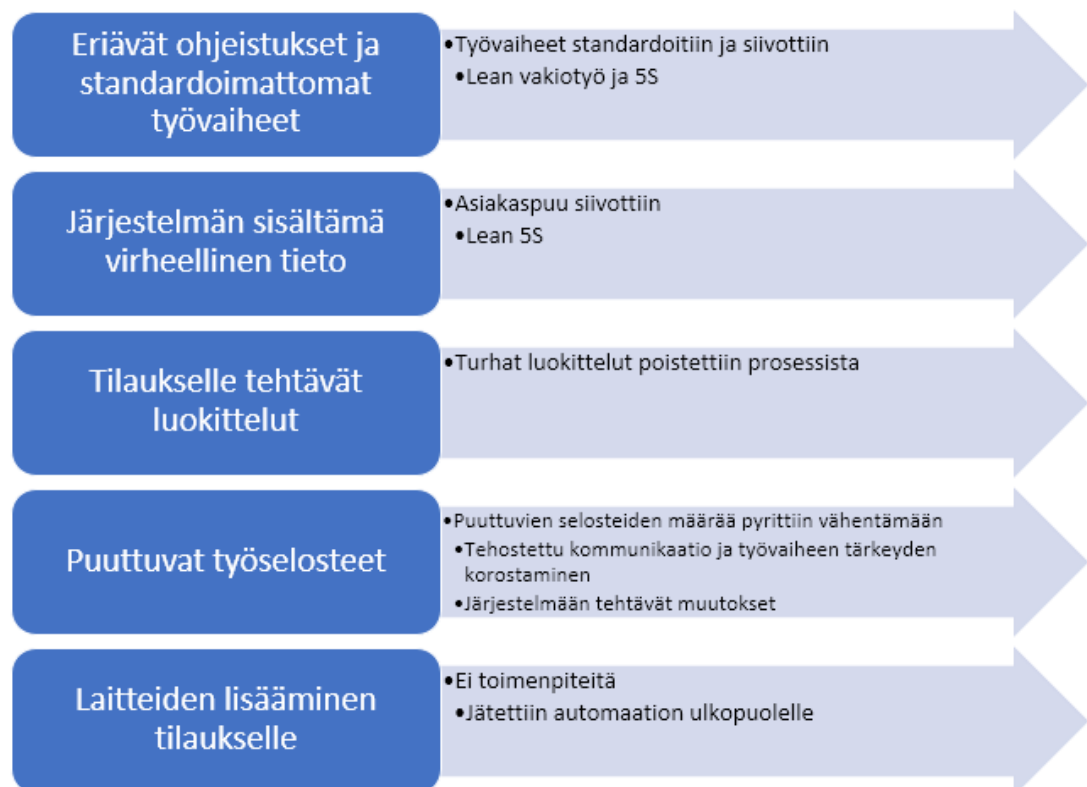
Toisena haasteena oli tilaukselle tehtävät luokittelut, vikakoodi ja tilaustyyppi, jotka tehtiin pääsääntöisesti tilauksella olevan työ kuvauksen pohjalta. Näistä kahdesta vikakoodiluokitus poistettiin työvaiheista kokonaan jo työtehtävien standardisointivaiheessa. Tilaustyyppi oli pakollinen tietue tilaukselle, eikä luokittelu vaihetta voitu poistaa. Päätettiin, että tilaustyyppi luokittelu pyritään ottamaan osaksi ohjelmistorobotin automatisointia.

Kolmas haaste liittyi työselosteiden puuttumisiin tilauksilta. Yhtenä vaihtoehtona oli muuttaa työseloste pakolliseksi kentäksi tilausta kuitattaessa, jolloin kuljettajien tai työn suorittajien olisi pakko kirjoittaa työseloste. Tällaista muutosta ei kuitenkaan haluttu vielä järjestelmään toteuttaa. Työselosteiden osalta kommunikointia tietueen tärkeydestä tehostettiin, ja kuljetusesimiehet veloitettiin muistuttamaan työsuorittajia asiasta ja puuttumaan puuttuviin selosteisiin. Järjestelmässä oli myös olemassa jonkin verran niin kutsuttuja vakioselosteita. Vakioselosteen valitessaan järjestelmä kirjoitti vakioselosteelle määritetyn vakioidun tekstin. Vakioselosteita oli saatu positiivisia kokemuksia, mutta niitä oli vähän, eivätkä ne soveltuneet kuin tietyntylaisiin toimituksiin. Näitä vakioselosteita päätettiin jatkokehittää, jotta niiden käyttöä voitaisiin lisätä ja näin helpottaa työselosteen kirjoittamista. Työselosteiden kehitys vaati kuitenkin muutoksia järjestelmään, eikä kehityksen suorittaminen vielä projektin aikana ollut mahdollista. Puuttuneiden työselosteiden määrää päätettiin alkaa seurata automatisaation valmistuttua. Seurannan pohjalta tarkennettaisiin jatkotoimenpiteiden tarpeellisuus. Automaation osalta tilaukset, joista seloste puuttui, päätettiin jättää automatisaation ulkopuolelle.

Viimeinen haaste liittyi tilauksiin, jotka sisälsivät myytävän laitteen. Laitteen valinta vaati kokemusta eri kumppanista ja laitteiden koodeista. Oikeaa laitetta ei ollut mahdollista loogisesti päätellä luotettavasti kumppanin järjestelmästä saatavilla tiedoilla. Laitteiden merkitseminen tilaukselle päätettiin tässä vaiheessa jättää kokonaan automatisaation ulkopuolelle.

5.1.3 Yhteenveto prosessinkehityksestä

Edellisissä vaiheissa suoritettiin optimointia ja standardisointia automatisaation piirissä oleviin prosesseihin ja työvaiheisiin. Tehtävillä toimenpiteillä pyrittiin mahdollistamaan ohjelmistorobotin käyttöönotto, ja parantamaan sillä suoritettun automatisaation määrää. Työvaiheiden standardisoinnin odotetaan myös helpottavan prosessien kehitystä tulevaisuudessa ja tehostaman myös automatisaation ulkopuolista työtä. Ongelmakohtat ja niihin tehdyt muutokset on esitetty kuvassa 22.



Kuva 22 Prosessien haasteet ja tehdyt muutokset

Tehdyt muutokset koettiin riittäviksi, jotta ohjelmistorobotti voidaan ottaa käyttöön. Nopea automatisaation käyttöönotto koettiin tässä kohtaa järkevämmäksi kuin prosessin jatkokehitys.

5.2 Ohjelmistorobotin käyttöönotto ja kehitys

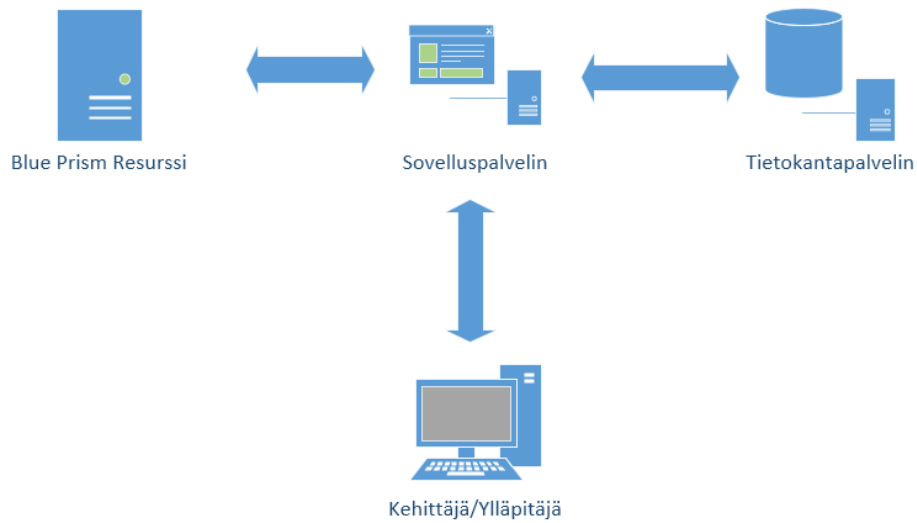
Tässä luvussa esitetään robotin käyttöönottoon ja kehitykseen liittyvät vaiheet. Kehitys vaiheen aikana kumppani 1:n järjestelmään suoritettavat automatisaatiot jouduttiin jättämään tämän työn ulkopuolelle. Tämän aiheutti kumppanin tarkentuneet tietoturvamääritykset, jotka aiheuttivat robotin käyttöön tulevien käyttäjätunnusten saamisen viivästymisen.

5.2.1 Ohjelmistorobotin käyttöönotto

Ohjelmistorobottia tarjotaan erilaisina pilvi-, SaaS- (Software as a Service) sekä On premises -ratkaisuina. Pilvi- ja SaaS-ratkaisuissa ohjelmistorobotti sekä sen tarvitsema infrastruktuuri on asennettuna pilvessä tai palveluntarjoajan ympäristössä. Tällaisissa ratkaisussa erillinen käyttöönotto ei ole tarpeellista. Tässä työssä päädyttiin On premises -ratkaisuun, jossa ohjelmistorobotin käyttämä infrastruktuuri ja palvelimet ovat ohjelmistorobotin tilaajan itse järjestettävä.

Toimiakseen ohjelmistorobotti vaatii minimissään tietokoneen, johon ohjelmistorobotin ohjelmisto asennetaan. Kun ohjelmisto on asennettu voi ohjelmistorobotti ohjata tietokonetta ja suorittaa sille määritetyt prosessit ja työvaiheet. Tietokoneen lisäksi robotti tarvitsee SQL-tietokannan ja palvelimen. SQL-tietokantaan tallennetaan kaikki robotin käyttämät tiedot kuten prosessit, käyttäjät ja lokitiedostot.

Tuotantoympäristössä ohjelmistorobotin infrastruktuurille suositellaan tämän lisäksi muutamia lisävaatimuksia. Blue Prismin mukainen suositus robotin infrastruktuurista on esitetty seuraavassa kuvassa.



Kuva 23 Ohjelmistorobotin infrastruktuuri

Kuvassa 23 infrastruktuuri on jaettu neljään osaan. SQL-tietokanta asennetaan kuvassa näkyvälle tietokantapalvelimelle. Tietokantapalvelimeen yhdistetään sovelluspalvelin, jonka tehtävänä on toimia yhdyskäytävänä muiden infrastruktuurin osien kanssa. Itse robotin työskentely tapahtuu kuvassa näkyvissä Blue Prism -resursseissa. Nämä resurssit voivat olla yksittäisiä tietokoneita tai virtuaaliympäristöjä, joihin ohjelmistorobotin ohjelmisto on asennettuna. Viimeinen infrastruktuurin osa on käyttäjän tietokone, johon ohjelmistorobotin ohjelmisto on asennettuna. Käyttäjän tietokoneen kautta tapahtuu robotin ympäristön hallinnointi. Sen sisällä tapahtuu muun muassa ohjelmistorobotin kehitys ja resurssien hallinnointi.

Edeltävän mukainen infrastruktuuri mahdollistaa useiden resurssien ja käyttäjätietokoneiden käytön samassa ympäristössä. Käytännössä resurssien määrä on kuitenkin yleensä rajattu ohjelmistorobotin lisenssissä. Usein lisenssin hinnoittelu perustuuakin samanaikaisesti työskentelevien resurssien määrään.

Opinnäytetyössä noudatettiin edellä mainittuja Blue Prismin suosituksia arkkitehtuurista. Tietokanta ja sovelluspalvelin perustettiin Windows Server -ympäristöön. Runtime Resourceksi hankittiin oma tietokone ja käyttöliittymät asennettiin käyttäjien omille työtietokoneille.

Itse sovelluksien asennus ja ympäristön pystytys on yksinkertaista. Blue Prism tarjoaa ohjelmiston asennukseen ja ympäristön määrittämiseen selkeät ohjeet. Pohjatieto SQL-palvelimien ja Windows Server -ympäristöjen hallinasta kuitenkin nopeuttaa käyttöönottoa huomattavasti. Tietokantaa ohjelmistorobotille ei tarvitse itse rakentaa vaan riittää, että palvelin ympäristö on pystytetty ja SQL Server -ohjelmisto asennettu palvelimelle. Ensimmäistä kertaa yhdistäessä tietokantaan Blue Prism -ohjelmiston kautta, luo Blue Prism -ohjelmisto tarvitsemansa tietokannan SQL-palvelimelle.

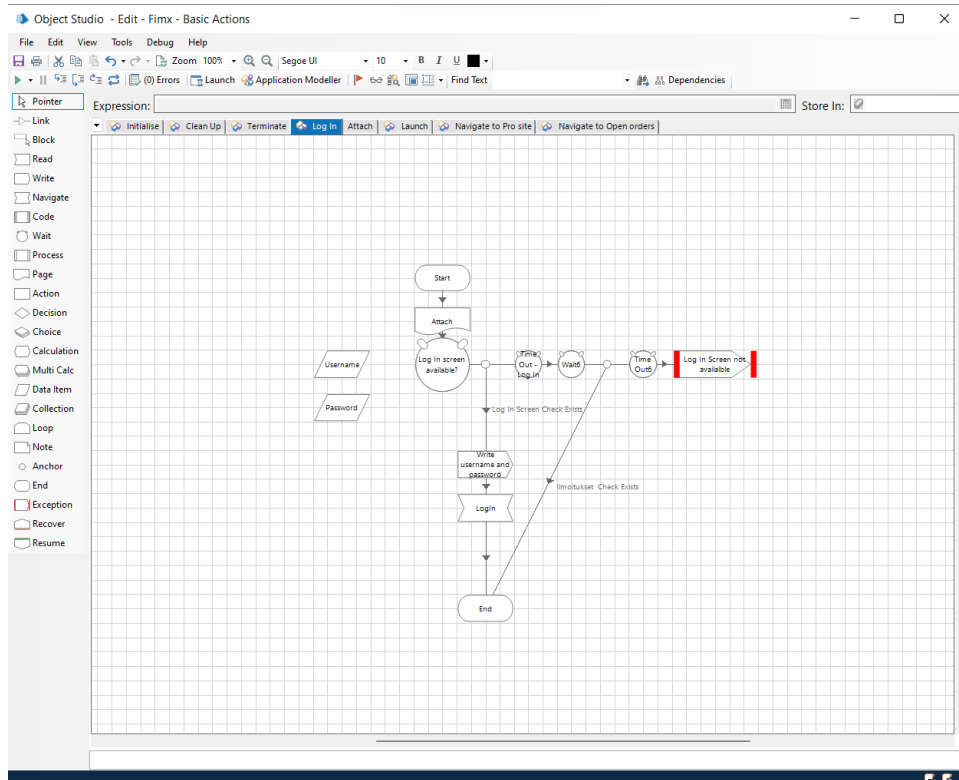
Jotta Blue Prismin kaikki ominaisuudet olisi mahdollista saada käyttöön joutui infrastruktuuria laajentamaan vielä jonkin verran. Blue Prism tarjoaa käyttäjilleen lisäksi Blue Prism Hub -sovelluksen. Blue Prism Hub tarjoaa yrityksen työntekijöille pääsyn ohjelmistorobotin ympäristöön verkkoselaimen kautta. Hubiin on liitettävissä myös erilaisia lisäosia ohjelmistorobotille. Lisäosat sisältävät esimerkiksi automaation elinkaaren hallinnan, koneoppimisympäristön ohjelmistorobotin päätöksenteon tueksi, ja interaktiivisen käyttöliittymän. Hubin käyttöönotto vaati uuden SQL-tietokannan sekä verkko- ja välittäjäpalvelimien perustamisen.

5.2.2 Ohjelmistorobotin kehitys

Blue Prismissä ohjelmistorobotin kehitys tapahtuu erilaisia prosessikaavion omaisia kaavioita tekemällä. Ohjelmiston kehitysympäristöä kutsutaan studioksi, joka jakautuu objekti- ja prosessistudioon. Objektien ja prosessien kehitys tapahtuu samankaltaisesti muutamina eroavaisuuksineen.

Objekteissa määritetään toiminnallisuudet robotille automaatiossa tarvittavien sovelluksien käyttämiseen. Yksittäinen objekti sisältää tavallisesti vain yhteen

sovellukseen liittyviä toimintoja. Parhaiden käytäntöjen mukaisesti objektit tulisi pitää pieninä niin, että sovelluksen eri toiminnalliset osa-alueet jaetaan eri objekteihin. Objektit jakautuvat edelleen yksittäisiin toimintoihin. Seuraavassa kuvassa on esitetty työn aikana tehty objekti yhden kumppanijärjestelmän perustoiminnoista.



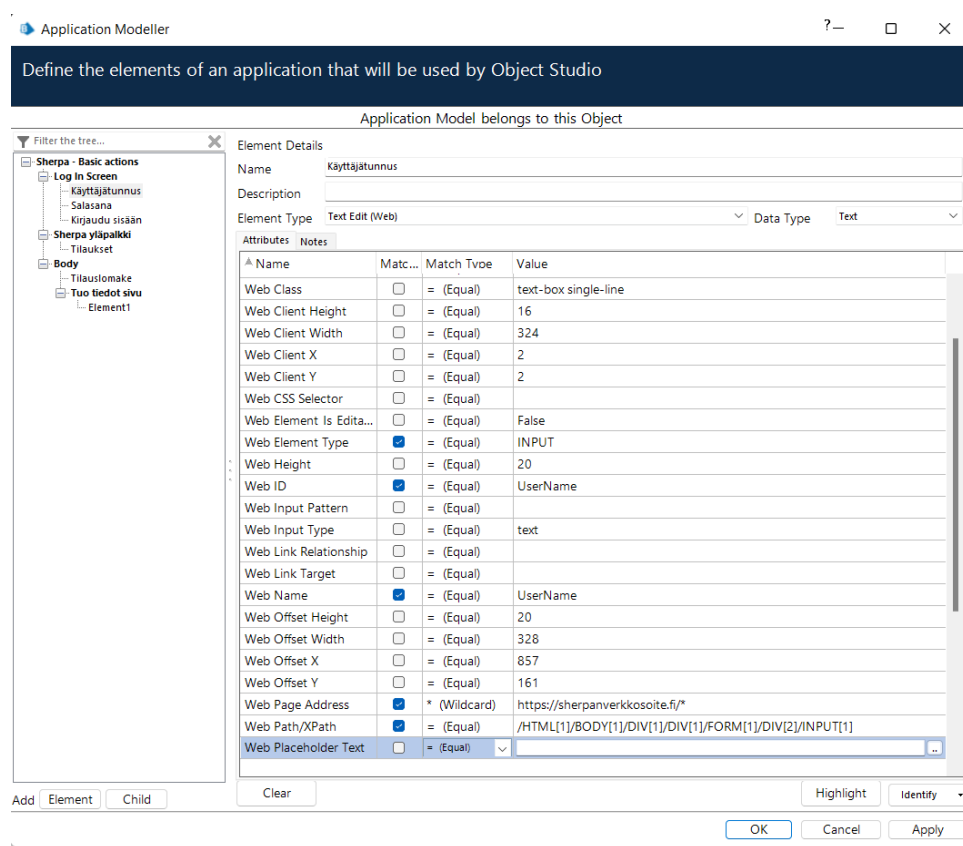
Kuva 24 Blue Prism Object Studio

Kuvassa 24 esitetään objektin sisältämä kirjautu sisään toiminto. Objektin sisältämät eri toiminnot on eritelty välilehdillä, jotka näkyvät rivissä prosessikaavion yläpuolella. Ikkunan vasemmalla reunalla sijaitsevalla valikolla on näkyvissä kaikki objektissa käytettävissä olevat vaiheet. Objektin kehitys tapahtuu raahaamalla jokin vaihe valikosta ruudukolle. Vaiheeseen tehdään tarkemmat määrittelyt sen suorituksesta, jonka jälkeen vaiheet yhdistetään toisiinsa.

Kuvassa näkyvässä toiminnossa robotti ensin tarkistaa *Attach*-vaiheen avulla, että kumppanin järjestelmä on auki verkkoselaimessa ja yhdistää robotin tähän sovellukseen. *Attach*-vaihe on viittaa objektissa määritettyyn toiseen toimintoon,

jossa on määritetty tarkemmin toiminnon sisältämät vaiheet. Jokainen objektin toiminto on määritetty suorittamaan aluksi tämä *attach*-toiminto. *Attach*-vaiheen jälkeen robotti odottaa, että kirjautumissivu on saatavilla. Heti robotin havaitessa kirjautumissivun syöttää robotti sivustolle käyttäjänimen ja salasanan, ja painaa kirjautu sisään painiketta. Mikäli robotti ei tunnistanut kirjautumissivua sille määritetyn odotusajan puitteissa tarkistaa robotti, onko sivuston etusivu saatavilla. Mikäli etusivu on saatavilla, tulkitaan, ettei kirjautumista tarvita ja robotti lopettaa toiminnon. Mikäli etusivukaan ei ole saatavilla, ilmoittaa robotti poikkeamasta.

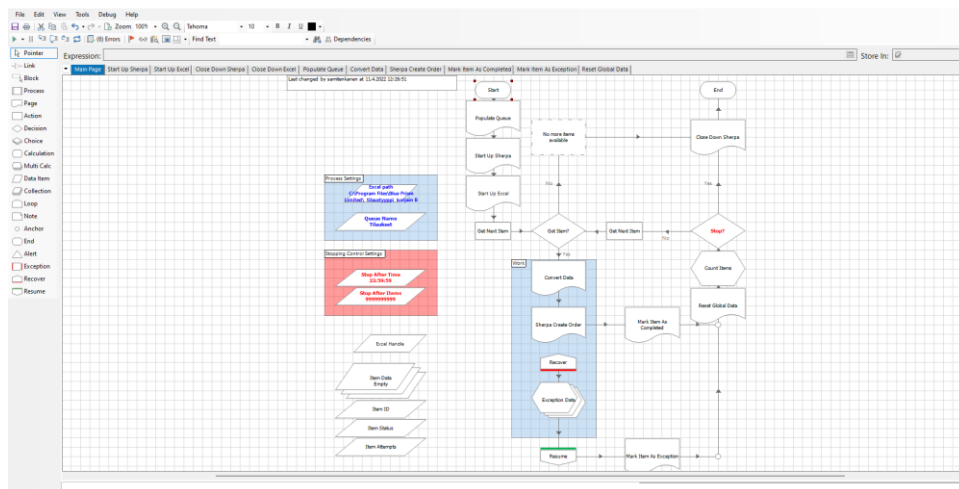
Jotta robotti pystyy toimimaan sovelluksen kanssa, täytyy robotti määrittää tunnistamaan sovelluksessa tai sivustolla olevat tarpeelliset osat. Tämä tapahtuu objektin sisältämän Application Modelerin avulla.



Kuva 25 Blue Prism Application Modeler -näkökulma

Edeltävässä kuvassa on esitetty Application Modeler -näkyvä Sherpan perustoiminnot objektista. Elementtien eli sovelluksen tai sivuston osien lisäys Application Modeleriin onnistuu yksinkertaisimmillaan klikkaamalla jotakin sivuston osaa. Osaa painettaessa Blue Prism hakee klikatun elementin attribuutit ja esittää ne kuvassa näkyvässä listauksessa. Attribuuttien avulla robotti pystyy tunnistamaan oikean elementin myöhemmin. Robotti käyttää ainoastaan valittuja attribuutteja oikean elementin tunnistamiseen. Usein käytettäviä attribuutteja ja niiden arvoja on muutettava, jotta voidaan varmistua siitä, että robotti pystyy varmasti tunnistamaan oikean elementin.

Objektien ei tulisi sisältää mitään liiketoimintaan liittyviä päätöksiä. Kaikki tällainen päätöksenteko tulisi sijoittaa prosesseihin. Prosesseilla määritetään sanan mukaisesti, kuinka jokin liiketoimintaprosessi tai työtehtävä suoritetaan. Prosessit voivat hyödyntää monia eri objekteja. Objekteja hyödynnetään prosesseissa viittaamalla yksittäiseen objektin sisältävään toimintoon.



Kuva 26 Prosessin etusivu Blue Prismissä

Kuvassa 26 on esitetty valmiin prosessin etusivu, jolla robotti hakee tilaukset yhdestä kumppanin järjestelmästä ja syöttää ne Sherpa-järjestelmään. Prosessissa on käytetty Blue Prismin tarjoamaa prosessipohjaa, jossa prosessikaavio ja poikkeamanhallinta on tehty valmiiksi parhaiden käytäntöjen mukaan. Valmiin

pohjan osat on korvattu prosessia varten tehdyillä toiminnoilla. Jokaista työvaihetta varten on tehty oma sivu prosessiin, jossa työvaiheen tarkemmat toiminnot on määritetty. Etusivulla määritetään näiden prosessin osien suoritusjärjestys.

Ensin prosessi suorittaa *Populate Queue* -työvaiheen. Tämä työvaihe sisältää toisen prosessin, jossa robotti hakee kumppanin järjestelmästä kaikki avoimet tilaukset, syöttää ne robotin työjonoon ja kuittaa tilauksen vastaanotetuksi. Tämän jälkeen robotti avaa Sherpa-järjestelmän ja Excel-sovelluksen sekä kirjautuu tarvittaessa sisään Sherpa-järjestelmään. Sovellusten avaamisen jälkeen robotti hakee seuraavan tilauksen työjonosta. Ensin robotti muuntaa kumppanin järjestelmästä haetut tiedot Sherpa-järjestelmän hyväksymään muotoon ja määrittää tilaukselle tilaustyypin Excelin avulla. Kun tilauksen tiedot on muutettu, syöttää robotti tilauksen tiedot Sherpa-järjestelmään ja luo tilauksen. Mikäli tilauksen luonti onnistui, merkkää robotti tilauksen käsiteltyksi omaan työjonoonsa. Lopuksi Sherpa-järjestelmä valmistellaan uuden tilauksen luontia varten, ja robotti hakee seuraavan tilauksen työjonosta. Mikäli tietojen muuntamisessa tai tilauksen luonnissa tapahtuu jokin virhe, merkataan tilauksen kohdalle poikkeama. Kun työjonosta ei ole enää saatavilla uutta tilausta sulkee robotti Excelin ja Sherpa-järjestelmän sekä lopettaa prosessin suorittamisen.

Automatisaatioiden kehitys aloitettiin tilausten syöttö -työvaiheesta. Ensin päätettiin automatisoida kumppani järjestelmä 3:sta siirrettävät tilaukset. Valintaan päädyttiin, koska järjestelmistä ei ollut saatavilla testiympäristöä, vaan kehitys täytyi tehdä suoraan tuotantoympäristössä. Järjestelmä 3:sta tulevien tilausten volyyymi oli kaikista pienin, jolloin saatiin minimoitua mahdollisten virheiden ja kehityksen aiheuttaminen viiveen vaikutus tilausten käsittelyyn.

Kehitys aloitettiin rakentamalla objektit ja toiminnot järjestelmien välille. Jokaisen toiminnon toiminta testattiin ennen seuraavaan toimintoon siirtymistä. Ensin valmisteltiin kaikki tarvittavat toiminnot tilauksien tietojen hakemiseksi järjestelmästä, jonka jälkeen kehitettiin toiminnot Sherpan tilauksen luontia varten. Tilauksien luomiseen harkittiin kahta eri keinoa: 1. Robotti luo tilaukset yksitellen

suoraan Sherpa-järjestelmään järjestelmän oman käyttöliittymän kautta. 2. Robotti syöttää tilauksen tiedot Excel-tiedostoon, jonka avulla kaikki syötettävät tietueet tuodaan Sherpa-järjestelmään samalla kertaa. Järjestelmä 3:n osalta päädyttiin vaihtoehtoon 1.

Tilausten syöttöä varten tehtyjä toimintoja testattiin ensin luomalla sen avulla testitilauksia Sherpa-järjestelmään. Laskutusasiakkaan valinta suoritettiin robotin avulla kirjoittamalla kohde asiakas kenttään ja valitsemalla ensimmäinen vaihtoehto, mikäli sellainen löytyi. Mikäli yhtäkään kohdetta ei löytynyt, merkitsi robotti asiakkaaksi "Ei Löytynyt" -nimisen asiakkaan. Aikaisemmin työssä suoritettuna asiakaspuun organisoinnin seurauksena oletettiin ensimmäisen vaihtoehdon olevan oikea tarpeeksi usein. Kun toimintojen toiminta oli testattu, rakennettiin prosessi tietojen muuntamiseksi. Tässä vaiheessa täytyi robotin myös suorittaa tilaustyyppin määrittäminen tilaukselle.

Tilaustyyppin määrittämiseen harkittiin koneoppimista hyödyntävän tekstinkategorisointi sovelluksen luontia sekä Excel-työkalua. Aluksi päätettiin kokeilla Excel-työkalua, ja mikäli määrittämisestä ei saada tarpeeksi tarkkaa tutkittaisiin koneoppimisvaihtoehtoja.

Excel-työkalua varten Sherpa-järjestelmästä haettiin 1000 järjestelmä 3:sta tullutta käsiteltyä tilausta. Tilauksille haettiin valittu tilaustyyppi sekä kumppanin järjestelmästä saatavilla oleva työnkuvaus. Työnkuvauksista haettiin tilaustyyppi kohtaisesti yleisimmän ilmenevät sanat, jotka listattiin kortistoon. Excelliin luotiin logiikka, jonka avulla tarkistetaan, löytyykö kortistosta löytyvä sana kuvauksesta. Mikäli sana löytyy, palautetaan sanaa vastaava tilaustyyppi. Näin saatuja tilaustyyppijä verrattiin tilauksien oikeisiin tilaustyyppihin ja kortistolle laskettiin onnistumisprosentti. Virheellisiä määrittämiä tutkittiin muun muassa silmämääräisesti ja laskemalla jokaiselle yksittäiselle sanalle virheprosentti. Tämän avulla kortistoa optimointiin, kunnes onnistumisprosentin koettiin olevan riittävä. Onnistumisprosenttia mitattiin kahdessa kategoriassa: 1. Palautettu tilaustyyppi ohjaa tilauksen oikealle osastolle. 2. Palautettu tilaustyyppi on täysin oikein.

77 cm	KOD - Kodinkoneasennus Technics	0	36	0%	työtilaus	Huolto	0	1	0,0%	Liesituulet Kodinkonehuolto - muut	1	11	5,3%
77cm	KOD - Kodinkoneasennus Technics	0	87	0%	toiminnan tarkastus	Huolto	0	2	0,0%	Liesituulet kodinkonehuolto - muut	0	4	0,0%
Tilaan	KOD - Kodinkoneasennus Technics	0	13	0%	pellit	Huolto	0	1	0,0%	Liesiku Kodinkonehuolto - muut	1	16	6,3%
Tilataan	KOD - Kodinkoneasennus Technics	2	56	4%	riitä	Huolto	0	0	0,0%	Tiskikone Kodinkonehuolto - apk	0	1	0,0%
Tilaamme	KOD - Kodinkoneasennus Technics	0	0	0%	tarkistetaan toiminta	Huolto	0	1	0,0%	Astianpesu Kodinkonehuolto - apk	0	7	0,0%
tilalle	KOD - Kodinkoneasennus Technics	0	15	0%	tilataan huolto	Huolto	0	1	0,0%	APK Kodinkonehuolto - apk	0	3	0,0%
Usitaan	KOD - Kodinkoneasennus Technics	0	71	0%	asennetaan kuiva	PA - Pesukoneasennus	0	6	0,0%	ASP Kodinkonehuolto - apk	0	9	0,0%
Usitaan	KOD - Kodinkoneasennus Technics	0	71	0%	asennetaan pesu	PA - Pesukoneasennus	0	1	0,0%	Pakastin Kodinkonehuolto - kylmä	1	31	3,2%
Usitaan	KOD - Kodinkoneasennus Technics	0	71	0%	uuden kuv	PA - Pesukoneasennus	0	0	0,0%	Pakastime Kodinkonehuolto - kylmä	1	58	1,7%
Tyhyä asunto	KOD - Kodinkoneasennus Technics	0	3	0%	uuden kuv	PA - Pesukoneasennus	0	0	0,0%	Jääkaappi Kodinkonehuolto - kylmä	1	88	1,1%
Vaihdetaan	KOD - Kodinkoneasennus Technics	0	3	0%	kodinkone tilaus	KOD - Kodinkoneasennus Technics	0	3	0,0%	Jääkaappi Kodinkonehuolto - kylmä	1	17	5,9%
Asennettuna	KOD - Kodinkoneasennus Technics	0	0	0%	kodinkone tilaus	KOD - Kodinkoneasennus Technics	0	3	0,0%	JPK Kodinkonehuolto - kylmä	1	126	0,8%
Asentaa	KOD - Kodinkoneasennus Technics	1	4	25%	Jääkaapin uusiminen	KOD - Kodinkoneasennus Technics	0	4	0,0%	pakaste Kodinkonehuolto - kylmä	0	11	0,0%
Asentaa	KOD - Kodinkoneasennus Technics	1	4	25%	Jääkaapin uusiminen	KOD - Kodinkoneasennus Technics	0	4	0,0%	JPK Kodinkonehuolto - kylmä	2	49	4,1%
asennetaan	KOD - Kodinkoneasennus Technics	0	1	0%	Jääkaapin vaihto	KOD - Kodinkoneasennus Technics	0	1	0,0%	Heila Kodinkonehuolto - liesi	0	22	0,0%
Liesi 77cm	KOD - Kodinkoneasennus Technics	0	0	0%	uusin uusinta	KOD - Kodinkoneasennus Technics	0	2	0,0%	Uuni Kodinkonehuolto - liesi	0	6	0,0%
Astianpesukone	KOD - Kodinkoneasennus Technics	0	0	0%	lieden uusinta	KOD - Kodinkoneasennus Technics	0	23	0,0%	Liesi Kodinkonehuolto - liesi	0	0	0,0%
jak 60	KOD - Kodinkoneasennus Technics	0	0	0%	lieden uusiminen	KOD - Kodinkoneasennus Technics	0	12	0,0%	Lieden Kodinkonehuolto - liesi	1	25	4,0%
Liesituulet	Kodinkonehuolto - muut	0	8	0%	pakastimen uusinta	KOD - Kodinkoneasennus Technics	0	5	0,0%	Tiskari Kodinkonehuolto - apk	0	0	0,0%
Liesituuletin	Kodinkonehuolto - muut	0	1	0%	Jääkaapin uusinta	KOD - Kodinkoneasennus Technics	0	3	0,0%	Pesula Pesulakonehuolto	0	33	0,0%
Liesiku	Kodinkonehuolto - muut	1	4	25%	Lieden vaihto	KOD - Kodinkoneasennus Technics	0	6	0,0%	Pesukone Pesulakonehuolto	0	38	0,0%
Tiskikone	Kodinkonehuolto - apk	0	1	0%	uusi jkp	KOD - Kodinkoneasennus Technics	0	4	0,0%	Pesutupa Pesulakonehuolto	0	11	0,0%
Astianpesukone	Kodinkonehuolto - apk	0	6	0%	uusi jääkaappi	KOD - Kodinkoneasennus Technics	3	27	11,1%	Kukuri Pesulakonehuolto	0	5	0,0%
APK	Kodinkonehuolto - apk	0	2	0%	Tilataan uusi	KOD - Kodinkoneasennus Technics	0	46	0,0%	Kuivausr Pesulakonehuolto	0	31	0,0%
Dish	Kodinkonehuolto - apk	0	0	0%	usitaan	KOD - Kodinkoneasennus Technics	0	105	0,0%	Pyykki Pesulakonehuolto	0	4	0,0%
Tiskari	Kodinkonehuolto - apk	0	0	0%	uusi liesi	KOD - Kodinkoneasennus Technics	1	15	6,7%	rumpu Pesulakonehuolto	0	3	0,0%

Kuva 27 Tilaustyyppien määrittämiseen käytetty kortisto

Kuvassa 27 on esitetty näin saatu kortisto ja sanojen virheprosentit sekä niiden tilauksien määrä, jotka on määritetty kyseisen sanan perusteella. Kuvassa on näkyvissä noin puolet koko kortiston sisältämistä sanoista. Riittävän tarkkuuden saamiseksi jouduttiin käyttämään sanakombinaatioita ja eri sanamuotoja samoista sanoista. Sanojen haku tapahtui kortistoa pitkin ylhäältä alaspäin pysähtyen ensimmäiseen osumaan. Ensin työkalu tarkistaa keskimäisen kortiston, jolla pyrittiin määrittämään käsittelevä osasto. Mikäli osastona oli, huolto siirtyi työkalu määrittämään tarkemman tilaustyyppin oikeanpuoleisesta kortistosta. Mikäli osastoa ei voitu määrittää keskimäisen kortiston perusteella päätettiin tilaustyyppi vasemmanpuoleisen kortiston perusteella.

Kortiston avulla voitiin testidatasta määrittää tilauksen käsittelevä osasto 98,2 %:n tarkkuudella, ja oikea tilaustyyppi 94,4 %:n tarkkuudella. Tarkkuus koettiin riittäväksi, jotta työkalua lähdetäisiin testaamaan tuotannossa. Tilaustyyppin määrittämisessä tapahtuneita virheitä tarkasteltiin ja suuri osa havaittiin olevan tilauksia, joissa tilaukselle oli olemassa useampi oikea tilaustyyppi. Näin todellinen virheprosentti jäi vielä laskettua pienemmäksi.

Lopuksi työkalusta tehtiin kopio, josta poistettiin testidata ja analyysiin käytetyt funktiot. Näin saatiin Excel-tiedosto, johon robotti pystyi syöttämään tilauksen työkuvausten yhteen soluun ja sai tilaukselle määritetyn tilaustyyppin poimittua toisesta solusta.

Kun tietojenmuunnosprosessi ja tilaustyyppimääritys-työkalu olivat valmiita, yhdistettiin näin saadut prosessin osat aikaisemmin kuvassa esitetyn prosessin kaavion mukaisesti. Prosessin testaus aloitettiin ajamalla robotin kautta tilauksia Sherpaan yksi kerrallaan. Robotin toimintaa seurattiin koko sen työskentelyn ajan. Tässä vaiheessa robotti ei vielä itse kuitannut tilauksia vastaanotetuiksi kumppanin järjestelmään, vaan kuittaus tehtiin manuaalisesti, kun tilauksen luonti havaittiin onnistuneeksi. Tilausten käsittelijöitä informoitiin robotin tekevän järjestelmään tilauksia ja heiltä pyydettiin antamaan palautetta sitä mukaan, kun he kehityskohteita tai virheitä havaitsisivat.

Kun yksittäisten tilausten luonti todettiin toimivaksi, alettiin robotin avulla siirtämään useampia tilauksia kerralla. Prosessin aikana tapahtuneet virheet jaettiin robotin toimesta parhaiden käytäntöjen mukaisesti järjestelmä- ja liiketoimintaperäisiksi. Liiketoimintaperäiset virheet olivat virheitä, jossa tilauksella olevissa tiedoissa oli virheitä, jotka estivät tilauksen luonnin onnistuneesti. Järjestelmäperäiseksi virheeksi luokiteltiin virhe, joka aiheutui robotin virheellisestä toiminnasta tai sen käyttämisestä järjestelmistä johtuvaksi. Virheille pyrittiin etsimään juurisyy sitä mukaan, kun niitä esiintyi ja tekemään robottiin tarvittavat muutokset niiden estämiseksi. Kun robotti oli onnistunut suorittamaan prosessin viisi perättäistä kertaa ilman järjestelmäperäisiä ongelmia, lisättiin robotille toiminto kuitata tilaukset prosessin aikana itse vastaanotetuiksi. Tämän jälkeen robotti ajoitettiin suorittamaan prosessi kahden tunnin välein. Ensimmäisten ajojen jälkeen käytiin tarkastamassa, että kaikki haetut tilaukset olivat kuitattuna vastaanotetuiksi kumppanin järjestelmässä. Ajojen välissä mahdolliset virheen aiheuttaneet tilaukset käytiin manuaalisesti luomassa järjestelmään. Virheille pyrittiin etsimään juurisyy, niiden merkittävyys priorisoitiin ja virheen korjaus laitettiin työjonoon. Samoin tehtiin työntekijöiden ilmoittamille palautteille.

Tässä vaiheessa aloitettiin seuraavan automatisaation kehittäminen, jolla siirrettiin vastaavasti järjestelmä 2:sta tulevat tilaukset Sherpaan. Tässä automatisaatiossa hyödynnettiin edellisessä automaatiossa kehitettyjä toimintoja. Myös aikaisemman automaation prosessi kopioitiin, josta korvattiin kumppanin järjes-

telmään liittyvät toiminnot. Tilaustyyppimääritys työkalusta tehtiin kopio, joka optimoitiin kyseessä olevaa järjestelmää varten. Samoin tietojenmuunnos prosessi kopioitiin ja muunneltiin tukemaan prosessia.

Automatisaation edetessä havaittiin nopeasti muutamia haasteita. Ensimmäisen automatisaation kohdalla valittu tilaustenluonti tapa, joka hyödynsi Sherpan käyttöliittymää ei tukenut tarvittavan pitkiä työnkuvauksia. Järjestelmä 2:sta tulevissa tilauksissa oli keskimäärin huomattavasti pidempi työnkuvaus kuin järjestelmä 3:sta tulevissa tilauksissa. Lisäksi tilaustenkäsittelijöiltä saatiin palautetta, että he kaipaisivat lisätietoja tilaukselle sellaisiin kenttiin, joiden kirjoittaminen ei ollut mahdollista tilausta luodessa käyttöliittymän kautta. Pidempien työselosteiden ja haluttujen tietojen lisääminen oli mahdollista joko avaamalla tilaus eri näkymässä sen luomisen jälkeen, tai tuomalla tiedot järjestelmään Excelin avulla. Tässä vaiheessa päätettiin muuntaa prosessi hyödyntämään Exceliä. Excelin hyödyntäminen tässä vaiheessa mahdollistaisi prosessien vertailun ja paremmaksi havaitun vaihtoehdon hyödyntämistä tulevissa automatisaatioissa.

Excel-pohjana hyödynnettiin yrityksen aikaisemmin käytettyä Excel-työkalua. Työkalusta pyrittiin havaitsemaan virheet, ja ylimääräiset asiat, jotka poistettiin. Exceliin lisättiin toiminto, jonka avulla se haki järjestelmä 2:en mukaisen kumppanin asiakaspuun Sherpa-järjestelmästä, jolloin oikea laskutusasiakas pystyttiin määrittämään Excel-työkalussa. Myös aikaisemmin kehitetty tilaustyyppimääritys työkalu yhdistettiin samaan Excel-tiedostoon. Työkaluun lisättiin myös tuki kaikille uusille tietueille, joihin robotti haki tietoa kumppanin järjestelmästä. Kun Excelin tuominen Sherpaan havaittiin onnistuvan robotilta, prosessi päivitettiin ja tarvittavat muutokset tehtiin.

Tilausten syötön automatisoinnin jälkeen automatisoitiin tilausten kuittaus kumppanin järjestelmään. Myös näissä prosesseissa käytettiin samaa prosessi pohjaa kuin aikaisemmissa automatisaatioissa. Robotti ohjattiin hakemaan kaikki yhden kumppanin valmiit tilaukset, jotka olivat tarkastettuja mutta eivät vielä laskutettuja, suoraan Sherpan tietokannasta. Kaikki tilaukset, joista puuttui

työseloste, jätettiin haun ulkopuolelle. Tämän jälkeen haetut tiedot lisättiin robotin toiseen työjonoon. Robotti määritettiin hakemaan tilaukset yksitellen kumppanin järjestelmästä ja kuittaamaan ne valmiiksi. Kuittauksen aikana robotti myös kirjasi työselosteen sekä toimitetut tavarat kumppanin järjestelmään. Onnistuneen kuittauksen jälkeen tilaus merkattiin työjonossa valmiiksi ja lisättiin vielä kolmanteen työjonoon.

Lopuksi robotille tehtiin vielä prosessi, jossa robotti haki tilauksien tiedot kolmannesta työjonosta ja muodosti näistä Excel-tiedoston, jonka robotti syötti Sherpa-järjestelmään. Tämän työvaiheen tarkoituksena oli tuoda tilaukselle tieto, että tilaus on kuitattu kumppanin järjestelmään. Näin tilauksen laskuttaja pystyi näkemään, että tilaus on valmiina laskutettavaksi.

Kun tilausten kuittaus automatisaatio oli valmis, kopioitiin prosessi ja muutettiin tukemaan toisesta järjestelmästä tulleita tilauksia. Lopuksi molemmat prosessit ajoitettiin ajettavaksi kerran päivässä.

6 Tulosten arviointi ja jatkokehitys

Tässä luvussa selvitetään työn vaikutusta prosessiin ja nykytila-analyysin aikana esitettyihin mittareihin. Lopuksi myös esitetään ehdotus prosessin sekä yrityksen ohjelmistorobotiikan jatkokehityksestä.

6.1 Projektin vaikutus mittareihin

Tilausten käsittelyaikojen osalta voitiin havaita selkeää positiivista kehitystä automaation käyttöönoton jälkeen. Käyttöönoton jälkeen mitatut käsittelyajat on esitetty seuraavassa kuvassa. Kuvaan on vertailun vuoksi sisällytetty Järjestelmä 1:tä tulleiden tilausten käsittelyajat. Lisäksi käsittelyajat on jaettu automatisaation kautta tulleisiin ja muita kautta tulleisiin tilauksiin. Muuta kautta tulleet tilaukset johtuvat robotin vaiheittaisesta käyttöönotosta sekä muun muassa sähköpostin kautta tulleista tilauksista.

Pääasiakas (ryhmät)	Huolto	Kodinkoneasennus	Muu	Yhteensä
☐ Järjestelmä 1	3,84	4,79	5,99	4,43
☐ Järjestelmä 2	2,50	2,91	8,53	3,42
Automaatio	2,50	2,31		2,33
Muu		4,78	8,53	5,97
☐ Järjestelmä 3	5,97	2,71	2,22	4,26
Automaatio	3,83	1,40		2,59
Muu	10,67	5,46	2,22	7,68
Yhteensä	4,23	3,91	6,53	4,21

Kuva 28 Tilausten käsittelyajat projektin jälkeen

Tuloksista nähdään, että järjestelmä 1:en osalta huollon käsittelyajat ovat hieman lyhentyneet, mutta muiden osastojen tilauksien käsittelyajat taas pidentyneet tarkastelujaksojen välillä. Robotin kautta tulleiden tilausten osalta käsittelyajoissa nähdään kuitenkin suurempi muutos. Aikaisempaan tarkastelujaksoon huollon ja kodinkoneasennuksen tilauksen osalta käsittelyajat ovat yli puoliintuneet. Suurin muutos nähdään Järjestelmä 3:n kohdalla kodinkoneasennuksen tilauksissa.

Mittaustuloksissa tulisi ottaa huomioon, että opinnäytetyön aikataulun vuoksi tuloksien ajallinen otanta on puolet pienempi kuin nykytila-analyysin aikana käytetty otanta. Myös automaation piirissä olevien järjestelmien osalta automaation kautta tulleiden tilausten määrä oli huomattavasti suurempi muita kautta tulleisiin tilauksiin.

Käsittelyn vasteaikoihin ei automaatiolla nähty olevan selkeää vaikutusta.



Kuva 29 Tilauks käsittelyn vasteajat projektin jälkeen

Kuvasta 29 nähdään, että automaation kautta tulleiden tilausten vasteaika oli jopa hieman suurempi verrattuna muuta kautta tulleisiin tilauksiin. Syynä tähän saattoi olla tilausten syöttämisen aikataulun muutos ja frekvenssin pieneminen. Käsittelijät ovat saattaneet käsitellä tilauksia edelleen vain kaksi kertaa päivässä vanhan totumuksen mukaan, vaikka tilauksia on automaation myötä tulleet kahden tunnin välein. Näin käsittelyiden välissä tulleiden tilausten vasteajat voivat olla hieman pidemmät.

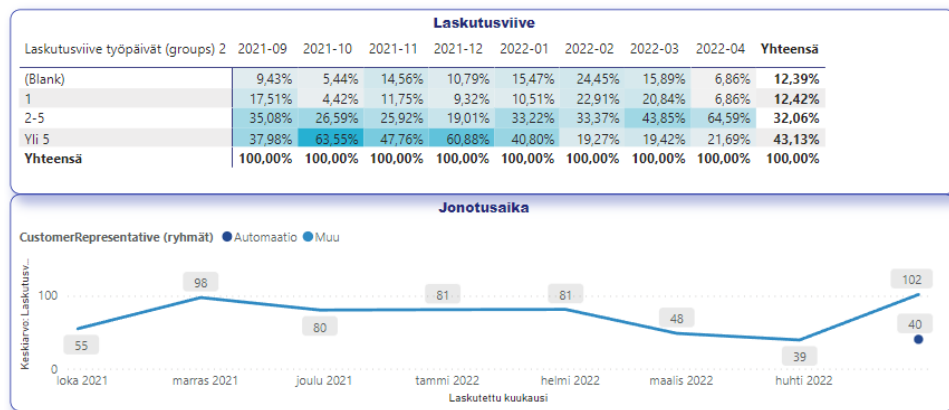
Käsittelyviive työpäivät (groups)	tammikuu 2022	helmikuu 2022	maaliskuu 2022	huhtikuu 2022	Yhteensä
(Blank)	72,35%	71,25%	54,76%	49,03%	63,11%
1	26,18%	24,28%	38,10%	35,48%	30,94%
2-5	1,47%	3,51%	6,19%	15,48%	5,37%
Yli 5	0,96%	0,95%	0,95%		0,57%
Yhteensä	100,00%	100,00%	100,00%	100,00%	100,00%

Kuva 30 Tilauks käsittelyn vasteajat kategorioittain projektin jälkeen

Automaatiolla ei nähdä selkeää vaikutusta myöskään kuvassa näkyvässä vasteaikojen kategorisella mittauksella. Tavoitteen ylittäneiden tilausten määrä oli

jopa hieman noussut ohjelmistorobotin käyttöönoton jälkeen. Toisaalta vasteaikojen ylittäneiden tilausten suhteellinen määrä on ollut kasvu trendissä jo vuoden alusta alkaen. Johtopäätöksiä tekeminen vasteaikoihin vaikuttavista tekijöistä vaatisi syvempää tutkimista.

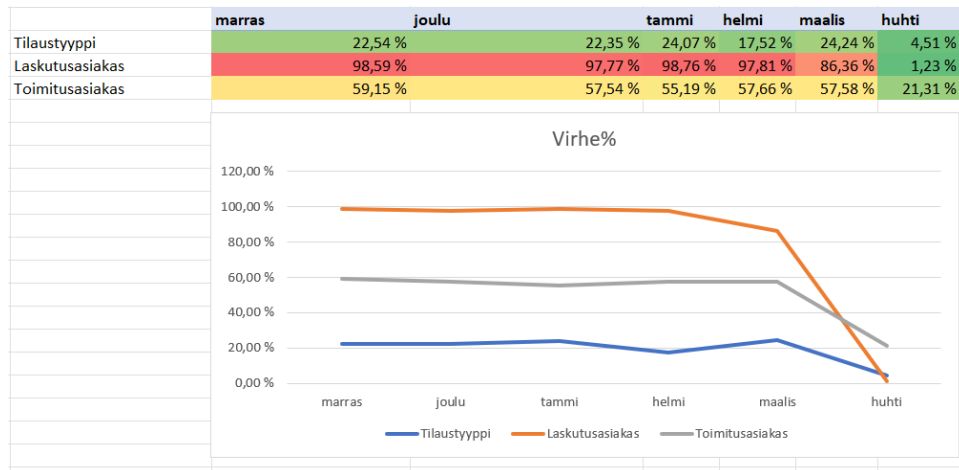
Laskutusviiveen osalta selkeää johtopäätöstä ei myöskään ollut mahdollista tehdä.



Kuva 31 Laskutusviiveet projektin jälkeen

Ohjelmistorobotin kuittaamien tilausten osalta laskutusviive pysyi samalla tasolla kuin ennen ohjelmistorobotin käyttöönottoa. Toisaalta automatisaation piiriin kuulumattomien tilausten laskutusviive oli kasvanut huomattavasti.

Virhemäärien laskeminen ei nykytila-analyysin mukaisesti ollut mahdollista mitata tilauksista, jotka tehtiin Sherpan käyttöliittymän kautta. Tämä johtui siitä, että käyttöliittymän kautta ei ollut mahdollista varmistaa robotin valitsemalle asiakkaalle yksilöivää arvoa, jota olisi myöhemmin voinut verrata käsiteltyyn tilaukseen. Järjestelmä 2:en osalta virhe määrien mittaaminen oli mahdollista, sillä robotin järjestelmään tuomat Excel-tiedostot tallennettiin. Tulokset on esitetty seuraavassa kuvassa.



Kuva 32 Tilauksien syötön virhemäärä projektin jälkeen

Kuvan 32 mukaisista tuloksista havaitaan virhemäärien pudonneen merkittävästi. Huhtikuun osalta tuloksissa on mukana ainoastaan ohjelmistorobotin luomat tilaukset. Tilaustyyppimääritys-työkalu osoitti toimivuutensa myös tuotantoympäristössä, ja virheprosentti vastaa testauksessa saatuja arvoja. Tilaustenkäsitteijöiden esille nostama laskutusasiakas tieto saatiin robotin avulla valittua melkein 100 % tarkkuudella. Virheellisissä tiedon sisältämissä tilauksissa oli kaikissa kohde, jota ei vielä tilausta syöttäessä ollut luotu Sherpa-järjestelmään, eikä robotti näin ollen voinut valita oikeaa asiakasta. Myös toimitusasiakas tietojen tarkkuus oli parantunut huomattavasti, vaikkakin virheitä oli vielä noin 20 prosentilla tilauksista.

Kuvissa esitettyjen tuloksien lisäksi robotti sai muun muassa merkattua toimitusasiakkaan numeron noin 60 prosentille tilauksista. Satunnaisotannalla tehtiin testi 20 tilaukselle, joihin robotti ei saanut puhelinnumeroa merkattua. Testissä havaittiin, ettei yhdellekään tilauksista ollut puhelinnumeroa saatavilla kumpparin järjestelmässä.

Mittauskohde	Tarkennettu mittauskohde	Tulos keskimäärin	
		Projektin alussa	Ohjelmistorobotin käyttöönoton jälkeen
Käytetty työaika	Tilausten syöttö	60-120 min/pvä	0 min/pvä
	Tilausten käsittely	5 min/tilaus	2,5 min/tilaus
	Tilausten kuittaus	1 min/tilaus	0 min/tilaus (Mikäli työseloste löytyy)
Vasteajat	Käsittelyn vasteaika ka.	5-8 tuntia	8 tuntia
	Tilauksia käsitelty tavoiteajassa	85 %	85 %
	Kuittauksen vasteaika ka.	50-80 tuntia	40 tuntia (Robotin kautta tulleet tilaukset)
	Tilauksia kuitattu tavoiteajassa	30 %	15-35 %
Virhemäärät	Tilautustyyppi	20-40 %	4,50 %
	Toimitusasiakas	50 - 70 %	20 %
	Laskutusasiakas	80 - 100 % (yhdessä järjestelmässä vain 25%)	1 %

Kuva 33 Nykytila-analyysin aikana toteutettujen mittarien tulokset ennen ja jälkeen

Kuvassa 33 on esitetty yhteenveto mittareiden arvojen muutoksesta. Käyttöönoton jälkeen mitatuissa tuloksissa otetaan huomioon ainoastaan robotin kautta tulleiden tilauksien tulokset. Yhteenvedossa esitetyt tulokset ovat suurin piirtein, ja tarkemmat luvut on esitetty aikaisemmin kappaleessa.

6.2 Projektin kannattavuus ja ROI

Saatujen tulosten pohjalta ohjelmistorobotille laskettiin vuosittainen tuotto sekä käyttöönottoprojektin ROI. Vuosittainen tuotto saatiin laskemalla automatisaation säästämä työaika erotettuna ohjelmistorobotin kustannuksilla. Säästetyn työajan laskennassa käytettiin tuloksista saatua arviota, että robotti säästää tilauksen käsittelyn työaika noin 2,5 minuuttia/tilaus. Säästettyyn työaikaan laskettiin lisäksi tilauksen kuittauksiin käytetty aika, joka arvioitiin olevan noin 1 minuutti/tilaus. Myös tilauksen syöttöön kulunut aika laskettiin säästyneeksi täysimääräisenä. Tämän arvioitiin säästävän aikaa noin 60 minuuttia/työpäivä. Lopuksi laskettiin järjestelmistä tuleva keskimääräinen kuukausittainen tilausmäärä, jolla kerrottiin tilauskohtainen säästetty työaika. Laskennassa käytettiin dataa viimeiseltä 12 kuukaudelta. Henkilötyötunnin arvona käytettiin kehityspäälliköltä saatuja arvioita henkilötyötunnin kustannuksesta yritykselle.

Ohjelmistorobotin kuluihin laskettiin ohjelmistorobotin lisenssi, robotin käyttämät palvelimet sekä robotin ylläpitoon käytettävä työaika. Ylläpitoon käytettäväksi

työajaksi laskettiin 12 h/vko. Ylläpitoon käytettävää työaikaa oli hankalaa tarkasti määrittää vielä tässä vaiheessa. Mikäli robotin toiminnassa ei tapahdu virheitä eikä sen käyttämissä järjestelmissä muutoksia, on ylläpitoon tarvittava työaika todennäköisesti huomattavasti pienempi.

Projektin kustannuksiksi laskettiin projektiin kulunut työaika sekä ohjelmistorobotin käyttöönottoon liittyvät kertaluontoiset kustannukset, kuten ohjelmistorobotin vaatimat laitteet.

	Vuosi	1	2	3
Nykyinen automaatio	Vuosittainen tuotto %	-1,71 %	13,10 %	13,10 %
	Projektin ROI	-13,03 %	73,94 %	160,92 %
Järjestelmä 1 mukaan lukien	Vuosittainen tuotto %	51,01 %	80,07 %	80,07 %
	Projektin ROI	403,92 %	935,61 %	1467,31 %

Kuva 34 Projektin ROI ja ohjelmistorobotin tuottavuus

Kuvan 34 mukaisista tuloksista voidaan nähdä, että nykyisellä ohjelmistorobotin käyttöasteella ohjelmistorobotti alkaa tuottamaan yhden vuoden jälkeen. Mikäli järjestelmä 1 saadaan mukaan automatisaatioon piiriin kasvaa robotin tuottavuus huomattavasti. Laskuissa järjestelmä 1:en kohdalla oletetaan säästyvän yhtä paljon aikaa per tilaus kuin aikaisemmissa automatisaatioissa.

6.3 Työntekijöiden kokemukset

Parhaat käytänteet osiossa todettiin, ettei ohjelmistorobotiikan kannattavuuden mittausta kannata rajoittaa robotin suorien rahallisten säästöjen mittaukseen. Lean-ajattelussa tuodaan esille työntekijöiden ja heidän mielipiteiden tärkeyttä yrityksen toiminnassa. Tästä syystä projektin lopussa pyydettiin tilaustenkäsittelijöitä vastaamaan viiteen kysymykseen projektiin liittyen. Kysymyksien tarkoituksena oli selvittää työntekijöiden mielipiteitä projektin onnistumisesta, sekä automatisaation vaikutuksista heidän työhönsä ja työskentelykokemukseensa. Vastauksia saatiin yhteensä neljältä työntekijältä. Kysymykset ja vastaukset on esitetty liitteessä 1 ja yhteenveto alla.

Vastauksista voitiin havaita, että yleinen mielipide automatisaatiota ja ohjelmistorobottia kohtaan olivat myönteisiä. Ohjelmistorobotin jatkokehitykseen ja lisää automatisaatioihin suhtauduttiin hyvin positiivisesti. Ohjelmistorobotin koettiin myös vapauttaneen työaika muihin työtehtäviin. Myös Agile-mallia hyödyntämällä saavutettu nopea palautteisiin reagointi, ja virheiden korjaus tuli esille yhden tilauksen käsittelijän vastauksissa.

6.4 Ohjelmistorobotin tulevaisuus yrityksen toiminnassa

Ohjelmistorobotin yksi selkeimpiä vahvuuksia on sen skaalautuvuus. Robotti pystyy työskentelemään kellon ympäri aiheuttamatta ylityökustannuksia tai lisäkuluja käyttöasteen kasvaessa. Tästä syystä jatkoautomatisaatiokohteiden kar-toittaminen robotin käyttöönoton jälkeen on suositeltavaa.

Tulevien automatisaatiokohteiden hyödyntämisessä voidaan muun muassa hyödyntää Lean-ajattelun mukaisia malleja ja työkaluja. Lean-ajattelun mukaisesti kehitys tapahtuu ensin prosessia kehittämällä, jonka jälkeen prosessia pyritään automatisoimaan mahdollisuuksien mukaan. Automatisaatio ja ohjelmistorobotiikka tulisi nähdä työkaluna prosessien kehityksessä sen sijaan, että se nähtäisiin ratkaisuna.

Automatisaation myötä työntekijöiltä vapautuu työaika muihin tehtäviin. Vapautunutta työaika voidaan myös ohjata jatkuvan kehityksen mahdollistamiseen. Tällöin työntekijät voivat käyttää vapautunutta työaika itsensä ja yrityksen prosessien kehittämiseen.

Lean mallia hyödyntäessä prosesseille tulisi ensin määrittää omistajat, jotka ovat vastuussa prosessien kehityksestä. Tämän jälkeen prosessien omistajien ja työntekijöiden esimiesten tulisi kannustaa työntekijöitä Lean-ajatteluun ja jatkuvan kehityksen malliin.

Jatkuvaa kehitystä voitaisiin ylläpitää toistuvien Kaizen-tapaamisten avulla, joihin osallistuisivat prosessin omistajat ja työntekijät. Tapaamisissa pyrittäisiin

selvittämään prosessien ongelmia, kehityskohteita sekä hukkaa. Tukena ongelmien löytämisessä ja ratkaisussa olisi suositeltavaa ensin hyödyntää A3-ajattelun mukaisia metodeja, jonka jälkeen muita Lean-työkaluja ja automatisaatiokeinoja voitaisiin hyödyntää.

Kehityskohteiden löytämiseksi yritys voisi myös hyödyntää arvoketjukuvauksia. Tällöin yrityksen tulisi pyrkiä määrittämään olemassa olevat arvoketjut ja niihin osallistuvat prosessit. Prosessien omistajien tulisi pyrkiä ymmärtämään prosessien merkitys arvoketjussa. Arvoketjukuvauksien avulla voitaisiin hukkaa tunnistaa arvoketjuista. Hukkien poistossa voitaisiin jälleen hyödyntää ohjelmistorobotiikkaa sekä erilaisia Lean-työkaluja.

Automatisaatiokohteita ohjelmistorobotille voi myös etsiä prosesseista ilman Lean-ajattelumallin implementointia organisaatiossa. Tällöin olisi suositeltavaa ensin kartoittaa automatisaatioon soveltuvat työtehtävät ja prosessit. Kun kartoitus on tehty, tulisi havaitut kohteet pisteyttää. Pisteytyksessä arvioidaan prosessin soveltuvuutta automatisaatioon sekä prosessin automatisaation kannattavuutta. Pisteytyksessä voidaan ottaa huomioon esimerkiksi: Prosessin työvaiheiden määrä ja kuinka moni työvaiheista voidaan automatisoida, kuinka paljon aikaa voidaan automatisaatiolla vapauttaa, mikä on työntekijöiden suhtautuminen automatisoitaviin työtehtäviin, kuinka usein prosessi suoritetaan, miten automatisaatio vaikuttaa prosessin suorituskykyyn ja kuinka nopeasti automatisaatio on tehtävissä. Pisteytyksen jälkeen prosessit lajitellaan pisteytyksen mukaiseen järjestykseen, jolloin parhaan pisteytyksen saanut prosessi automatisoidaan ensimmäisenä.

7 Yhteenveto

Yrityksen tilauksenkäsittelyn prosessin suurimmat haasteet olivat rutiinin omaisen työn suuri määrä, huonot integraatiomahdollisuudet järjestelmien välillä sekä puutteelliset ja eriävät ohjeistukset työnsuorittajien kesken.

Opinnäytetyössä keskityttiin ohjelmistorobotin käyttöönottoon ja kehitykseen tilausten käsittelyn prosessin tueksi. Ohjelmistorobotin käyttöönoton ohella prosessin nykytila analysoitiin ja prosessiin tehtiin muutoksia automatisaation hyötyjen parantamiseksi. Lisäksi työssä selvitettiin ohjelmistorobotiikkaan ja prosessien kehitykseen liittyviä parhaita käytänteitä ja tutustuttiin Lean ajattelumalliin. Ohjelmistorobotin kehityksessä hyödynnettiin parhaiden käytänteiden lisäksi Agile-metodologiaa. Robotin käyttöönoton jälkeen selvitettiin automatisaation kannattavuutta, ja sen vaikutuksia automatisoitavaan prosessiin.

Ohjelmistorobotin käyttöönotolla havaittiin olevan positiivinen vaikutus automatisoitavaan prosessiin. Ohjelmistorobotin myötä työvaiheiden virhemäärä sekä niihin kuluvat henkilötyötunnit havaittiin vähentyneen merkittävästi.

Suurin muutos virhemäärissä havaittiin laskutusasiakkaan valinnassa, jossa virheellisen tiedot sisältävien tilausten osuus väheni 80–100 prosentista noin yhteen prosenttiin. Muiden tietueiden osalta virhemäärät vähenivät noin 50–80 prosenttia riippuen mitattavasta tietueesta. Tilausten syöttöön ja kuittaamiseen kuuluvat työtehtävät saatiin automatisoitua täysin, ja tilauksen käsittelyyn käytettävä aika puoliintui noin 5 minuutista/tilaus noin 2,5 minuuttiin/tilaus. Automatisaatio ja ohjelmistorobotin käyttöönotto nähtiin prosessiin osallistuvien työntekijöiden kesken selkeästi positiivisena asiana.

Insinöörityöprojekti oli kokonaisuudessaan hyvin mielenkiintoinen. Projektin aikana pääsi tutustumaan eri aihepiireihin laajasti. Varsinkin parhaita käytäntöjä selvitettäessä oppi paljon asiaa myös opinnäytetyön aiheen ulkopuolelta.

Projektin aikana sai paikoitellen todella haastaa oman osaamisensa ja oppimiskykynsä. Muun muassa ohjelmistorobotin vaatimasta infrastruktuurista ja sen

valmistelusta ei itselläni ollut juurikaan aikaisempaa kokemusta. Palvelimien pystyttämiseen ja käyttöönottoon sai siis perehtyä työn aikana toden teolla.

Agile-metodologian hyödyntämisestä ohjelmistorobotin kehityksessä olisi mielenkiitoista nähdä lisää tutkimuksia. Toistaiseksi tietoa oli vasta vähän saatavilla, ja olemassa oleva tieto pohjautui yksittäisiin kokemustutkimuksiin.

Lähteet

- Antony J., 2011. Six Sigma vs Lean: Some perspectives from leading academics and practitioners. Verkkoaineisto <https://www-emerald-com.ezproxy.metropolia.fi/insight/content/doi/10.1108/17410401111101494/full/html> Luettu 8.4.2022.
- Bell, S. ja Orzen, M., 2010. Lean IT: Enabling and Sustaining Your Lean Transformation. Verkkoaineisto <https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=589866> Luettu 7.3.2022
- Fantina R. Ym., 2021. Introducing Robotic Process Automation to Your Organization: A Guide for Business Leaders. Verkkoaineisto <https://learning.oreilly.com/library/view/introducing-robotic-process/9781484274163/> Luettu 8.4.2022.
- Laakkonen K., 2014. Contracts in Agile Software Development. Verkkoaineisto <https://aaltodoc.aalto.fi/handle/123456789/14435> Luettu 19.4.2022.
- Philipp, I., 2021. Agile RPA Verkkoaineisto <https://www.scrum.org/resources/agile-rpa> Luettu 13.4.2022.
- Serrador, P ja Pinto, J., 2015. Does Agile work? — A quantitative analysis of agile project success. Verkkoaineisto <https://www-sciencedirect-com.ezproxy.metropolia.fi/science/article/pii/S0263786315000071> Luettu 13.4.2022.
- Syed R. ym., 2019. Robotic Process Automation: Contemporary themes and challenges. Verkkoaineisto <https://www-sciencedirect-com.ezproxy.metropolia.fi/science/article/pii/S0166361519304609> Luettu 8.4.2022.
- Taulli T., 2020. The Robotic Process Automation Handbook : a guide to implementing RPA systems. Verkkoaineisto <https://learning.oreilly.com/library/view/the-robotic-process/9781484257296/html/Cover.xhtml> Luettu 8.4.2022.

Liitteet

Työntekijöiden kokemukset projektin lopuksi

Kysymys 1: Kuinka tyytyväinen olet robotin toimintaan asteikolla 1–10?

- Vastaja 1: “tyytyväisyyteni on tässä vaiheessa 10”
- Vastaja 2: “8”
- Vastaja 3: “Robotin toiminnassa oli joitakin puutteita ehkä ensimmäisen vuorokauden aikana. Tosin joko minä huomasin puutteet myöhässä tai ne korjattiin älyttömän nopeasti, sillä Robotin toiminta oli 8 luokkaa jo seuraavana päivänä (Vertauksena käytän aikaisemmin tehtyä järjestelmä integraatiota toisella alueella) Robotti ei pääse ihan kymmppiin ehkä enemmän sen takia että Robotin työ ei ole minulle niin paljon näkynyt VIELÄ”
- Vastaja 4: “Olen tyytyväinen asteikolla 9”

Kysymys 2: Onko automatisaatio helpottanut/vaikeuttanut työtehtäviesi suorittamista?

- Vastaja 1: “on selvästi helpottanut perustöitä”
- Vastaja 2: “On helpottanut”
- Vastaja 3: “Robotin tultua kumppani 3:en tilausten käsittelyyn menee huomattavasti vähemmän. Eli kyllä.”
- Vastaja 4: “Automatisaatio on helpottanut työtäni”

Kysymys 3: Koetko, että sinulta on automatisaation myötä vapautunut aikaa? (esimerkiksi muiden työtehtävien suorittamiseen)

- Vastaja 1: "Kyllä on vapautunut aikaa vaativampien tehtävien tekemiseen"
- Vastaja 2: "On säästänyt hieman aikaa"
- Vastaja 3: "Kyllä"
- Vastaja 4: "On vapautunut jonkin verran vapaata aikaa muihin työtehtäviin"

Kysymys 4: Kuinka positiivisena asiana koet nykyisten kaltaisten automatisaatioiden lisäämisen (1-10)

- Vastaja 1: "koen, että näissä perustöissä ehdottomasti 10"
- Vastaja 2: "10"
- Vastaja 3: "10, en ymmärrä miksi näkisin asian negatiivisena"
- Vastaja 4: "Asteikolla 9"

Kysymys 5: Avoin palaute. (Olisiko esimerkiksi projektissa voinut tehdä jotain toisin)

- Vastaja 1: "Tämä projekti on edennyt erittäin jouhevasti ja olen ollut tyytyväinen. Hyvä mieli tuli kun vapautuvan ajan voi käyttää vaativampiin töihin ja ei itse tarvitse olla robottina"
- Vastaja 2: -
- Vastaja 3: "Paha sanoa, joten sanon mikä erottui vahvasti minulle. Palautteen käsittely toimi valonnopeudella joka on ymmärtääkseni järjestelmämuutoksissa / päivityksissä stereotyyppistä poikkeava asia"

- Vastaja 4: "Tämä on jo nyt hyvä asia minkä Robotti tekee, ei tietenkään pysty kaikkea tekemään mitä pitää mutta todella hyvä näin