Aalto University

School of Science

Master's Programme in Industrial Engineering and Management

Tuomas Kyheröinen

Implementation of Robotic Process Automation to a Target Process – a Case Study

Master's Thesis

Espoo, April 10, 2018

Supervisor: Professor Kari Tanskanen

Thesis advisor: Tony Leppänen, M.Sc. (Tech.)

Aalto University
School of Science

Abstract of master's thesis

Author Tuomas Kyheröinen

Title of thesis Implementation of Robotic Process Automation to a Target Process – a Case Study

Master's programme Industrial Engineering and Management

Thesis supervisor Prof. Kari Tanskanen

Major or Minor/Code Operations and Service Management / SCI3049

Thesis advisor Tony Leppänen, M.Sc.(Tech)

Date 10.04.2018 **Number of pages** 68+7 **Language** English

Abstract

Advances in technology have throughout the history enabled organizations to improve their efficiency. With robotic process automation (RPA), organizations can finally begin to properly improve the productivity of knowledge-based work, too. With this prospect now at hand, companies need to figure a way to effectively and efficiently implement RPA to their processes. In knowledge-based work, automation solutions need to be tailored and customized to adapt to the needs for each target process. The RPA literature and project literature offer the building blocks to construct a model for implementing RPA solution to a target process.

The aim for this thesis is to develop an understanding on RPA implementations and the forces that govern such projects. To this end, the study first explores relevant literature and synthesizes it to develop a template model for RPA implementation process, which then can be tested in a real life scenario, a case project. Producing, testing and improving the RPA implementation model is the core objective of this thesis. Through meticulously analysing the findings from this course of study, required understanding can be formed.

RPA literature in general, due to the technology's novelty, is not very comprehensive or mature as of yet. The existing literature is focused mostly on presenting what RPA is, instead of diving further down to nuanced connections. This study contributes to the existing literature by binding different established concepts together to form new knowledge. Using the context of RPA as a glue, literature about project life-cycles and project success were fused together to create an implementation process specifically tailored for RPA projects. In creating the RPA process model, project CSFs were closely and meticulously examined and adapted to specifically fit RPA projects. The case project in this study helps in establishing the validity of the aforementioned findings.

From a managerial viewpoint, the main contribution of this study is establishing a template for this RPA process model, enabling proper RPA implementation projects. Indicated by the success of the case project, the RPA implementation process model proved to be properly made, successful and suitable for RPA. To help make RPA projects successful, the study introduces multiple critical success factors that need to be fulfilled; by concentrating sufficient resources on each of these factors managers can ensure successful RPA implementation projects. RPA projects can be treated as isolated and still be successful, but by following the established set of CSFs, there is more to gain than just success in singular projects.

Keywords Robotic Process Automation, Project Management

Aalto-yliopisto
Perustieteiden
korkeakoulu

Diplomityön tiivistelmä

Tekijä Tuomas Kyheröinen

Työn nimi Tapaustutkimus robottisen prosessiautomaation käyttöönotosta kohdeprosessissa

Koulutusohjelma Industrial Engineering and Management

Valvoja Professori Kari Tanskanen

Pää- tai sivuaine/koodi Operations and Service Management / SCI3049

Työn ohjaaja Tony Leppänen, M.Sc.(Tech)

Päivämäärä 10.04.2018 **Sivumäärä** 68+7 **Kieli** Englanti

Tiivistelmä

Läpi historian teknologian kehitys on mahdollistanut organisaatioiden tehokkuuden kasvun. Robottimaisen prosessiautomaation (RPA) avulla organisaatiot voivat vihdoin kunnolla aloittaa myös tietopohjaisen työn tehokkuuden parantamisen. Tämän mahdollisuuden edessä yrityksien täytyy löytää tehokas ja vaikutuksellinen tapa ottaa RPA käyttöön prosesseissaan. Tietopohjaisessa työssä automaatioratkaisut täytyy räätälöidä ja kustomoida jokaiseen käyttökohteeseen ja prosessin sopiviksi. RPA- ja projektikirjallisuus tarjoavat rakennuspalikat kohdeprosessin RPA-ratkaisun käyttöönottomallin rakentamiseen.

Tämän diplomityön päämäärä on ymmärtää RPA-käyttöönottojen ja siihen liittyvien projektien taustalla vaikuttavia voimia. Tätä varten tutkimus aloitetaan kirjallisuudesta, joka syntetisoidaan prosessimalliksi RPA-käyttöönottoprojekteille. Sitten kyseistä mallia testataan käytännössä case-projektissa. Tämän RPA käyttöönottomallin tuottaminen, testaaminen ja parantaminen on työn avaintavoite. Analysoimalla työn löydöksiä tarkasti ja perinpohjaisesti voidaan saavuttaa riittävä ymmärrys työn päämäärän saavuttamiseksi.

Johtuen RPA-teknologian uutuudesta, sitä koskeva kirjallisuus ei yleisesti ottaen ole vielä kattavaa. Olemassaoleva kirjallisuus keskittyy suurilta osin esittelemään RPA:n ominaisuuksia, sen sijaan että käsitteitä ja yhteyksiä käsiteltäisiin syvällisesti. Tämä työ täydentää aiempaa teoriaa yhdistelemällä vakiintuneita käsitteitä muodostaen uutta tietoa. Käyttämällä RPA:ta kontekstisena liimana projektielinkaari ja projektien onnistumista käsittelevää kirjallisuutta yhdistetään muodostamaan erityistesti RPA:lle räätälöity käyttöönottoprosessi. Tätä mallia luodessa projektien kriittisiä onnistumiskriteerejä tarkasteltiin ja mukautettiin RPA:han sopiviksi. Case-projekti puolestaan auttaa vahvistamaan edellämainittujen löydösten validiuden.

Johtamisen näkökulmasta työn tärkein anti on RPA-käyttöönottoprojektimalli, joka mahdollistaa kunnollisen RPA-projektien läpiviennin. Case-projektin onnistuminen osoittaa, että käyttöönottomalli on asianmukaisesti rakennettu, onnistunut ja sopiva RPA:han. RPA-projektien onnistumisen edesauttamiseksi työ esittelee monia kriittisiä onnistumiskriteerejä – projektijohtajien tulee kohdistaa riittävästi resursseja jokaiseen näistä kriteereistä, jotta projektien onnistumisesta voidaan varmistua. RPA-projekteja voidaan kohdella itsenäisinä ja ne voivat silti onnistua, mutta onnistumiskriteerejä seuraten voidaan mahdollistaa myös kokonaisen projektiverkon onnistuminen.

Avainsanat Robottinen prosessiautomaatio, projektin hallinta

Acknowledgements

First of all, I wish to thank those who helped me in completing this thesis. Thank you Tony for your help, for all the feedback, but especially thank you for giving me the chance to work on this thesis. I'm also very grateful for Prof. Kari Tanskanen for his help in steering me and the thesis to the right direction, especially at the very beginning when it was difficult to properly define and limit my topic.

Special thanks also to my co-workers, who in their part allowed me to find time for writing the thesis, week after week. Working on a thesis during office hours is not possibly in all work environments, especially when they are open offices.

Also, thank you for all the staff at Aalto. I count the courses I have completed have had at least 50 different teachers and countless more assistants, visitor lecturers and others who on their part aided in me completing my degree. Thank you for all the memories!

Tuomas Kyheröinen

Helsinki, April 2018

Contents

List of abbreviations

AI	Artificial Intelligence
BPM	Business Process Management
BPMS	Business Process Management Software/System
CIMO	Context, Intervention, Mechanism, Outcome
CRM	Customer Relationship Management
CSF	Critical Success Factor
ERP	Enterprise Resource Management
FTE	Full-Time Equivalent
RPA	Robotic Process Automation

1 Introduction

1.1 Background and motivation

The technological frontier of humanity is ever expanding. In all areas of our society, researchers are constantly seeking new technological innovations in order to increase productivity and make human life easier. For over a hundred years, automation has been a major source for such innovations, a phenomenon that has enabled organizations to increase their productivity and gain competitive advantage. Automation has reduced the need for attention and effort from human employees, allowing a higher productivity per person. This, in turn, has allowed the society in general to grow, both in financial and technological dimensions. At first, automation revolutionized industries relying on heavy physical efforts. For example, the upkeep of a farm used to require multiple people, but now with technological innovations and automation the human resources needed have decreased. Furthermore, while replacing human workers, automation has transformed the skill requirements: a farmer from a hundred years ago had did everything manually; today farmers mainly need to know machine maintenance. Another example of industries with completely revolutionized business processes is the car industry. Manufacturing cars was at first handcrafting. Then, first with the invention of the assembly line and further the introduction of robots has shaped the industry in ways few could predict based on where it all started.

In the last few decades, automation has revolutionized knowledge-based work. The invention of computers and later their widespread availability to organizations has changed the way people in knowledge-based industries work. From then on, the digital tools at workers' disposal have developed exponentially, from first text editors to fully automated complex business processes. There is still a lot left to discover when it comes to improving the degree of automation.

Two key aspects that form the backbone of knowledge-based work are data transfers and data manipulation. For example, in the payroll industry, payroll specialists use the input data of employment contracts as the basis to calculate the salaries for employees of an organization. Automation revolves mainly around these two aspects, and potential improvements in the degree of automation are a result of finding new ways to automate these tasks. For example, the data can be transferred and imported via automated file transfers and in some cases parts of the data manipulation can be performed automatically, by using e.g. macros in a spreadsheet to perform calculations. The main problem in increasing the degree of automation is not all systems are capable of importing or exporting files. Most often, such systems are old and no-longer-maintained legacy systems. These systems have once gained ground and a large enough user base that they cannot be simply replaced. One archetype of these systems are web- and form-based applications, which rarely support file importing. Another equally important reason can be cost: the changes to enable these features are often additional, meaning they cost extra. Whatever the barrier may be, to be able to increase the use of automations with these outdated systems and further increase the level of automation, in a strive for increased productivity, a new solution is needed.

Robotic Process Automation (RPA) is a form of digital process automation that uses robotic software, a type of program that can mimic user's behaviour. This type of software can be programmed to perform tasks just as a human would, only more quickly and precisely. It is possible to even see RPA perform the actions for the users, moving the cursor around as it interacts with applications. This type of interaction works well with legacy systems, because just like a human would be able to fill a form or a sheet, the robot is able to perform the same task. RPA can address this gap to progress automation and much more.

There is a lot of potential in implementing RPA in business processes that do not deal with legacy systems, as well. RPA can further the degree of automation, freeing up human resources to be used elsewhere or to plainly cut costs. This efficiency improvement is why many companies are interested in this new concept and RPA is on the rise. However, as with any technology, to introduce RPA effectively, a proper way of implementing the technology in a form of process model must be determined.

1.2 Research problem and research questions

From a researcher's point of view, the main intrigue with RPA is how it differs from existing IT solutions, especially from the implementation perspective. In a nutshell, it all comes down to the design approach. RPA is designed on top of other systems and does not function independently like e.g. Enterprise Resource Planning (ERP) systems could. You could compare RPA to a blanket covering different systems together, a tool that acts like a glue and does nothing by itself. The design approach also means that most – if not all – RPA solution need to be tailor-made for each system and process. This gives higher emphasis on the interaction between the developers and those who work with the underlying processes, than in similar automation cases involving traditional IT projects.

Based on this description, it can be argued that the research on traditional IT project implementations is not entirely applicable to RPA implementations as is, and, therefore, a research gap exists. The primary objective of this thesis is to explore this gap, to discover what is important in RPA implementations and how they differ from traditional implementations. The focal phenomenon itself could be described as factors of implementation success with RPA.

However, there is no point in re-inventing the wheel; while RPA differs from traditional IT projects, there are bound to be transferrable elements from their implementation projects. An important point is to recognize similarities and differences with other IT project implementations. After finding the most applicable elements, seeing how well they apply to RPA implementation is only a matter of testing. To this end, a template for the timeline of such implementation should be formed; this can be achieved by examining research on project life-cycles, and combining the best fitting elements to form a project life-cycle model for RPA implementation projects. Following this logic, the Research Questions guiding this thesis are:

RQ1 What elements have been recognized as crucial in IT project implementations?

RQ2 What stages make up a project life-cycle that fits RPA?

RQ3 How will these elements fare when combined and applied to RPA implementations?

These questions will form the backbone of this study, and they will be revisited in the conclusions section.

1.3 Methodology and structure of work

The aim for this thesis is to develop an understanding on RPA implementations and the forces that govern such projects. To this end, the study first explores relevant literature and synthesizes it to develop a template model for RPA implementations, which then can be tested in a real life scenario, a case project. Producing, testing and improving the RPA implementation model is the core objective of this thesis. Through meticulously analysing the findings from this course of study, required understanding can be formed and the research questions detailed before can be answered.

A case study with critical realism perspective will be used as the basis for research design in this thesis. Case study is applicable here because the phenomenon is really context-dependent.

In operation management, studies have traditionally focused on the theoretical side, examining courses of action that have already taken place. However, when exploring implementation of a new technology, the implications need to be studied beforehand. To this end, the core approach for the thesis is taken from design science. Design sciences, in general, focus on pre-emptive impact assessment, rather than examining effects after the fact. This approach better suits the nature of this study. Denyer et al. (2008) have introduced a four-part logic that can be used to construct a design theory around a new intervention. The method was dubbed CIMO, based on its facets: Context, Intervention, Mechanisms and Outcome. The logic of this framework is to analyse a setting, where an Intervention is introduced to a Context. There are, then, certain Mechanisms that this will trigger in interaction with the Context. These consequences will lead to Outcomes. The output of following this logic is a set of design propositions. They are a summarization of the discovered forces in play. For example, they can depict one Mechanism and its logic to add to how the Intervention should be designed. These suggestions can aim to improve the process of implementation of the Intervention, but this is not necessarily always the case.

The CIMO logic will provide the background theme for the analysis in this thesis. The first steps in this thesis, in form of the literature review, aim to give the reader a better understanding on all four areas of the CIMO logic. First, existing literature about RPA is explored to better define the concept, which will produce a deeper understanding on the Context. Then, literature on project success allows more insight on Outcomes to be gained. Focusing, then, on project implementations and their success factors, will give an insight on what important aspects and concepts can be outlined. This way, the project literature will help to depict the Interactions and Mechanisms at play.

These findings from literature are used together, to pick parts of models and success factors which are most applicable to RPA. This way, a model for the implementation of RPA is formed, by combining the best-suited parts. In section 3, the research design is laid out. As the research uses a case study, this section also explains the logic in selecting the case for deploying and testing the newly developed model.

The findings from applying the model to a case are outlined in section 4. These discoveries will help understand the interplay of the four parts of CIMO logic. Based on the findings and gathered data, a set of design propositions for RPA implementations can be outlined. These suggestions, as well as other conclusions, will be covered in section 5.

2 Literature Review

The structure of this literature review section is three-fold. First, a literature review on RPA is done to give a better indication of what it actually is about. Then, the review delves deeper in the implementation of projects. The purpose of this part is to discover what has been previously studied about IT projects and what has been deemed important in their implementation. The purpose of this half is to outline different elements that could be then used to form a process model for RPA implementations. Finally, the literature is summarized and synthesized into a RPA implementation model, which features the most applicable elements from the literature with the context of RPA in mind.

2.1 Characteristics of RPA

Robotic Process Automation as a term has a futuristic ring to it. The concept often leads people to imagine physical robots roaming around office space performing tasks just like humans would (Lacity & Willcocks, 2015B; Lacity et al., 2015). In reality, RPA is not as glamorous. Just like with all automation, the concept means replacing processes previously done by humans, but this time done by configuring a robotic software to perform the tasks, interacting between different systems such as spreadsheets, Customer Relationship Management (CRM) systems or Enterprise Resource Planning (ERP) software (Lacity & Willcocks, 2015; Lacity et al., 2015). In short, RPA provides the tools – software and platform – to automate rule-based, logical processes involving well-defined and structured data with a deterministic set of output values (Lacity & Willcocks, 2016). In addition, the tasks are often repetitive and less desirable to do by hand (Lacity & Willcocks, 2016). Such tasks can be labelled "swivel chair", referring to moving inputs from one side to outputs on the other side without much need for consideration (Willcocks et al., 2015B). Without all the mysticism, this is all RPA does – interacts with systems just like a human. However, the robot should, if given a suitable process and well-defined working logic, outperform humans in terms of quality, time and cost (Willcocks et al, 2015B).

The goal when using robots is not to simply assist humans in the processes. Instead, where used, RPA should replace humans entirely (Lacity et al., 2015). This comparison is similar to different tools, for example Excel sheets, which is a tool to assist users to perform different calculations – the human presence is still required. In RPA, the logic is different, with the calculations done behind the scenes, entirely by the robot, with only input and output visible and accessible by a human.

Since RPA does not involve actual physical robots, a need arises to define a way to quantify robots. In RPA terms, one robot equals one software license (Willcocks et al., 2015B). The robots are usable full-time, but to run, for example, 10 processes simultaneously one would need 10 robots or licenses.

There are two main differences between other methods of automation and RPA, both of the two aspects can be described with "lightness". Firstly, contrary to what it might seem like, configuring the robots to do their job does not require a vast knowledge of programming (Willcocks et al., 2015B). Instead, the software is configured more like a logical flowchart, like a solution logic to a puzzle. Of course, this is what programming is fundamentally about, but RPA does away with syntaxes and languages, focusing on the core logic (Willcocks et al., 2015B; Lacity & Willcocks, 2015). In summary, RPA can be quickly deployed without a heavy skill set.

Secondly, RPA is considered "lightweight" IT when it comes to its design (Willcocks et al., 2015B; Lacity & Willcocks, 2015; Slaby, 2012). This refers to the magnitude of couplings between different systems: the robot does not e.g. write directly into a database, but rather uses the presentation layer of a software, it only has access to systems on a user-interface-level, just like a human. This is visualized in Figure 1. RPA does not disturb the underlying system (Lacity & Willcocks, 2015). Every action a robot takes can be easily logged; the risk of non-compliance is minimal (Lacity et al., 2015). This is different to most classic business process automations, which can e.g. manipulate data directly in a database or "under the hood".

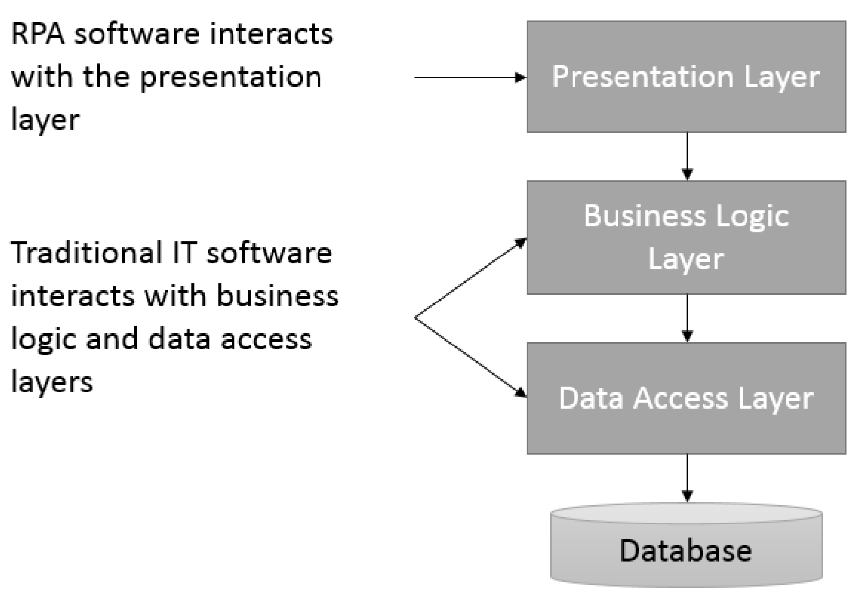


Figure 1. The layers of software. Adapted from Willcocks et al. (2015B).

2.1.1 Automatable band

With a solid grasp on the basic meaning of RPA, it is natural to follow with the question where and when the concept is applicable. As with all automation and programming, robots need explicit rules to follow (Lacity et al., 2015), which effectively rules out non-rule-based processes as feasible for RPA (or any other automation). Lacity et al. (2015) define the best-suited target processes for automation to have high transaction volume, high level of standardization, well-defined implicit logic, and high maturity. These aspects result in the best payoff for a process when automating it. High volume means high savings in working time, standardization and suitable logic enable smooth development and robot configuration, whereas maturity means the process is most likely not going to disappear and, thus, void the resources used to develop the RPA solution. Willcocks et al. (2015A) add to this list, noting repetitive tasks are suitable candidates because repetition is often a cause for human errors.

To illustrate the suitability for automation, Lacity et al. (2015), studying RPA implementation at Telefónica O2 in UK, have introduced a concept called automatable band, seen in Figure 2. It visualizes the relationship between transaction volume and process length. The underlying logic is that a process needs to reach a certain time saving for it to qualify as a candidate for automation. This saving can either come from a low volume of lengthy processes, a high volume of short processes, or a combination of the two ends.

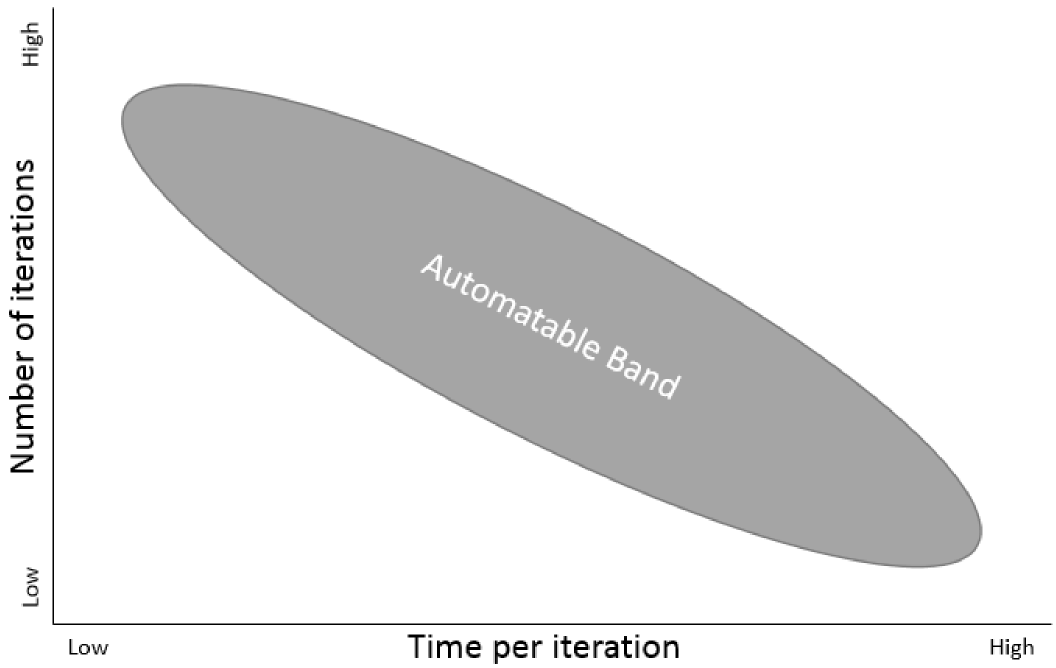


Figure 2. The automatable band. Adapted from Lacity et al. (2015).

It should be noted, that the automatable band applies to all automation, not necessarily just RPA. RPA is, in fact, just one automation tool in a company's repertoire. (Lacity et al., 2015)

One example of a more traditional tool of automation is Business Process Management (BPM). This concept is an approach that aims to analyse and improve the activities of a company, in a continuous manner for increased improvements. BPM is most often used in such areas of operations that produce a large share of added value; that way, even a smaller improvement can result in large gains in productivity. BPM relies on mapping and documenting activities, measuring their efficiency and trying to continuously tweak, improve and optimize the processes. To summarize, BPM is a comprehensive method to wholesomely map and optimize business processes. (Zairi, 1997)

RPA and BPM are different in many aspects. In terms of business, the goal for BPM is to reengineer processes, meaning e.g. introduction of new components of sub-processes. For RPA, this goal is to automate what already exists, to model an existing process as is and then perform it using a robot. The technical outcome, then, for RPA is that no new applications are required (other than the RPA software). BPM, on the other hand, often introduces new software for the automation. BPM has a deeper access to the back-end of systems, whereas RPA utilizes the user interface layer without the need for a direct connection to e.g. a database. As mentioned before, the competences required for RPA do not include programming skills; the developer only needs to have knowledge on the logic of the process. In BPM, the developers are software engineers, with the ability to create tools for users but no knowledge of the target process itself. Consequently, BPM developers need more experience due to the vast number of different concepts they need to master. Finally, the requirements for testing differ: again, RPA is lighter with the need to only verify outputs; BPM as a new software needs system testing. The degree of component reuse in both BPM and RPA is high, though in BPM this is more difficult; in RPA reusing elements is very agile. Table 1 summarizes these aspects. (Lacity et al., 2015; Slaby, 2012)

Table 1. BPM and RPA compared (Slaby, 2012; Lacity et al., 2015)

Attribute	**BPM**	**RPA**
Business Goal	Re-engineer processes	Automate processes
Technical Outcome	New software	Automated process using current software
Integration Method	Heavyweight; Access business logic layer	Lightweight; Access user interface
Developer competence	Extensive; software development	Modest; process expertise
Testing Requirements	System Testing	Output verification
Component re-use	High, difficult	High, easy
Payback time	Years	Months

As stated earlier, RPA can be described as “lightweight” IT (Willcocks et al., 2015B; Lacity & Willcocks, 2015; Slaby, 2012); BPM, on the other hand, is a lot heavier, requiring more effort and resources to implement (Lacity et al., 2015; Slaby, 2012). Willcocks et al. (2015B) visualize this difference with a chart (Figure 3). This figure highlights the differences between RPA and other traditional process automation tools: the skillset required for RPA is different from BPM, and they require different amount of resources. RPA is better fitted for processes that require fewer resources and focus more on process expertise. The chart also emphasizes how RPA and BPM fit together in a continuum – instead of replacing each other, they are complementary concepts (Willcocks et al., 2015B).

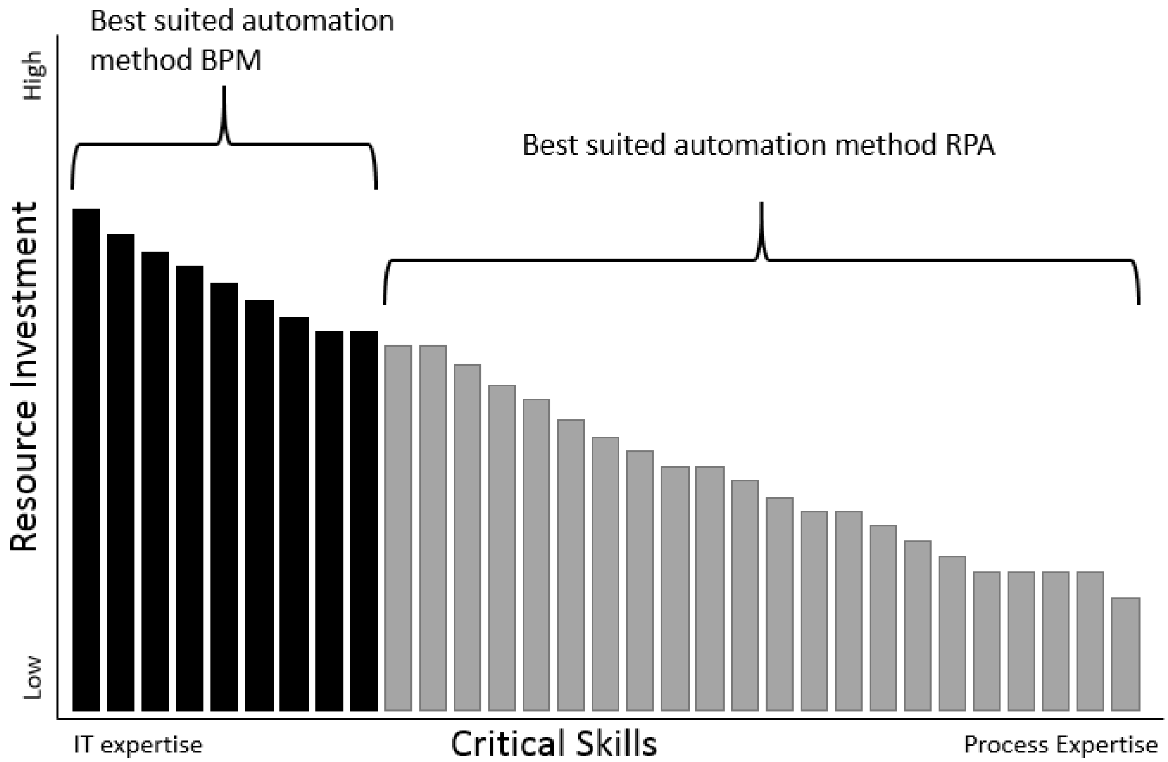


Figure 3. Adapted from Willcocks et al. (2015B)

Focusing on a single part in automation is not beneficial. While BPM and RPA complement each other, they are still only parts of a greater whole. It is important to look at the overall technology strategy and treat RPA only as part of that strategy (Willcocks et al., 2015A).

2.1.2 RPA and Change Management

RPA is all about replacing humans in performing repetitive tasks, and often implementing robots is associated with Full Time Equivalent (FTE) savings. Therefore, it is easy to understand why media uses headlines that speak of humans losing their jobs to robots, and why people are scared and anxious about RPA (Slaby, 2012; Lacity & Willcocks, 2016). It is, however, only a misconception: in reality, robots are meant to replace humans only in the most tedious and repetitive parts of the process, and the saved FTE can be directed to other tasks which robots cannot handle (Lacity et al., 2015; Lacity & Willcocks, 2015).

Introducing RPA means a shift in the content of work. Robots can be used to amplify and augment human strengths, meaning humans will do what they can do best: handle all the unstructured parts of processes, e.g. data that needs deduction beyond rudimentary logic, something only a human can do (Lacity & Willcocks, 2016). Lacity & Willcocks (2016) highlight this with an example of reporters:

Assignments that require linear and repetitive reporting, such as corporate earnings, were disliked among reporters. Automating these tasks, the company was able to shift the focus of its reporters to covering more creative stories. Using robots, not only did the number of earning reports go up ten-fold, but it was done at no extra cost, customers were happy with sped up delivery and improved quality. All reporters kept their jobs and were able to concentrate on more appealing aspects of their jobs.

Employees are wary about how automation will impact themselves, and it is imperative managers do not simply ignore this aspect of RPA (Lacity & Willcocks, 2016). The shift will be visible especially in talent development. Considering how RPA can be developed and where those skills can be acquired is, naturally, important, but it is also important to keep in mind the shift in capabilities of the retained human workforce. For example, if robots perform all the repetitive tasks, the work content for humans will shift more towards creativity and problem-solving, resolving error cases (Lacity & Willcocks, 2016). A plan must be made to redeploy these resources, otherwise possible FTE savings will not be realized properly, with nothing to do for the resources freed by RPA (Slaby, 2012).

As a novel technology with radical impacts on the life of employees, RPA is bound to encounter change resistance from the organization. The organization implementing RPA must be alert to see possible mismatches with the existing culture and structure of the organization. There are a number of barriers that have to be overcome. Organizations with more suitable culture, e.g. in terms of transformation capability, have an advantage and are more likely to perform well in the implementation phase (Willcocks et al., 2015A), but with proper management these barriers for other organizations can be lowered significantly (Willcocks et al., 2015B).

To strengthen the buy-in from the entire organization and properly enable strategic service automation, the automation project requires support from senior management (Willcocks et al. 2015B; Lacity & Willcocks, 2016). It is not only enough to develop strategies, the management must also enable the execution. One way to do this is for the project to have a sponsor or a champion – a visible spokesperson who will actively try to sell RPA to the organization, not least by actively trying to battle obstacles and lower barriers that threaten the progress (Lacity & Willcocks, 2016; Willcocks et al., 2015A). Lacity & Willcocks (2016) give an example, where in a hospital environment the process automation champion learned of a tedious and time-consuming task, searching product specification data online. He proceeded to automate this task with first priority. By doing this, not only did the hospital save in terms of resources freed, but also by lifting a painful and visible task from the operations, the employees learned first-hand of the benefits RPA can offer. This created enthusiasm and helped further the automation a great deal. The sponsors are often people not from the IT department, but from the business side of operations (Willcocks et al., 2015B).

It is necessary to pay attention to the internal communication surrounding the implementation of RPA (Willcocks et al., 2015A). Willcocks et al. (2015A) describe how: in their communication strategy, a company planned ahead and tactically countered the fear of RPA. The message management wanted to give was that RPA would allow people to shift their focus at work based on their own interests, promising that everyone would keep their jobs. Communication also ties in with the previously mentioned project champion; the role is an essential tool in conveying the message effectively (Lacity & Willcocks, 2016).

A strengthened buy-in does not only make the initial deployment more fluent, it also helps in generating follow-up projects. To combat change resistance and help overcome their anxiety over RPA, the company should help employees understand the benefits service automation will bring (Lacity & Willcocks, 2016). With employees knowing what process automation can achieve and what the benefits are, they will be happy to give further suggestions for possible target processes (Willcocks et al. 2015B). An example of this from Willcocks et al. (2015A), shows that instead of fearing RPA, a team welcomed the robot as a team member, even giving it a name "Poppy". Eventually, they asked if "Poppy" could be trained to handle more work.

2.1.3 Skillset

A notable aspect of RPA is the ease of developing and modelling processes. Developers do not require programming skills, but rather need to be acquainted with knowledge about business processes (Lacity & Willcocks, 2015). In practice, the developer does not program the robot, but rather teaches it the process logic to follow. This includes rules and instructions on what keys to press and how to handle certain exception cases (Willcocks et al., 2015B). Since the skillset is not associated with traditional IT, it is not wise to locate the development to IT specialists, RPA must be managed as a business and operations project (Willcocks et al., 2015B; Lacity & Willcocks, 2015). RPA can benefit more from a multi-functional team, involving the users from operations, IT, developers and even suppliers, working together on the development of both the process and the solution (Willcocks et al., 2015B).

While the skillset is different and somewhat "lighter", it does not mean training would not be required (Lacity & Willcocks, 2015). People and their roles within the organization must be well-defined and appropriate training must be applied to ensure proper functionality (Willcocks et al., 2015A; Willcocks et al., 2015B). Operational responsibilities include exception handling, testing, system support, process support and product support (Willcocks et al., 2015B). For each task, it must be clarified which role or function is responsible and each role must be properly trained (Willcocks et al., 2015B). The need for teaching and learning is especially high right after implementation (Willcocks et al., 2015A), but it is important to continue the support beyond implementation (Willcocks et al., 2015B).

2.1.4 Target process selection

The first step of automation should not be automation itself; rather, care should be taken to select suitable processes. To this end, the implementation should begin with questions such as should this process exist in the first place. Process elimination, optimizing and simplifying should be the first steps; this is done to avoid automating redundant tasks and allow focusing on the tasks that benefit most from RPA. (Lacity et al., 2015)

All in all, in the entire RPA implementation, there are three major phases on the way to production. First, the process is combed through for any parts that could be eliminated, improving the efficiency of the process. This way, all other gains in productivity are exhausted before relying on automation. Next, a proof-of-concept type of project is carried out, to see whether RPA actually works with all the required systems the organization needs. Finally, the rollout for RPA happens, covering everything from documenting the target process to handing the configured RPA solution over to the users. (Lacity et al., 2015, Willcocks et al., 2015A)

Lately, there have been significant advances in artificial intelligence (AI), which in long-term can further revolutionize business process automation, and change the selection criteria for which process to target with RPA. This is especially true with unstructured, shapeless data or processes without implicit logic for the robots to follow. (Willcocks et al., 2015B)

2.1.5 Scalability

An important aspect of RPA is its scalability. RPA is easy to develop compared to traditional BPM tools, but on top of that, it is more easily scalable. This is crucial for BPO providers, who already in their business are seeking similar strategies in their business processes – the ability to define a process and reuse it for multiple customers (Slaby, 2012). To enable scalability, it must be included in the implementation and development strategy (Willcocks et al., 2015B). To get the maximum out of the robots, they must be multi-skilled (Willcocks et al., 2015A). Internal infrastructure must grow in pace with automation, to keep up with the demand in terms of resources, and the internal technical architecture must make this possible (Lacity et al., 2015; Willcocks et al., 2015B). This way, the software can be easily scaled up and down to meet changes in workload (Lacity & Willcocks, 2016). The scalability must be a culture and should be adopted throughout the organization (Willcocks et al., 2015A).

2.1.6 Benefits

In the future, technological advances will play a major role in service industries. Service automation can deliver multiple benefits (Lacity & Willcocks, 2016), some of which are summarized in Table 2.

Table 2. RPA value delivered. Adapted from Willcocks et al. (2015B).

<table>
<tr><th>Company</th><th>Automated processes</th><th>RPA transactions per month</th><th>Business Value</th><th>Return on Investment</th></tr>
<tr><td>Telefónica UK</td><td>35 % of back office (15 processes)</td><td>400 000 to 500 000</td><td rowspan="3">• Faster delivery
• Better service quality
• Higher compliance
• Unbeatable scalability
• Strategic enablement
• FTE avoidance
• FTE redeployment
• FTE savings</td><td>650 % - 800 % - 3 years</td></tr>
<tr><td>Utility</td><td>35 % of back office</td><td>1 000 000</td><td>200 % - 1 year</td></tr>
<tr><td>Xchanging</td><td>14 core processes</td><td>120 000</td><td>30 % per process</td></tr>
</table>

These are not only financial benefits, but the robots also help in improving service speed and quality. The robots can work 24-hour days without tiring, further increasing service availability and delivery. Another benefit is a decrease in number of errors, as robots can repeat tasks without lapses in concentration. For example, in the Telefónica O2 UK case the number of follow-up calls due to mistakes and slow response times were reduced by over 80 % (Lacity et al., 2015). RPA can automate services, shortening the processes by 60 % and increasing the accuracy. This, in turn, leads to higher customer satisfaction, while costs are also cut with savings of 25-50 % reported (Bals et al., 2015). A typical figure for return on investment during the first year of RPA is 30 % or more (Lacity & Willcocks, 2016). Telefónica O2, as of April 2015, had deployed over 160 robots, which collectively process from

400 000 to 500 000 transactions monthly. This yielded a three-year return on investment between 650 and 800 %.

While RPA is often considered a method for cost savings and diminishing head count, these cuts have not realized in many projects (Willcocks et al., 2015A; Lacity & Willcocks, 2015). Successful implementations show evidence that while the robots do replace humans in areas they are implemented in, it is possible to redirect and refocus the human resources to other tasks (Lacity & Willcocks, 2015), reusing the employees in other areas without the need to let anyone go.

2.2 Projects

Developing a RPA solution is a process that has a distinct beginning and a distinct ending. It begins with the idea of using RPA to a target process, need of the solution arises from the business operations. It ends when the RPA solution has been developed and taken into use. Therefore, it is reasonable to consider RPA implementations as projects.

2.2.1 Project Life-cycle

The definition of project states that it must have a timeline with an ending and a beginning (Artto et al., 2006). This definition highlights the temporal nature of projects, but at times, it is beneficial to include stages from before and after the project itself. This leads to one definition for project life-cycle, with three stages (visualized in Figure 4): work stages preceding the project, work stages during the project, and work stages following the project (Parr & Shanks, 2000; Artto et al., 2006).

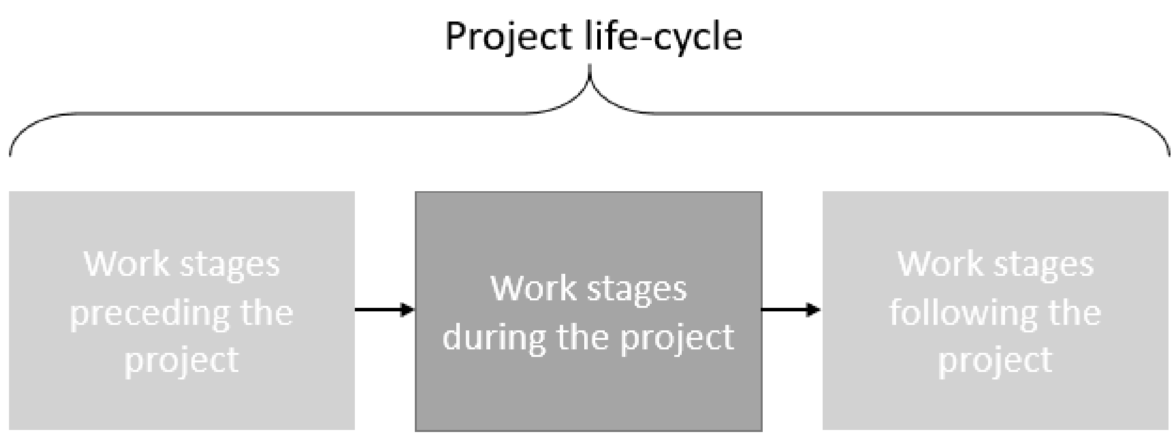


Figure 4. Project life-cycle. Adapted from Artto et al. (2006).

The actual project comprises many phases. Multiple authors (Pinto & Slevin, 1988; Parr & Shanks, 2000; Hartman & Ashrafi, 2002) refer to this project work phase as the project life-cycle, not including the preceding and following stages of the project.

The work in the project is divided in four major stages (Pinto & Slevin, 1988; Hartman & Ashrafi, 2002; Artto et al., 2006): Conceptualization, Planning, Execution and Termination, visualized in Figure 5, and their contents summarized in Table 3. Conceptualization is the stage that includes specification of the project. One major question in this phase is the need for the project – why is it needed, what goals and objectives it should have.

In planning phase, all the activities required in the project are defined and scheduled. This step outlines the timeframe for the project, as well as setting the resource requirements.

Execution is when the actual work takes place. For the users, this is the most visible and tangible part of the project life-cycle.

The last phase is termination, which comprises closing the project and handing it over from the project team to the users or owners. This step includes finalizing project documentation and transferring all the knowledge to the owners.

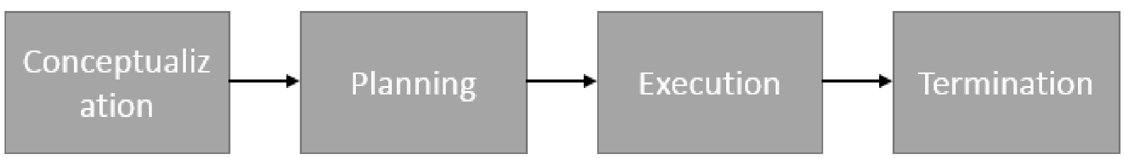


Figure 5. Implementation phases of a project.

Table 3. Project stages and their contents summarized.

Project Stage	**Contents**
Preceding	Ideation, selection of provider
Conceptualization	Specifications, scope
Planning	Scheduling, plan, resource allocation
Execution	Work on the project
Termination	Wrap up, finalize documentation, transfer knowledge
Following	Support use, additional work

Parr & Shanks (2000) describe in more detail the stages of the project of an ERP implementation contains (Figure 6). The work during the project is divided in six steps, set up, re-engineering, design, configuration & testing and finally installation. Parr & Shanks have, in description, the conceptualization and planning phases merged as one and the finalization phase is called enhancement.

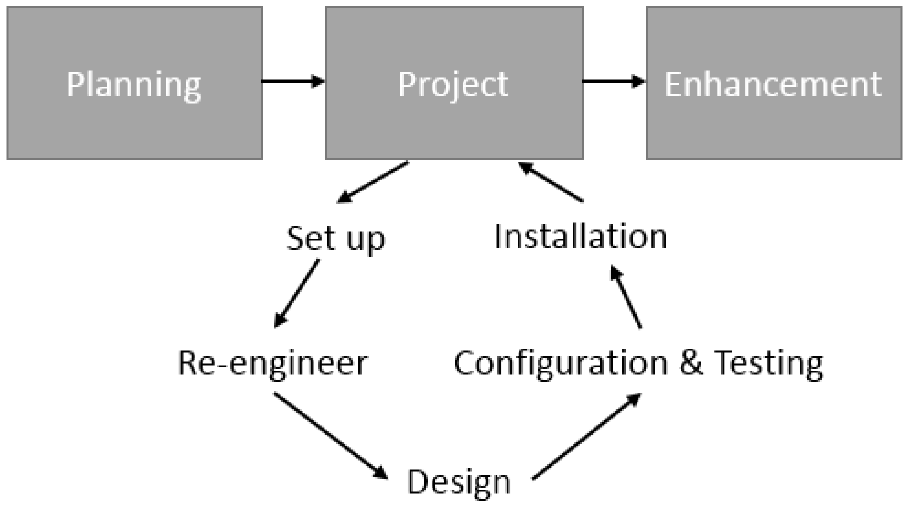


Figure 6. Phases of an ERP implementation. Adapted from Parr & Shanks (2000).

Cooper & Zmud (1990) consider an IT project to consist of six stages (Table 4). Initiation, adoption, adaptation, acceptance, routinization and infusion. Initiation is the phase of identifying a need and a possible area for implementing an IT solution. Adoption contains the decision-making process for laying foundation for the project and deciding what course to take. Adaptation, then, is the stage where the application is configured and implemented. Furthermore, this is when processes within the organization are refined to fit together with the new solution. Next, in acceptance, the installed application is embedded to the organization. Routinization takes this even further, with the solution becoming increasingly integral part of the organization. Finally, infusion is when the solution begins to show its benefits, supporting organizational activities.

Table 4. Stages of an IT project, based on Cooper & Zmud (1990).

Project Stage	**Contents**
Initiation	Identify need and area for implementation of solution
Adoption	Decide on solution, project and its scope
Adaptation	Configure and implement solution
Acceptance	Install and begin use
Routinization	Integrate solution
Infusion	Extract benefits using solution

2.2.2 Project Success

Defining project success is a necessary step to take before discussing successful projects. There are two important concepts here: project management success and product success. *Project management success* is the degree to which a project and the project process fulfil their goals, mainly cost, time and quality. In addition, the clients must accept the project. *Product success* relates to the aspects of the final product, mainly how its requirements are met. It is important to emphasize that these two dimensions have different success factors, and either one can fail while the other succeeds. (Pinto & Slevin, 1988; Baccarini, 1999)

These two dimensions of success are very similar in many ways and thus easy to mix up. However, it is important to distinguish project management and product success from each other. According to Baccarini (1999), there is a hierarchy to the two types of successes: product success is more important than project management success. This follows directly from their definitions – product success is what the users of the project deliverable think, while project management success is considered to be how the project management feels. Arguably, the opinion of the users is more important than that of the management team, since it is the users that use the product, after all. This hierarchy is outlined in Figure 7, where *goal* and *purpose* of the project are categorized under product success while *outputs* and *inputs* are tools of project management success.

The *goal* is defined as how the project will be able to help the organization reach its strategic goal. What the goal provides, is the reasoning for why the project, as part of a bigger picture, is needed. The *purpose* of the project is a slightly shorter-term objective of the project. In terms of hierarchy, purpose is lower than goal: the purpose provides the tacit tools for achieving the goal. The project output is the project deliverable in its traditional sense, i.e. the concrete and tangible output of a project. Project inputs are the resources that are used in order to complete the project. These resources can be e.g. monetary and can be described using the three traditional measures of project: scope, schedule and cost. Again, these aspects can be organized as a hierarchy: outputs serve as a tool to complete purpose, while input, similarly, is a tool to complete the outputs. This hierarchy is emphasized in Figure 7. (Baccarini, 1999)

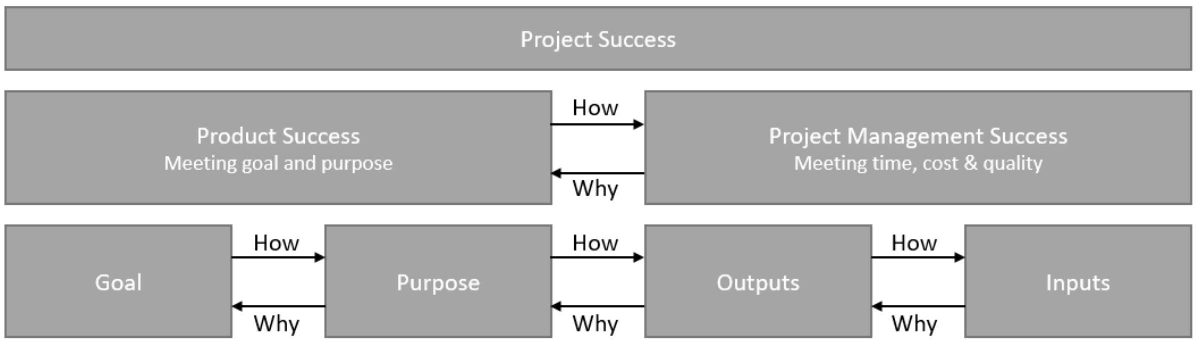


Figure 7. Hierarchy of the project success (adapted from Baccarini, 1999).

Project managers easily focus on making the project successful in terms of project management success, disregarding product success. This approach is understandable: long-term goals, such as user happiness, can be difficult or even impossible to measure during the project. Focusing only on short-term goals such as budget can be more beneficial in steering the project, but might not lead to successful product implementations. A project can be a success even if it fails in terms of product success.

2.2.3 Critical Success Factors

In project management, Critical Success Factors (CSF) are aspects of a project that, if adequately satisfied, will enable a project to be completed successfully (Leidecker & Bruno, 1984). In general, CSFs are just tools to direct management attention to the right areas of the project. Projects tend to succeed or fail based on the attention the CSFs are given (Wateridge, 1995). In research, there is some agreement that certain factors are more common and more influential in projects (Wateridge, 1995). However, due to the variation and differences between projects, not all factors are universal (Wateridge, 1995), and thus, a comprehensive list of CSF is impossible to compose.

One of the more common factors is *top management support* (Pinto & Slevin, 1988; Järvenpää & Ives, 1991; Parr et al., 1999; Somers & Nelson, 2001; Hartman & Ashrafi, 2002). To enable proper completion of a project, the top management must be on the same side, enabling the execution.

Without the support from the top, necessary resources and authority to further the project are difficult to achieve.

Defining *clear goals* and objectives for a project is crucial (Pinto & Slevin, 1988; Wateridge, 1995; Somers & Nelson, 2001; Hartman & Ashrafi, 2002). This has a two-fold impact: with clear goals, it is easier to figure out if they are feasible or not, and they serve to guide participants towards right direction (Pinto & Slevin, 1988; Wateridge, 1995). From a reversed perspective, poorly defined, often conflicting targets lead to failure (Wateridge, 1995). Closely tied to goals is the *management of expectations*. Somers & Nelson (2001) mention this as a separate CSF, but it can be seen as part of setting and re-evaluating the goals as the project progresses.

One CSF for projects is *user involvement*. This aspect has many names, some more direct (user involvement (Wateridge, 1995), client consultation (Pinto & Slevin, 1988), user participation (Parr et al., 1999)) than others (owner's consultation (Hartman & Ashrafi, 2002), interdepartmental cooperation (Somers & Nelson, 2001)). The bottom line here is that for a project to succeed the users must be heard (Pinto & Slevin, 1988) and the project must be guided to meet the actual needs of the users. The users must be involved in the design of the system (Parr et al., 1999).

User satisfaction (Parr et al., 1999) is closely related to other CSFs, mainly user involvement and management of expectations. The need for the project arises from the users, so a priority goal should be to meet the needs and expectations of the users (Wateridge, 1995). Another name for this CSF is owner's approval (Hartman & Ashrafi, 2002), which while not being the exact same thing, conveys the same message: the project must be managed to meet its requirements and get user approval. If users are not satisfied with the results, they are not likely to utilize them to their full potential. This CSF also ties in with the product success dimension of project success.

Closely related to user satisfaction is *user acceptance* (Pinto & Slevin, 1988; Parr et al., 1999). While user satisfaction is mainly a passive phenomenon, user acceptance can be increased by "selling" the project to the users (Pinto & Slevin, 1988). One method of this "selling" is using a *project champion*, which can also be regarded as its own CSF (Beath, 1991; Parr et al., 1999; Somers & Nelson, 2001). The champion is usually a member of the user organization, tasked with advancing the project and battle any obstacles that could harm the progress. The champion does not necessarily need to be a person of authority, but must have enthusiasm, optimism, vision and talent for conflict management to drive the organizational change (Beath, 1991).

While some CSFs are more abstract, there is room for concrete and traditional tools, such as a *schedule* or plan for the project (Pinto & Slevin, 1988; Wateridge, 1995; Grover et al., 1995; Hartman & Ashrafi, 2002). Success in projects is attributed to working with a detailed plan, and failure can result from not having such a plan (Wateridge, 1995).

Project management is important, too (Pinto & Slevin, 1988; Grover et al., 1995; Wateridge, 1995; Somers & Nelson, 2001). Timely actions based on real-time information on the status of the project help steer a project towards success (Wateridge, 1995; Pinto & Slevin, 1988).

Without proper *skillset*, the result of the project cannot be utilized to its full potential. To this end, users need to be educated and trained (Nelson & Cheney, 1987; Pinto & Slevin, 1988; Somers & Nelson, 2001). Furthermore, attention must be paid to any changes in underlying business processes; users need to be made aware on how this change affects their work (Somers & Nelson, 2001).

In addition to the skillset of users, the *project team competence* must be on a proper level, too (Grover et al., 1995; Parr et al., 1999; Somers & Nelson, 2001; Hartman & Ashrafi, 2002). This covers both the skills and resources, i.e. team members must have proper skills but the team technological resources and their usage must be competent, as well. Additionally, the resources must be adequate (Pinto & Slevin, 1988; Parr et al., 1999). This means having *dedicated resources* for the project team (Somers & Nelson, 2001), to ensure smooth and continuous progress.

While the project managers do everything they can to ensure the progress in the project, it is impossible to avoid setbacks altogether. To this end, it is imperative the project team is able to handle unexpected situations; this activity is called *troubleshooting* (Pinto & Slevin, 1988) or *support* (Somers & Nelson, 2001).

One of the more important CSFs is *communication* (Pinto & Slevin, 1988; Wateridge, 1995; Somers & Nelson, 2001; Hartman & Ashrafi, 2002). Communication itself does not affect the progress that much; rather it enables other key areas (Wateridge, 1995). Properly conveying, for example, user expectations to the development team is crucial and requires sufficient communications practises. As an enabler, communication needs proper focus; project managers often focus too much on concrete measures of success (Wateridge, 1995).

A project often brings something new to the user organization. To adapt and implement the end result, *business process re-engineering* needs to be considered (Somers & Nelson, 2001). The same result can be achieved by having minimal customization in the project (Somers & Nelson, 2001) or delineating and re-designing processes already prior to the project (Grover et al., 1995).

Change management, both in the project and the organization around it, is important (Grover et al., 1995; Somers & Nelson, 2001; Hartman & Ashrafi, 2002). Especially large projects may require significant changes in the organization and its processes to enable proper implementation (Somers & Nelson, 2001).

Table 5. Summary of the most common Critical Success Factors.

Critical Success Factor	**Description**	**Source**
Top Management Support	Adequate support from the top management to drive project forward	Pinto & Slevin, 1988; Järvenpää & Ives, 1991; Parr et al., 1999; Somers & Nelson, 2001; Hartman & Ashrafi, 2002
Clear Goals and Objectives	Defining clear goals for the project team to strive towards	Pinto & Slevin, 1988; Wateridge, 1995; Somers & Nelson, 2001; Hartman & Ashrafi, 2002
Management of Expectations		Somers & Nelson, 2001
User Involvement	Involving users to the project and its design	Pinto & Slevin, 1988; Wateridge, 1995; Parr et al., 1999; Somers & Nelson, 2001; Hartman & Ashrafi, 2002
User Satisfaction	The users must be satisfied with the end result of the project	Wateridge, 1995; Parr et al., 1999; Hartman & Ashrafi, 2002
User Acceptance	The users must accept the end result of the project	Pinto & Slevin, 1988; Parr et al., 1999
Project Champion	A person who drives the project forward	Beath, 1991; Parr et al., 1999; Somers & Nelson, 2001
Project Schedule	Schedule for the project activities	Pinto & Slevin, 1988; Wateridge, 1995; Grover et al., 1995; Hartman & Ashrafi, 2002
Project Management	The activity of steering the project to right direction	Pinto & Slevin, 1988; Wateridge, 1995; Grover et al., 1995; Somers & Nelson, 2001
User Skillset	Educating users to the project	Nelson & Cheney, 1987; Pinto & Slevin, 1988; Somers & Nelson, 2001
Project Team Competence	Skillset of the project team and technological competence of the resources	Grover et al., 1995; Parr et al., 1999; Somers & Nelson, 2001; Hartman & Ashrafi, 2002
Dedicated Resources	Adequate, dedicated resources allocated to the project	Pinto & Slevin, 1988; Parr et al., 1999; Somers & Nelson, 2001
Troubleshooting / Support	Support in the case of problems	Pinto & Slevin, 1988; Somers & Nelson, 2001

Communication	Communication between different stakeholders of the project	Pinto & Slevin, 1988; Wateridge, 1995; Somers & Nelson, 2001; Hartman & Ashrafi, 2002
Business Process Re-Engineering	Re-designing processes to fit the project	Grover et al., 1995; Somers & Nelson, 2001
Change Management	Managing change of organization and processes	Grover et al., 1995; Somers & Nelson, 2001; Hartman & Ashrafi, 2002

In addition to these common critical success factors, there are countless more. Some of the additional factors can be included in a broader definition of these factors, or they are more specific to certain types of projects. For example, Somers & Nelson (2001) have based their 22 CSF list on the implementation of ERP systems. Listing all possible CSFs and their variations falls outside the scope of this thesis and, therefore, this list is not expanded further.

2.2.4 RPA implementation project lessons

This sub-section collects together relevant lessons for project management, from different RPA implementations.

Willcocks et al. (2015B) have detailed eight crucial steps for implementing RPA (Table 6). First, the organization should align RPA with its business. This means matching strategies and making sure RPA strategy is in line with the corporate strategy. Then, the organization around RPA should be clear and well-defined, including someone as main responsible, a role of Head of RPA. This does not only cover the roles, but also the design of implementation process. The design must be tailored to the organization's needs: a large and complex organization requires a more centralized RPA department, where as a smaller organization benefits more from a flexible and light model.

The third step is a continuation of the previous step; there should be a RPA Governance board, which is responsible for managing the demand pipeline for RPA. This way, the tasks can be properly assessed and prioritized, and the ones with most potential can be implemented first.

The fourth step comprises the entire internal RPA delivery methodology. A process should be agreed on, which is followed when a new process is under implementation.

A service engagement model should be implemented, to support the operational processes. This includes support for errors and exceptions and other system support. The roles for these functions must be cross-organizational.

In the organization, people should have clearly defined roles with set responsibilities. Training must be provided, to enable efficient operations.

RPA can be easily scaled up, but to enable it properly, it must be included in the implementation process from early on. Willcocks et al. (2015B) have two aspects they think are important for enabling scalability. In terms of technical environment, a scalable low maintenance infrastructure should be established. For example, if establishing new servers for RPA is easy, then multiplying capacity is possible to do on a short notice. Furthermore, scalability must be included in the design philosophy of the solution itself. This way, established solutions can be copied across different processes, decreasing the development time.

Table 6. List of key RPA factors. Adapted from Willcocks et al. (2015B).

Step	**Description**
Establish Business-RPA alignment	Define what is the role and meaning of RPA
Define the organizational design	Structure of organization, responsibilities
Form an RPA Governance board	The board to keep RPA wheels turning
Agree on RPA delivery methodology	Define the pipeline process for RPA
Establish support	Support for end-users in their problems
Define roles and responsibilities	Self-explanatory
Define a scalable environment	Self-explanatory
Plan for scaling	Scaling is one of the most important aspects: prepare to leverage it

It is important to assess what the software can and cannot do, where are its strengths and where its weaknesses. This is something that can be done with the aid of academic research (Lacity et al., 2015).

IT-department must be involved early in the RPA implementation process. There have been cases where omitting this has caused IT to create unnecessary resistance towards the implementation. (Lacity & Willcocks, 2016)

There is much misunderstanding about the attributes and features of RPA. This is the case especially with the relation between RPA and IT. IT has a huge role to play in the implementation of RPA, which means that keeping them in the loop about features such as governance and security will decrease the probability of unnecessary delays and obstacles. (Willcocks et al., 2015B)

The management should aim to enable continuous improvement past the initial implementation (Willcocks et al., 2015A). This notion has an impact on everything: the projects must not be about short-term gains, rather everything should focus on enabling long-term success.

Internal RPA capability should be built to evolve. Deploying RPA as a mere tool will not enable the capture of the full benefits; RPA should be implemented as a strategic resource. Deploying and leaving it as is will not give the full benefit. (Willcocks et al., 2015A)

2.3 Synthesis: Outlining a RPA implementation process

This section is dedicated to analysing and synthesizing different aspects of project life-cycle and success literature in terms of RPA, and then using the results of that analysis, to outline a model for the RPA implementation project process (henceforth *RPA process*). The RPA process focuses on the RPA solution implementation for a *target process*. In this study, target process refers to the process which is being targeted by RPA automation.

One key assumption moving forward will be that the robotic software, along with its infrastructure, is already installed. This means the RPA process developed here will not deal with setting up the software and the platform, but rather focus only on implementing the robotic solution to the target process. From this perspective, some steps in the process life-cycle or critical success factors will be redundant.

2.3.1 Roles

A clarification must be made about the roles in the RPA process. While the RPA literature does not explicitly outline definitive roles involved in RPA development, there are some that can be distinguished. A *RPA developer* is responsible for developing the RPA solution for the target process. This role must also have sufficient knowledge of what the RPA can or cannot do, to be able to judge whether RPA is the right solution for the case or not. There can be a single RPA developer in the project, or there can be multiple. The second role is the *end-user*, the one to actually use the developed RPA solution. It is not feasible to involve all of the end-users in the project, as there can be hundreds of users. Rather, the manager should select a few key people to first test and pilot the solution. Finally, there needs to be a *manager* from the end-user's side, representing the target organization in full-scale. The manager and the end-users combined are called *business* side. The roles are listed in Table 7, and their hierarchy is visualized in Figure 8.

Table 7. Roles in the RPA process.

Role	**Description**
RPA developer	Responsible for RPA know-how, development
End-user	User of the project deliverable
Manager	Manages implementation on business side

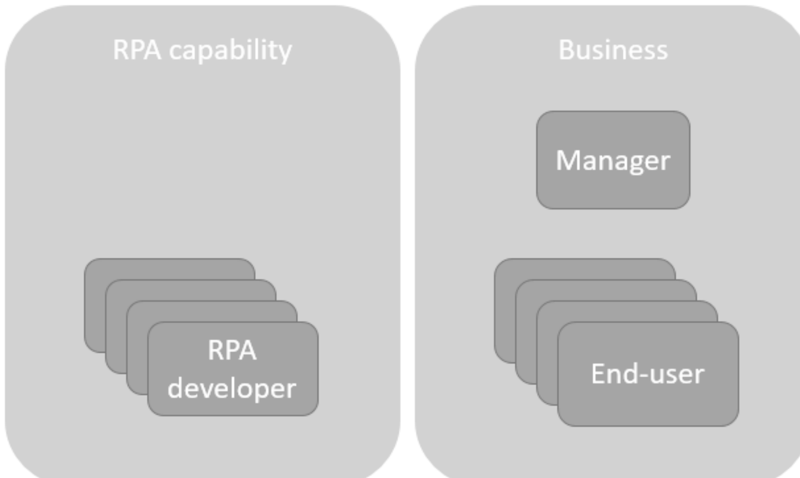


Figure 8. Roles and their hierarchy.

2.3.2 Life-cycle

To begin with, the process needs a project life-cycle, a timeline to anchor other nodes to. As a foundation, the large scale, abstract three step project process (where the process is divided in work preceding, during and following the project) described by Artto et al. (2006) will be used. The model offers a basis that is at the same time both abstract and concrete enough to serve as a basic foundation. The preceding and following stages of the process are necessary, because the RPA process is not independent, but a dependent part of the general RPA strategy. To detail the project implementation stage, it will be divided in four sub-stages, *conceptualization*, *planning*, *execution* and *termination*. Conceptualization comprises all specifications of the process, but also answering why the project is needed in the first place, what goals and objectives it will have. This phase covers also what Cooper & Zmud (1990) call *initiation* and parts of *adoption*, the former dealing with identifying what is actually needed. The latter of the two deals more with project management and the project scope, something that is included in the planning phase. This stage consists of scheduling the project and determining its scope, having already decided on what the solution will be. For example, in conceptualization RPA is verified as the most viable solution for the target process, then, planning is used to layout a schedule for developing and implementing the solution. Termination as a phase includes the conclusion of the project, it also marks the transfer of responsibilities from the project team to business.

Of these four activities, conceptualization and termination are most visible to the end-users, whereas planning and execution are more apparent to the RPA developers. These four steps and the overall project life-cycle are visualized in Figure 9.

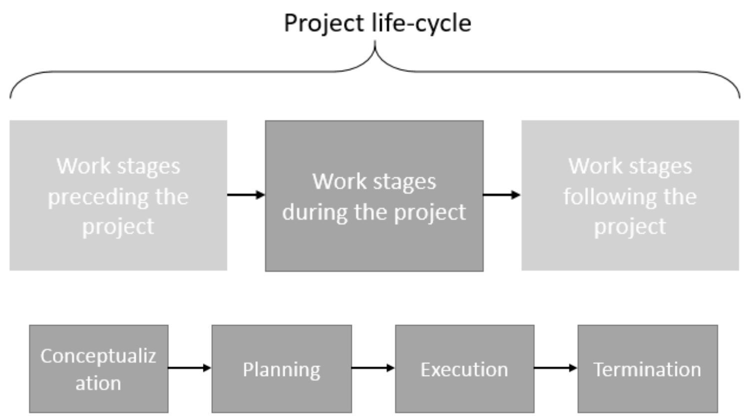


Figure 9. Project life-cycle in general form.

Drawing from both Parr & Shanks (2000) and Cooper & Zmud (1990), the activities in execution can be detailed further. Tasks *set up* and *re-engineer* (Parr & Shanks, 2000), can be regarded as part of setting up the system itself, in this thesis's case the robotic software and required infrastructure. However, as software set up is not considered to be in the scope of this process model, they will not be included in the model. A step for *design* should be included, a stage where the RPA solution is developed. This step is the most concrete of the entire process, as the target process is translated to a framework the robot can use to perform the tasks needed. Another key element to any software project is the *testing* phase. Development is never perfect, meaning errors are bound to happen, which is why testing is a necessary step to root out and fix errors. This step is done first just by the RPA developers, but later it is necessary to include the end-users to make sure specifications have been met. Design and testing steps should form an iterative feedback loop, allowing discovered errors in testing to be fixed by re-designing and re-configuring the solution.

A stage that Parr & Shanks (2000) describe as *installation* and Cooper & Zmud (1990) as *acceptance*, the activity where the solution is actually taken to use, should be included in the process. In the RPA process, this element will be called *rollout*. The name refers to the phase being gradual and interdependent, an iterative loop rather than a one-off task. In this stage, the RPA solution is introduced to the end-users and first piloted by a small sample to verify everything is working as intended. Then, if accepted by the users, the process is gradually rolled out to all the users, while still keeping an eye for any issues and fixing them along the way. The reason why this kind of iterative multi-stage testing is required, is because RPA is so closely knit to the underlying business processes. With more sensitive processes, there is simply no room for error. On top of that, to enable user acceptance, the process must be adjusted to fit the user requirements; the iterative nature of testing makes sure the users have been heard and their requirements are actually met.

The last two steps of Cooper & Zmud (1990) are *routinization* and *infusion*. These two steps continue further integrating the solution to the organization. It could be argued that these are vital to a successful implementation, but there are also arguments against including them into the model. For one, a project has a clear timeline: a start and an end, but these two steps are ambiguous, meaning their timeline is not easily defined. In addition, the responsibility of completing these tasks falls outside the project scope. It is up to the business to fully implement the RPA solution and reap all the benefits they can get. This RPA process is more about developing and delivering the RPA solution, not so much about long run tactics. These two steps will not be included in the process as is, but rather some of their content will be included in the rollout phase, as well as termination.

Figure 10 presents the outlined project life-cycle.

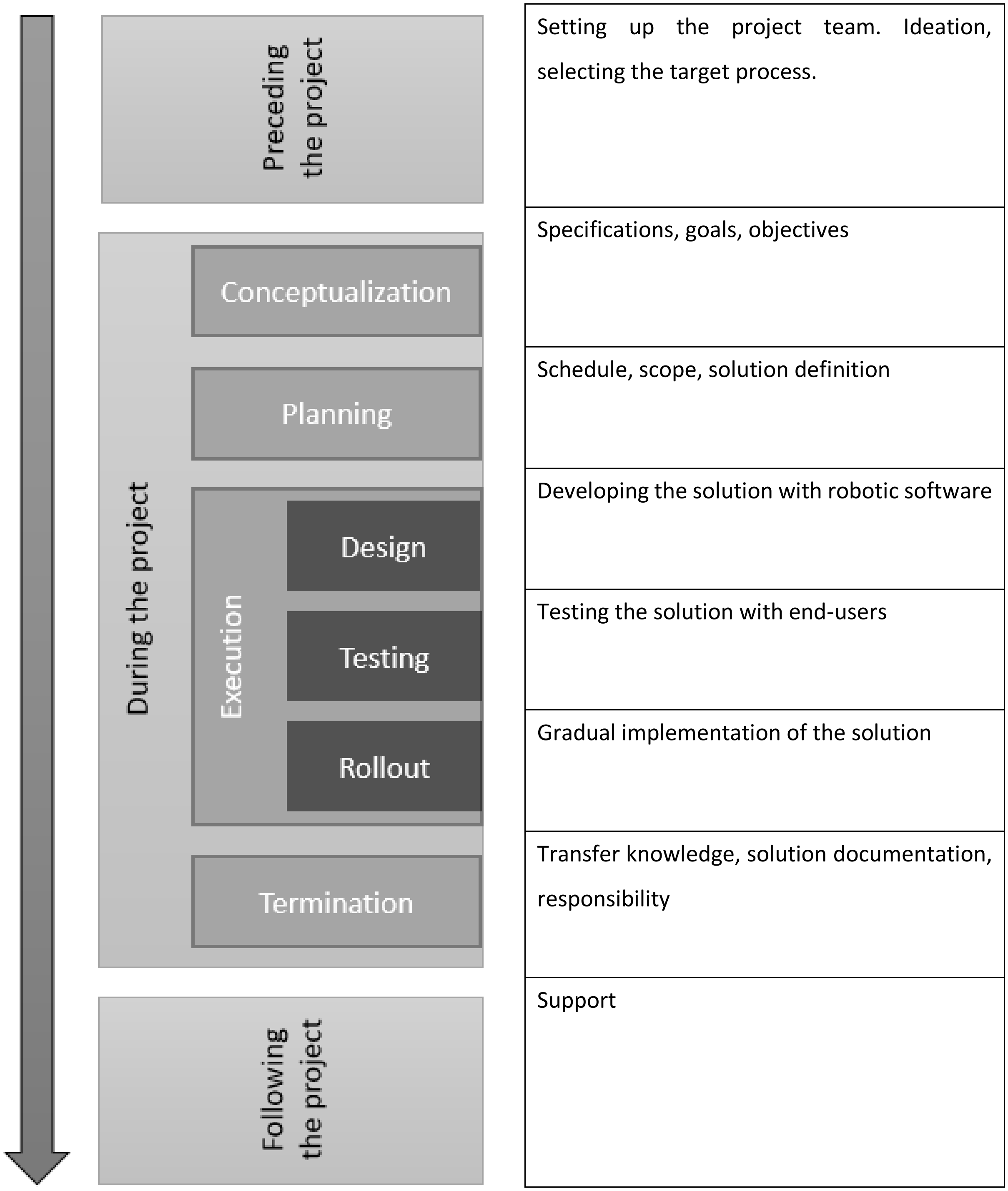


Figure 10. Project life-cycle and its stages explained.

2.3.3 Success factors

In this section, all of the CSFs discovered in the literature review will be meticulously examined, to determine whether they are relevant to RPA or not, how they will contribute to further project success, and how do they best fit in the RPA process. Combining the insights on RPA from the literature with project management and CSFs is what will result in a process model properly tailored for RPA projects.

Pinto & Slevin (1988) state that Critical Success Factors are both time sequenced and interdependent, and they can be laid down as a critical path. This means the CSFs are not just values that should be kept in mind, in the background, while developing the RPA solution: they can be translated into actions, stages or checkpoints in the project. The earlier a CSF appears in the RPA process, the more it should be treated as a checkpoint where failure will stop the entire project. Later on, resources have been invested and therefore it is not anymore viable to halt the entire project, but rather failed CSFs should be noted and necessary precautions should be taken to prepare for a possible failure of implementation. Some of the CSFs will directly translate into elements in the process, while others will convert into definitions or requirements that must be fulfilled.

Top Management Support is imperative in RPA. Without it, there are too many obstacles in the way, and a successful RPA implementation will be nearly impossible. While this factor could be an overarching CSF, it will not be included as such because, assumably, the top management of the organization is pro-RPA, and thus, the factor would fall outside the scope of this thesis. Instead, top management support should be made a checkpoint in *conceptualization*, where without support, RPA is not considered to be a viable candidate.

Defining *clear goals* is important for any IT related project. Computers are only able to follow strictly defined and formatted rules, therefore the goals must also be clearly defined. In IT, there is a term called *definition of done*, which means translating goals to a list of requirements which must be fulfilled for the project to be considered completed (Davis, 2013). This list can be formatted as a checklist, to further amplify its purpose as a set of targets. For RPA, the added clarity is equally important, as ambiguous goals can be difficult to translate to instructions for the robotic software. A definition of done would act as a tool to keep track of all the things the robot must achieve, for example what makes the target process considered completed. Implementing this tool reduces the risk of development derailing to a direction that does not drive the project towards completion. This CSF should be included as part of the *conceptualization*.

RPA projects are all about helping the end-users in their work, giving new automation tools to complete tasks faster and smoother. *User involvement* is, therefore, crucial to RPA projects. This factor is also closely related to *user satisfaction*, both of which can be satisfied by involving the end-users in the project. The end-users should be involved in translating the target process they perform to definition of done, which allows them to have a say in what is actually beneficial to automate, while also clearing any confusion about what the process comprises. The RPA developers need not have knowledge of the business side; with end-users involved they can rely on the translation of the target process provided to them. Allowing end-users to have their say also adds a sanity check in the process; if the target process being automated is somehow redundant, the users can express their opinion. Looking at this from another viewpoint, not including end-users would be like automating something unknown without any idea of value added for the process and project. End-users should be involved in the RPA process from beginning to end.

User acceptance is a crucial part in any project, just as in RPA projects. With the RPA solution being developed with end-users involved, risk of the deliverable not being used is significantly reduced, but not completely nullified. To further decrease this risk, a *project champion* should be involved in both RPA in general, but the RPA project at question, too. Having a person with a role or responsibility closer to the end-user enable the champion to have a stronger impact. Therefore, the champion should be chosen from the business side. In RPA projects specifically, the champion could be the manager, as it will be his responsibility anyway to make sure the implementation of the RPA solution is successful. The champion, with his RPA positive attitude, should be able to inspire the end-users to receive RPA as a positive and beneficial tool.

Besides the champion, user acceptance will benefit from clear and transparent way of working and delivering what is promised, on time. There is nothing RPA specific to this approach, rather it is something project management should consider in all projects. However, given the smaller size of RPA projects, the two traditional CSFs, *schedule* and *project management* are not as important as in projects in general. This is because both the project scope and organization size are small with only a few participants, meaning there is no need for high-level governance and management. Both these traditional factors are sufficiently built-in to the model of project life-cycle and the definition of each stage and considered rather self-evident, that they do not require any further attention in RPA projects.

Skills of individuals are involved in two CSFs, *user skillset* and *project team competence*. Again, the nature of the RPA projects as small and agile means there is less need for formal project management and, therefore, project team skills in the management area are not an absolute requirement. When it comes to RPA developers, they must have sufficient knowledge of RPA to be able to develop the required solution. In more detail, this entails required skills in using the RPA software as well as knowing how RPA works, both as a technical solution and as a process. One solution for acquiring this skillset could be training or studying for certificates of the software. For the end-users, their skill in using the RPA solution is a key enabler. Without proper training in using the new tool, the project will not be a success, resistance to change could be too high with a frightening unknown. Furthermore, making sure end-users are trained to understand what RPA is and what it is not, helps spark ideas for future target processes and development. Drawing together for these two skill-related CSFs, project team competence is partially critical, but does not fall in the scope of the RPA process specifically, whereas user skillset is a truly critical factor that must be accounted for. For end-users, it must be included in the rollout at latest, but is better fitted to work along the entire execution phase. This way, there is still enough time to start the required training. For developers, assumably they were educated before the project as part of a larger strategic implementation of RPA, which means their training only acts as a checkpoint at the very beginning, validating the viability of RPA as a method.

Dedicated resources can mean multiple things. One, it can be the availability of the developers, which, of course, is critical for the projects progress. Two, they can refer to the end-users and their involvement in the project, which can be, depending on the organization, complicated. End-users are often busy with their own work, and involving them in a development project can seem secondary in their own prioritization. Therefore, they should be clearly instructed and tasked to take part in the RPA project, and, if necessary their other workload should be lightened to counter this new duty they have.

Support and *troubleshooting* are important to any new and unfamiliar technology, and the same applies to RPA. Especially when testing the solution, end-users tasked with the trial should be given adequate and easily reachable support. This will decrease the mental barrier they might have in communicating their problems with RPA and the new solution to the project team and, especially, the developers. Additionally, fixing issues quickly allows building a lasting good impression of RPA, further selling the concept for the future. After the project has concluded, a model for support should be established as not to detach the business side completely. This entails mainly setting up channels for technical RPA support, dealing with any arising issue about the RPA solution. Support after the project should be distinguished from the support during testing and development; the RPA developers in the project might not be responsible for the support following the project.

A project consists of many individual parts that need to perform together for the project to succeed. To enable this, *communication* is critical. Different stakeholders must be able to, without considerable effort, communicate with each other, conveying critical information across the project team. Communication should act as a back-bone for the entire project, following through in every stage and every activity. With RPA development relying on iterative feedback, it only further highlights the importance of communication. Without low barriers for communication, the feedback will never reach the developers, leaving the RPA solution shy of its expected requirements and further leading to non-successful implementation. While the communication is mainly important during the project, the channels should be kept open even once the project has been concluded, to allow new ideas and any problems be communicated across the organization.

Business Process Re-Engineering, as stated in the literature, is important for any automation. The essence of this CSFs is to make sure the process is first optimized for RPA, not just automate what is given. For example, a process could involve moving files to a temporary folder for easier visual management for the end-user, but for a robot this step is not necessary and should be cut from the process. The business process re-engineering approach means involving the production as closely as possible, to make sure the developer understands the logic that is behind the business process. This way, instead of replicating non-optimized naturally developed routines, the target process for the RPA automation should be optimized to the T. It is possible that while analysing the target process with this mind-set, the process is found to be redundant completely, thus ending the RPA project early. Therefore, this step should be included in conceptualization as early as possible, as to not waste resources.

RPA is all about changing certain processes are performed, replacing human work with robotic one. Therefore, *change management* is a key concept in RPA projects. In some ways, it is integrated with other CSFs, such as user acceptance, but it is worth listing it as a separate part of the RPA process. The responsibility of having end-users accept change lies with the manager. This aspect is especially important because end-users can see RPA as a threat, and the fear of losing their jobs can complicate the implementation. With proper communication and change management this impact can be mitigated.

Concluding, there are some critical success factors that do not sit well with RPA, but a majority of them are relevant to RPA projects. Building from the preceding analysis, the RPA process model detailed in the previous section can be further elaborated by populating the timeline with each CSF. The relevant CSFs are presented in Table 8, along with life-cycle phases they best fit in. Figure 11 represents the same information in a visual form.

Table 8. RPA relevant CSFs and their life-cycle phases.

Factor	**Phase**
Communication	Overarching
Change Management	Overarching
Project champion	Overarching
User involvement	Overarching
Top Management Support	Conceptualization
Clear Goals	Conceptualization
Business Process Re-Engineering	Conceptualization
Dedicated resources	Planning
User satisfaction	Execution
User acceptance	Execution & Termination
Support & troubleshooting	Execution & Following
User skillset	Execution

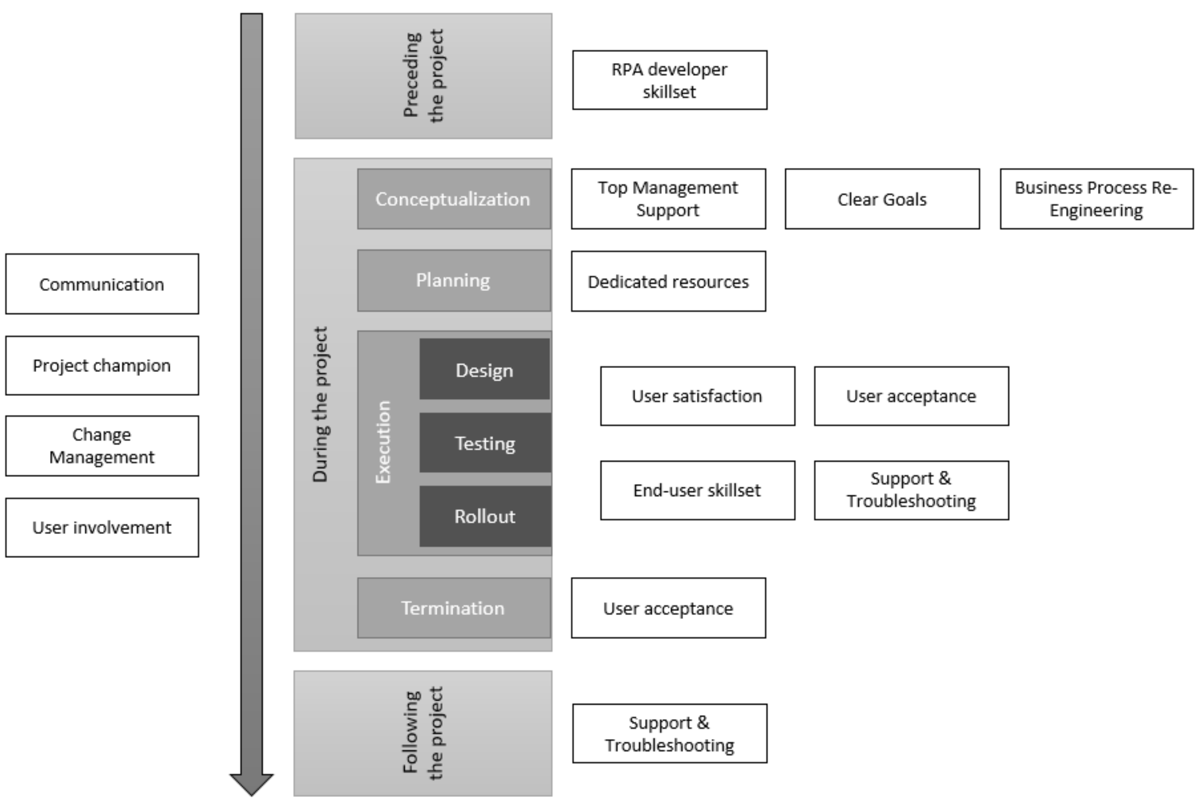


Figure 11. Project life-cycle and CSFs combined.

2.3.4 Other factors

Large portions of RPA implementation literature revolve around setting up the initial capabilities. For the most part, such lessons fall outside the scope of a single target process automation and, thus, also outside the scope of this thesis. However, there are some key notions that should be taken into account in the design of the RPA process.

One such aspect is the cross-organizational nature of the roles. While it is best to define set roles for the project, these definitions must not be too restrictive. RPA projects will always be cross-organizational; creating any additional barriers by defining roles too divisively should be avoided. Dividing responsibilities is necessary, however they should only act as guidelines. For example, RPA developers cannot withdraw to saying they only develop the solution based on translated process description, rather they must have an active role in facilitating the success of the project in all its areas. Same goes for business, who must realize RPA is beneficial for them, instead of trying to resist the change.

RPA is not about short-term success, but rather a tool for long-term gains. This theme should be kept in mind at all times, and decisions should be made in order to optimize long-term benefits, not short-term optimization. One concrete example of this mind-set is scalability, which is another essential theme to include in the RPA process. Scalable solutions might be more demanding to develop, but they will pay off in the long-term not having to re-invent the wheel with every target process.

2.3.5 Summary

In this section, the insights gathered from literature on both project management and RPA were fused together to form a model for RPA process. Doing so resulted in a process outline that is grounded on project management theory but has been enriched with knowledge of RPA. Furthermore, this model was then flavoured with observations based on literature on RPA implementations.

The model is not detailed to the finest level, but rather emphasizes crucial aspects and steps that need to be considered, in order to maximize the chances of success for the RPA solution development and implementation in the target process.

3 Research Design

This section will outline the course of the research and its key components. First, the reasoning for selecting a case study as the approach is presented, and after that the case project is selected and introduced. Further down the line, data collection and analysis methods are covered.

3.1 Case study research

Based on the research questions and the nature of the phenomenon in question, a case study design was chosen. A case study is applicable also because RPA implementation is difficult to detach from its context. With phenomena where such boundaries are not clear, the most suitable approach for studying them is a case study (Yin, 2003).

3.2 Case selection

Once a model has been developed, it will be used in an actual RPA implementation case. This is to test the model in a real setting, with the goal of critically analysing it afterwards, noting the strong points and the shortcomings.

The testing itself will take place in a company which has recently adapted RPA as a new method in their toolbox. This guarantees that the initial RPA capability is already set up, and allows the research to focus on implementing the RPA solution on a target process, rather than getting tangled with setting up the infrastructure itself.

The case is selected based on two main criteria. Firstly, the case itself should be small and simple enough for RPA to be implemented within a feasible timeframe and to fit the scope of the thesis. This simplicity also guarantees independency of the case; the more complex the process selected, the more there are dependencies on other processes and other factors, which makes the scope of analysis rather wide. A smaller scope, in this case, allows for a more outlined analysis. Secondly, the target process should fit the feasibility requirements of RPA; in essence, the process must be automatable with RPA, otherwise testing the entire process model is impossible.

Using these two criteria, a process of reporting international travel expenses of a customer to a travel expenses system was selected as the target process for RPA development. The input for the process is the travel data in Excel sheets, sent by the customer via email. Each member of the customer's staff has their own data file. The files are sent monthly in the beginning of a month, containing the previous month’s travel data. Using these files, the personnel’s’ travel data is interpreted and their journeys during the month are entered under the employee’s name in the travel expense system.

The process comprises mostly of copying incoming information based on a pre-determined logic, which makes it independent and rather simple. The process does not require too much thinking; for the end-user, it is tedious and repetitive. While these attributes make it an annoying process for the end-user, they are also the reason for its automatable nature. Figure 12 illustrates the course of the process.

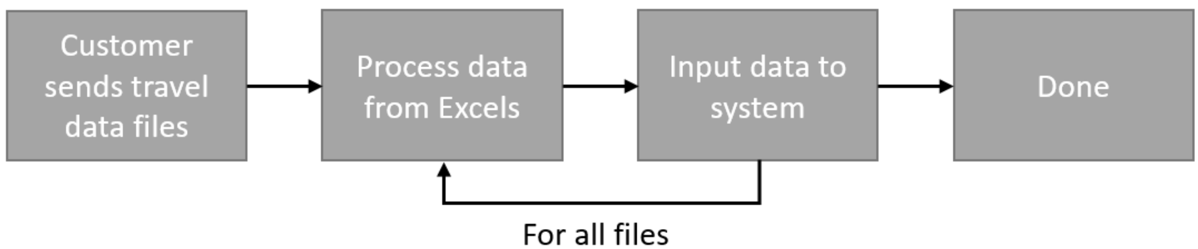


Figure 12. The course of the target process.

Furthermore, this process deals with a system that does not directly allow importing data. It would be possible to enable a feature for import, but that would be too expensive in relation to the benefits. In this case, the RPA is more efficient and suitable for the process and system.

3.3 Data collection

The backbone of data collection for the test case will consist of writing memos as the project goes on. Corbin & Strauss (2008) detail this method, where the researcher should both take field notes during the events itself, and after each session they should write memos, essentially recording the train of thought for later analysis. This forces the researcher to work with concepts and refine thinking along the project. In addition, this enables a periodical recording of the work, which is beneficial especially in a longer project. Memos are a flexible tool: they can be about both data collection and data analysis. The purpose for the data is to give insight on how the RPA implementation actually progressed, and the memos are able to do just this.

Memos were written throughout the test case, in total 13 of them. They were written by me, i.e. the RPA developer in the project, and they are mostly a first-person narrative. Each memo is dated by its writing date and the memos contain detailed description of the progress of the case project. The memos, on top of describing concrete events, include thoughts and insights on how the project is doing in terms of success and how the developed RPA process is fairing in each step of the way. For example, memo 3 depicts the kick-off meeting of the project and the thoughts and feelings after the meeting. Based on the scope of the thesis, that the design philosophy of RPA processes with any programming related practices is considered outside of the scope, the memos do not contain detailed records of the development itself. However, whenever the decisions in development had any relevance to other areas of the project, their descriptions were written down

To complement the memos, project documentation and workshop proceedings were archived; they provide an objective and non-biased counterbalance to the memos. The different data types are summarized in Table 9. The process documentation of the target process in one of such documents, it can be found in appendix 1. In addition, the instruction document provided by the RPA developers as a short instruction manual to use the RPA solution was archived, and is attached as appendix 2 The case project was kept lean and agile, meaning there are no comprehensive meeting agendas or notes other than what is mentioned in the memos.

Table 9. Summary of the collected data.

Data type	**Description**
Memos	13 memos detailing the project progress
Target process description	Detailed step-by-step description of the target process and its course
RPA solution instructions	Short instructions for the end-users to aid in the use of the RPA solution

3.4 Data analysis

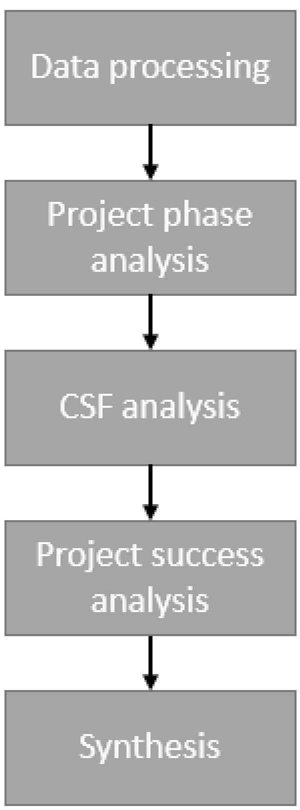
The memos were already directly in a proper format to begin the analysis, there was no need to transcribe or reformat the data on their part. The first phase of the analysis was to code the data. Coding means taking the data and extracting concepts and meaning out of it (Corbin & Strauss, 2008). Initially, the codes included only the phases in the project and the over-arching CSFs, but more of these categories were added as coding progressed and a pattern was starting to emerge. With coding done, the data snippets were then grouped based on the assigned tags. This way, mentions of each project phase were grouped together to form a more coherent picture of what had actually taken place in the phase in question.

Next, the grouped data was analysed by project phases. Insights from the literature were compared with the results from the test case. The contents of each stage in the project were described to paint a full picture of what each step contains. After reaching saturation with this analysis and having proper understanding of the project lifecycle, the analysis moved forward to include the over-arching CSFs. Again, the grouped data was used to describe and elaborate on what the role of each of the CSFs had been in the test case, and this role was compared to that of the literature and theory.

Then, the success of the case project was analysed, this time the data was grouped and filtered by categories relevant to project success. This analysis was required to understand whether the project was a success or not, as it would have implications for the analysis later down the line.

Finally, as a summary based on all other findings, the original RPA process model was re-visited. This way, any modifications and improvements could be added to finalize the proposed model. Basically, each step of the project and each CSF were re-visited, to see whether there had been any problems or additions that should be made to the process.

The analysis process is summarized and visualized in Figure 13.



- Formatting data - Coding data based on elements in the process model
- Categorizing data based on project phases - Inspecting each phase
- Categorizing data based on CSFs - Analysing over-arching CSFs
- Categorizing data based on project success - Assessing project success
- Connecting analysis to initial process model, modifying the model based on the findings

Figure 13. The analysis process summarized.

4 Findings

The purpose of this thesis is to understand what is important in RPA project implementations. The literature was already fused as a RPA process model, but the actual validity of this model still remains questionable. To establish how well the literature matches real-life, the model was tested on a case project. Now, the findings of this case study must be analysed, in order to determine how the developed model performed and what modifications should be made to it to improve its accuracy. This section covers the course of the data analysis; the section is structured based on the logic and phases of the analysis, which were established in the previous section.

4.1 Project phases

The RPA process model was built around project life-cycle literature. The first step in the data analysis is to inspect each stage in the project life-cycle. Here, steps are briefly re-visited to see what the step comprised in the literature-based model, list what the step contained in the case project, and how the two differed.

4.1.1 Phase preceding the project

The stage preceding the project was supposed to deal mostly with the selection of the target process and setting up the project team. From the counter perspective, this step can be seen also as the step where the initial idea for using the robot for the process is come up with. Either the target process is selected for RPA implementation, or RPA is selected as the tool to improve the target process.

In this phase in the test case, the idea of using RPA in the target process was conceived, and it was selected to be the test case in this study. Furthermore, in this step the project team was selected. The target process defines which departments in business need to be involved, effectively narrowing down the options of who should be in the project team. Of course, it is viable not to have all the end-users in the team, the selection within each role in the team is done on this step. In this test case, there was a single end-user who had carried out the target process for the past few years, it was natural to select someone with the highest amount of experience in the process. In addition to this one individual, most other members of the team responsible for the system otherwise were able to perform the process fully.

> *"Mary was not available at the time, so another end-user Tim, knowing how the process works, was brought along to help John fill the documentation." M4*

"While it was not initially going to go this way, it is good to have Mary validate the output from the robot, as she knows the best what the output should look like." M7

Initially, the idea was to set in stone who is the representative of each role in the project team. Instead, during the project it became increasingly clear that roles in this project are more like labels for a subset of people. Any end-user can join the project at any time and still perform the tasks required of them. When the original end-user was unable to participate in the solution definition with the given schedule, another end-user was able to fulfil the role. This realization was key for the success of the project; without it, the project would have been delayed by a few process cycles.

On top of the individuals fulfilling the roles changing, the roles themselves changed quite a bit during the project. For example, the end-users ended up taking more responsibility over the solution definition and the process documentation, and testing, than initially intended. This change was for the positive, as those with best knowledge in each section were able to give their insights and advance the project.

4.1.2 Conceptualization phase

Earlier in this study, this stage was described to contain the very beginning of the project, where specifications and goals for the project would be laid down. In the case project, the contents of this step were more or less the same. A project kick-off meeting and its aftermath contained most of the functions in this step.

In the kick-off meeting, the main goal and aim was to familiarize the RPA designer with the target process. This succeeded well in the step, and was also crucial to have it happen this early on in the process.

One crucial element missing was briefly introducing the business side to RPA. It was left out in the proposed model, but should be included here. For example establishing that RPA is indeed a viable tool for the process is a key step in this stage, and deciding on this requires brief insight on how RPA operates and what it requires. The RPA developer has all the necessary knowledge, it is only a matter of conveying it to the rest of the project team.

Another important step in the process should be a checkpoint that RPA is actually both a viable method of improving the efficiency of the process, and that it is the best option to do so. In the meeting, it was agreed that the process did not have any other possible alternatives for improving efficiency. The possibility of automations had been discussed with the expense system provider but all changes would have been too expensive to consider. This meant that RPA, as a cheaper and lighter method, was the way to go. Including such checkpoint to check the compatibility of RPA and the target process in terms of business logic in absolutely necessary, and it should be included in the conceptualization. If the project were to proceed without the check, it is possible that the entire project would be done for nothing, as soon after a better alternative would override all the changes.

One major point covered in this step was setting up channels for communication. To setup communication, a channel was setup in the intranet of the company. The intranet uses a Facebook-like system, where channels act as groups for people to chat about different topics. For this case project, a channel was setup with all the project team members included. This channel proved beneficial, as sharing information about progress and asking questions about key problems would have been more difficult using the conventional tools, e.g. email.

The kick-off meeting should have covered setting clear goals and steps for the project. In the case, this did not explicitly happen. The memos mention that the project would progress on a first-priority basis, but there are no concrete statements about a set schedule or milestones. The process itself is run once a month, which created problems for scheduling, as the design phase and its validation would be sequenced in monthly intervals. Memos afterwards mention that a brief plan, perhaps highlighting the big milestones along the way would have been beneficial later in the project. For this case study, the schedule was agreed to only be "as soon as possible". The RPA projects are not that long, and therefore a detailed long-term plan is not needed. However, not having a schedule at all created unnecessary need for extra communication, as different parties were not aware of how the project was progressing, at times.

Another problem noticed during the project was that the roles in the project were defined, but what was needed of each role and when was not clear enough. For example, the process course outline and documentation and the responsibility of creating such document was not clearly defined. Therefore, in the project there were times where the RPA developer would have to instruct other members of the project to do certain things that were required of each role. This problem would again be solved by having a project outline with milestones, a timeline that could include different roles and their responsibilities over time. This way, different roles would know what would be required of them and when. In the case project, the work progressed based on what the developer required from other roles, but having different parties work autonomously is the only way forward in the long-run.

4.1.3 Planning phase

Originally, outlining the schedule of the project would have taken place in this step. However, in the case project a more suitable place for this to happen would have been in the previous phase, as the nature of the planning phase changed from initial “planning the project” to “planning the solution”. Schedule needs to come before the project, not during the project. Furthermore, having all of the kick-off based things in one meeting, and that meeting taking place in the conceptualization stage, seems to have worked perfectly as far as structuring the beginning of the project goes.

In the test case, the schedule was not defined at all. This was because of two main reasons: firstly, the target process is executed only once a month, which means sketching a rollout plan in more detail than monthly would be useless. Secondly, the project was considered small enough in terms of resources needed, that there would be enough resources available to the project regardless of the schedule. Here in a project like this, such a schedule was sufficient, but generalizing this result to processes that are more frequent and require more resources is not possible.

In the proposed model, this step would include solution definition. In the case project, this definition was expanded to a full process documentation, a detailed explanation of the course of the target process made by its end-users. This way, the RPA developers were able to perfectly replicate the business logic behind the process without much effort. The document was shared in the intranet channel, allowing modifications and improvements to be made afterwards.

Documentation of the target process did take place in this step, but given its depth and importance to the entire process, it should be its own step in the process. Its significance should be enough to warrant an entire step. Now, while it was initially depicted to take place in planning, it was intertwined between planning and design. As it really is a prerequisite for the design phase (to a degree), it should be its own sub-stage in “Execution”. In the case study, the documentation was created out of the need to convey the ideas between the different parties, but this phase should be standardized and a template for the documentation should be established.

> *“I want to underline that having an example of the documentation shown in the kick-off is essential; it will help end-users reach the right mind-set quicker. We did have this in the kick-off, but since Tim was not present, he was not up-to-speed yet.” M4*

Furthermore, a template for the documentation should be available before the project and this template should be shown to the end-users at the kick-off meeting. This way, the end-users would know what the depth of the documentation needs to be and what is really required for the RPA developer to be able to model the process for the robot. Every click needs to be documented, which is something the end-users did not right away realize in the test case.

Having dedicated resources is a CSF that was linked to this stage of the project. Part of the planning would be to cement resource allocations to the project, and to have some resources dedicated to this project only. In the case project, the resource allocations were agreed already in the conceptualization and the kick-off meeting.

4.1.4 Execution phase

Initially, the execution phase was to consist of three separate sub-stages: design, testing and rollout. Design would cover the development of the RPA solution, testing would comprise testing the said solution with the end-users to ensure it was working correctly, and finally rollout would be where the solution is taken into use. The model would first have a process documentation prepared and validated, then the RPA developer would design the solution to match the course of the process in this documentation. After, the end-users would test the solution to validate the outputs it produces and either accept it or return it back to development.

However, instead of having distinctly separate phases for planning (mostly the process documentation part) and execution in the test case, both of these phases were used intermittently and iteratively throughout the project. The first version of the documentation for the process was initially prepared before design by a different set of users that who were currently performing the process. This was made possible because the process is rather simple and all members of the end-user team were able to carry out the process – meaning there were enough resources for preparing the documentation available. What this meant, however, was that the validity of the documentation would remain unknown until the current end-user – with the best and latest knowledge of the process – would be able to validate it concurrently with importing the next batch of travel data. Later on, this first draft of the documentation proved to be rather accurate and complete from the get-go, which in turn enabled the design phase to start earlier, before validating the process documentation in actual use. Afterwards, the document was modified multiple times during the project, as missing components were identified during the design phase.

“A key realization during the design was that the process would be essentially split in two […]. Essentially, the first half of the process takes the data and re-assembles it to another datasheet. This sheet can be verified by Mary when she enters the data to the system.” M6

In the design, the target process was split into two functional blocks for development purposes. The first step would comprise the handling of data files, extracting and interpreting the travel data to match what the travel expense system requires. The second part would then take this pre-formatted data and input it to the system. This re-formatting of the process is visualized in Figure 14.

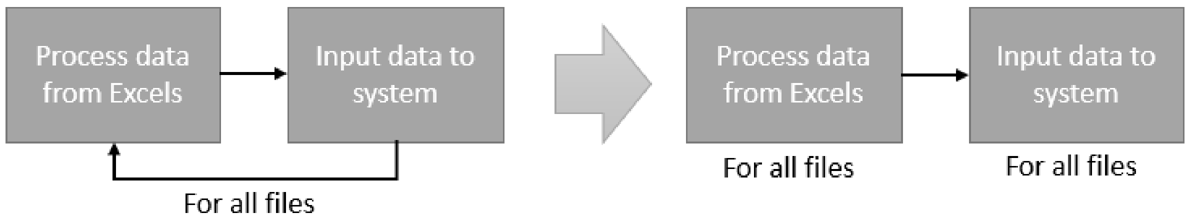


Figure 14. Overview of the target process logic for human (left) and for robot (right)

The validation for the process documentation took place only after the design for the first functional half of the process was already complete. An early start to design allowed it to reach a half-way point before the next set of data was sent by the customer. This way, the RPA solution and manual perform of the process were used in parallel, as well as checking the documentation’s validity. The data handling part of the RPA solution proved to be accurate, as it provided the same set of output data that the end-user did. A few additions and corrections were made to the documentations and then it was considered to fully match the course of the process.

Splitting the design of the solution into functional blocks enabled concurrent design and testing. Both tasks were carried out gradually, instead of designing the entire process at once and then testing it at once. This meant that errors were spotted earlier and only small corrections were needed to resolve them, instead of having to possibly re-do the entire design from that point onwards. Perhaps all the target processes in the future should be divided in functional blocks and structure the iterative design and testing around each block.

Starting the design before validating the process documentation might have resulted in redundant work done (in case of inaccurate documentation), but the core structure of the process and the accuracy of the initial version of the document meant that there was only little risk for this to happen.

The benefit for this was that the design phase proceeded early. Without this, the data for next month would have arrived too early for the testing, basically moving the rollout one month forward.

While design and testing overlap in the test case, so does the rollout. Using the first half of the process already in testing with actual data can be considered as having the RPA solution in use. The test case in this case study was used in parallel with the manual target process, so the robot was already doing some work but not replacing any manual work yet.

In terms of CSFs in this step, the communication was flawless. Whenever errors or undocumented courses in the process were discovered, they were communicated effectively to the end-users and resolutions to these cases were also received quickly.

Validation of the documentation and aligning design with testing and rollout allowed the end-users to have their say in how the robot should be performing. This way, at the end of the project the end-users have their needs met, resulting in higher user acceptance. In this case this worked fantastically, as no complaints were made after the project was concluded. Everyone were simply happy that the robot had taken over this process.

End-user skillset was a CSF in this step. Initially this meant educating the end-users to be able to properly use the RPA solution and reap the benefits it produces.

> *"I decided to compose a small instructional document detailing what is required of the end-users once the robot takes over the process. This helps the users accept the solution and also helps in terms of change management." M11*

In the test case the skills required for the end-users were deemed not so critical, that formal training was not needed. Rather, the RPA developers wrote a document outlining different error cases and possible deviations that the end-users would have to react to. With this documentation, the end-users would be able to perform needed tasks properly. In more complex processes, however, the documentation might not be enough simply because of the diversity of actions that the end-users must be able to perform.

4.1.5 Termination phase

The termination phase of the project, in the model, included topics for transferring knowledge and documentation of the process over to the end-users, as well as deciding on where the responsibilities of the process would lay after the project. In the case project, this step was covered in a project conclusion meeting with all of the project team members present.

For the purpose of knowledge transfer, the RPA developers had produced a short instruction document detailing the process, its steps, possible common error cases and how to solve them. The purpose of this documentation was to enable efficient knowledge transfer. Naturally, this serves also as the documentation for the developed RPA solution.

One point on the agenda of the meeting was to solidify user acceptance. In the meeting, the RPA solution was demonstrated by visually showing what it looks like when the robot is working on the process. This way, all the project members, especially the end-users, would be able to see the final version, and either accept it or not.

An additional function for this phase in the project was to serve as a final check that everything in the solution was done right. There were two changes that were identified in the meeting and solved afterwards. This could count as testing, further emphasizing how intertwined each step really is with the others, the process really is not linear.

> *"I would also demo the process in this meeting to show them how the robot works. This aims to sell them the idea of how "cool" robots are and give further momentum in the use of robots." M10*

Another addition to this step was deciding on how to proceed with support in the future. In the original mode, support and troubleshooting are covered after the project had concluded, but here a need for setting up practices for these beforehand was identified. To that end, the end-users were informed of how they would be able to contact the RPA developers in error cases. In the solution documentation common error cases and their instructions are already listed, which helps lighten the load on RPA developers in those cases. Furthermore, the RPA developers agreed to keep an eye on the first execution of the process by the robot, only transferring full responsibility of monitoring to the end-users after that.

4.1.6 Phase following the project

In the RPA process model this step only includes the support and troubleshooting. In the handover meeting, basic guidelines and practices for support after the project, were agreed on

In addition, the future of the RPA in terms of this business unit was discussed. The business side was impressed with how the robot behaved and they already in the meeting began to discuss further ideas for target processes. To exploit this enthusiasm in advancing RPA within the organization, the implementation process should have in the termination phase a notion about outlining ways to collect these ideas in the future. In the meeting, the RPA developers encouraged the business to list their ideas and that the best ideas would be implemented as soon as possible.

4.2 Over-arching CSFs

In this section, all the over-arching CSFs are analysed based on findings from the case project. Again, the literature-based process model and the roles of these CSFs in it are used as a base for comparison to how these factors impacted the case project.

4.2.1 Communication

> *"I feel like I have said that many times over, but it is the truth, communication is what enabled this project and what made possible the execution of it in a timeframe as tight as this." M13*

In the RPA process model, communication was regarded as the backbone of the project. Without proper communication a project quickly falls apart as it fractures into small individual bits unable to act together. In this test case communication was sufficient and successful all the way from start to finish. Furthermore, communication seemed to be effortless and different parties were able to contact each other on demand.

> *"Tim and Mary were both on sick leave. This left part of the open communication between us three out of the picture. Therefore, having good established channels in our intranet was essential; any progress could be documented there for all to see after returning to work." M6*

For communication purposes, the company's intranet was used, and in there a channel was founded to host all discussion about the project and its status. The channel also acted as a platform for file sharing, e.g. the process documentation was uploaded there. This way not only were the files shared, but different people could work on the file and improve it incrementally, at different times. This form of collaboration can also be regarded as communication, a close-knit form of that. Without such established practices for communication at the start, the project would not have been as successful. This channel also serves a purpose in the future, as further improvement ideas and target process suggestions can be openly discussed between business and RPA developers.

"The fact that I am directly receiving this feedback [about an exception in the process] means that communication in the project is working as intended. Open communication allows people to feel comfortable to give any feedback; in the case of RPA, even a small detail can be crucial in the RPA design." M7

One example demonstrating the strength of communication can be seen in the way needs for change were conveyed. Whenever there was an error, the resolution to that problem was easy to form. The RPA developer was able to effortlessly contact those in the project who were able to help. For example, when there was an issue connecting customer employee names in their respective files to customer employee names in the expense system, the issue was immediately laid out and communicated to end-users. They were then able to contact the customer and sort out a naming convention for the employees that would overcome the issue. All this happened within a few days, which highlights the efficiency of the channels available.

On top of the errors, also suggestion for changes were communicated and reacted on quickly. There were cases where change requests were not necessarily errors and in a project where communication would be weaker, could have been completely forgotten and missed or not even conveyed at all due to a mental barrier of difficult communication. In the test case for example, a last minute change in the scheduling of the process from midnight to half past one at night was conveyed only at the final meeting of the project, but it was changed quickly.

"I spoke with Mary, re-iterating what I had already commented to intranet about next steps […]. I realized that maybe there had not been clear enough steps for the project, and it was all only in my head. Having a small roadmap for steps (documentation, design, testing, pilot, etc.) would have been good. Now, the next step was always something unknown. While communication through intranet is a viable strategy, it only details each step at a time, rather than painting the big picture." M6

While the communication was good, it was not flawless. There were times in the project when not all members were fully on board and there was confusion about who was supposed to do what next. To counter this, a small roadmap detailing the key points of the project and where the responsibility for each step would lie. In the test case the end-users might not have always known what the big picture was and what was going to happen next.

The final verdict on communication based on the results of the test case is that it is a crucial part and indeed a backbone of the project.

4.2.2 Change management

In the initial RPA process, change management was considered to play a critical role. Mainly, this was because of anticipated resistance of change, something that is common when trying to change well-established traditions and ways of working.

> *"Everybody were happy to get rid of this process that had been an extra burden for the entire team." M12*

However, the nature of the test case was distinctively different in that all end-users considered the target process as extra burden. This, in turn, meant that no one was opposing the introduction of RPA solution to replace human work, which is not what was expected before the project began. Due to this surprise result, it is questionable to generalize the results of the test case when it comes to change management. There are bound to be cases where end-users will try to resist the change, therefore change management should not be overlooked just because in the test case the project proceeded smoothly. However, a key lesson that can be taken away from the case is that it is possible that, by targeting the right areas of business processes, to select such processes that the end-users are happy and willing to give away for the robots to work on.

4.2.3 Project champion

As was the case for change management, the findings for project champion as a CSF are somewhat inconclusive, as well. In the project, the role of the RPA developer as the champion was more apparent at the beginning of the project rather than at the end. At first, business did not know anything about RPA, and it is here that the knowledge and message from the developer was crucial. However, as the project progressed onwards, and partially due to the end-user wanting to get rid of the process, there was little need for convincing anyone of the benefits of the RPA. The resistance to change was non-existent and therefore the need to actively convincing end-users to accept RPA as a tool that is beneficial for them, was not present during the project either. However, it is possible that prior to the project, the end-users had already realized RPA was being introduced with their best interests in mind.

> *"Demonstrating how the robot handled the process went well. All participants were impressed with how fast the robot was, and how funny it is when the robot just moves the cursor around." M12*

It is crucial that at all times RPA as a new tool is being discussed in a positive manner; the word of mouth can have a large effect on how an organization adapts to new technologies. Demonstrating the process in action at the handover meeting seemed to have really helped in this case. This part of the meeting can indeed be considered championing RPA.

4.2.4 User involvement

> *"From the get-go, the end-users were part of the project team and were able to have their say in everything about the RPA solution." M13*

User involvement was, too, considered a key factor in the RPA process model. To this end, the entire process model was built around users and their input. Therefore, users were involved throughout the project. Rather than thinking of this CSF as something outside the process model that you would have to constantly try to enforce, it should be seen as already integrated in it. For example, having end-users do the documentation of the process meant that their input for the design phase was evident. Furthermore, having the end-user perform the process and validate the output from robot in parallel made sure the data processing was properly made.

4.3 Project success

The final analytical part of the data analysis is looking at how the test case succeeded as a project. This investigation is necessary because determining the outcome of the project effects on how the initial model should be approached when modifying it. If the project was a sound success, there is little that needs to be changed in the model; however, if the case project failed to deliver, then more drastic measures are needed and the model should be more fundamentally redesigned.

In the literature review, project success was divided in two dimensions, project management success and product success. Both of these factors containing further two sections of success measures each: product success has goal and purpose, while project management success has outputs and inputs. The tone of the memos suggests that the case project would be a success, however further analysis using these concepts should be made.

> *"While the project originally did not have other schedule than "as soon as possible", I feel all the work was done in a rather short time-frame and everyone was happy about that. The project took only a bit more than a month from start to finish. The fact that the data from customer comes only once a month meant that the project would never be able to finish in a shorter*

timeframe than one month, the project taking only five weeks means we were close to the minimum time." M13

Starting from lower down the hierarchy from project management success. In terms of inputs to the project, the memos have no mention of anything being restricted because of lack of resources. This would suggest that sufficient resources were available throughout the project. Furthermore, it seems that these resources did not go overboard; the use of available resources was rather lean. Schedule, then, while it was not explicitly outlined before the execution of the project, was a success too. Drawing together from these, the project was a success in terms of how it used its inputs.

Outputs, then, are very easy to analyse. The project aimed to deliver a RPA solution to the target process, this solution is the deliverable of the project and the success in this category can be assessed by seeing whether this output was produced or not. In this case, the solution was built. Furthermore, the solution was accepted by the end-users, meaning it meets pre-determined requirements: it can do what was originally required of it. Deriving from this, it seems that the project was success also in the output category.

Project managements success consists of both input and output, and as both of these aspects can be checked as success. Thus, in terms of project management success, the project was a success.

"Throughout the project the end-users were involved through close communication, which meant that they were able to have their say in shaping the RPA solution to what they actually need. This way, user satisfaction was reached, which further contributed to the project being a success." M13

Next in the hierarchy is purpose. The purpose of the project can be seen as actually satisfying the needs of the target audience, in this case the business. It is not sufficient to just provide the output, the deliverable must also be useful and well-functioning to have a proper impact. Based on the enthusiastic response from the end-users would suggest that they were happy with the RPA solution, implying the purpose was fulfilled.

Finally, there is the goal of the project. This is a more abstract target, something that is often not only part of the current project. In this case, the goal could be increasing the efficiency of the organization through introduction of RPA, or improving workdays of employees by replacing tedious and repetitive tasks with RPA solutions. Either way, the target is to use RPA and have a positive impact on the organization as a whole. In this test case, the project was well-received. The target process was such that everyone wanted to let go of it, replacing the human work with robotics made the end-users happier. The end-user who had been performing the process, on loan from another team, was especially delighted that this process was now automated. With all these in mind, the project seems to have been a success in this section, too, and therefore success in all four domains (Table 10).

Table 10. Success measures of the case project.

Success measure	**Meaning in case**	**Success?**
Goal	Improve efficiency	Yes
Purpose	Users accept RPA solution	Yes
Output	RPA solution	Yes
Input	Work resources	Yes

4.4 Revised model

This section aims to synthesize the findings from the data analysis by re-visiting the proposed RPA process introduced in section 2.3. This way, the model can be improved on where it lacked the most, establishing a solid basis for future RPA implementations. As the project was successful, it could be reasonable to assume that, already in its current form, the model developed in this study could work as a good template for further projects. However, there are some revisions and additions noted based on the findings that should be made to the model.

In the beginning of the project, there are few things that should be changed in order to have an effective start for the project. In the kick-off meeting of the case project, the RPA developer and the end-users briefly exchanged some knowledge about the context of the project: The RPA developer outlined what RPA is and what it can do, to allow the business to better understand what is going on. In exchange, the business representatives explained and demonstrated the target process, by showing input data and the expense system. This knowledge transfer at the beginning was not documented in the original process model, however, it proved to be useful down the line. It should be added to the agenda of the conceptualization phase in the project.

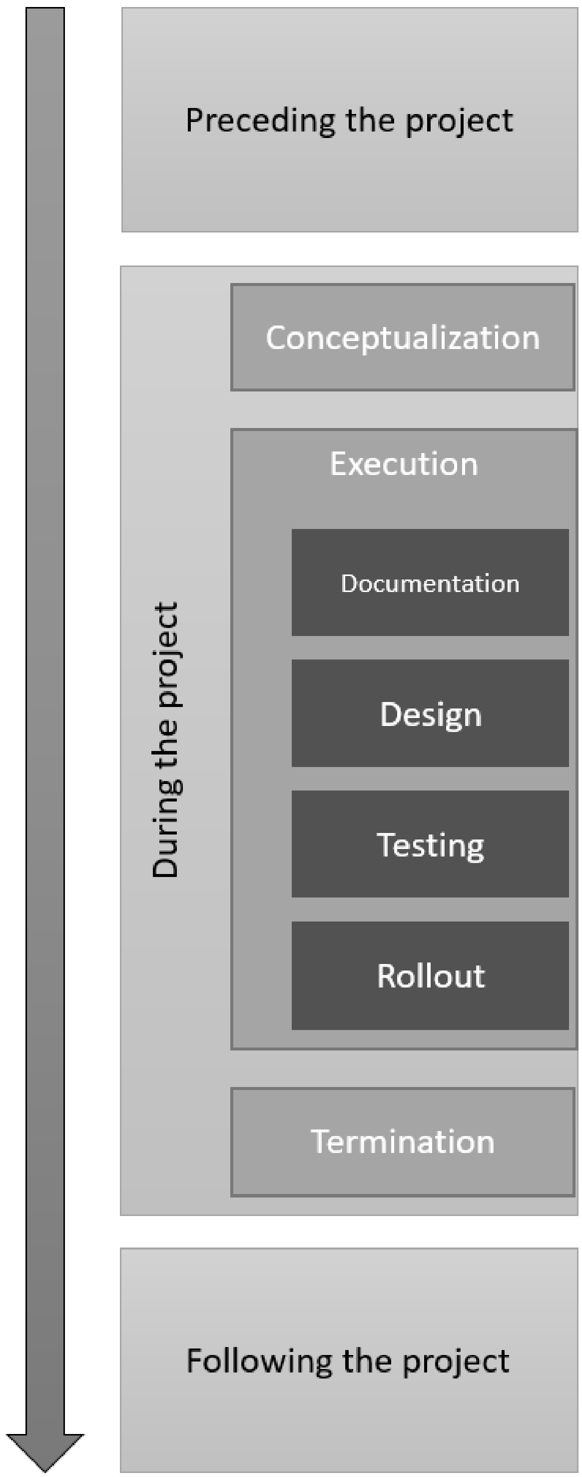
In the case project there was some confusion among the end-users of what was going on at the project at times, and who was responsible for the next steps in the process. To this end, a light framework for the schedule of the project with major milestones highlighted, a roadmap, should be introduced. This brings with it improvements to the management of the project. The lightness of this framework would enable an adjustable approach to project management, with possibility for changes later on in the project. RPA projects are light in nature, therefore heavy management should be avoided. The roadmap could mirror the proposed model skeleton outlined in Figure 10, with key people added to each step to highlight responsibilities. This would counter any misunderstandings and every step would have someone responsible for the progress.

By far the largest and most significant change revolves around the process documentation. In the proposed model, the documentation was not given its rightful share of attention. Due to its importance for further steps in the process and even the overall progress of the project, documentation needs to be given its own step in the process. Creating the process documentation can be regarded as the first step of execution, when the action in the project properly starts. It is also a prerequisite for designing the RPA solution. Therefore, the documentation step in the model should be included inside execution as the first step.

Having documentation as its own step means there is not much content left in the planning phase. In the test case, planning and conceptualization were actually completed together, the contents of both steps were handled in the project kick-off meeting. To this end, it would be natural to merge these two phases into one, having only a single phase for the start phase of the project.

A key realization, not necessarily a change in the model, was that the steps in the execution phase of the project are not sequential. Rather, they happen in an iterative manner, taking place by demand based on the progress and need of the development of the RPA solution. For example in the test case, the design and testing took place bit by bit, overlapping each other. As the design went on, different sections of the RPA solution were validated right away instead of testing all of the solution in one big chunk entirely after the design had finished. This led to instant fixes in cases of errors; if the errors were only found after the design had entirely completed, the magnitude of redesign could have been multiplied.

Other steps in the RPA process model functioned well and have no further needs for improvements. The improved process in its entirety is depicted in Figure 15.



Selecting target process, preliminary roster of the project team
Specifications, goals, objectives, roadmap introduction to RPA, introduction to process
Producing the RPA solution. Sub-stages happen iteratively
Composing a detailed documentation of the process
Designing the RPA solution based on the documentation
Testing the RPA solution, validating the outputs
Gradual implementation of the RPA solution
Transfer knowledge, instructions, responsibility
Continuous support of use, continuous ideation for improvements and new ideas

Figure 15. Revised model of the RPA process

5 Conclusions

Advances in technology have throughout the history enabled organizations to improve their efficiency. With robotic process automation, organizations can finally begin to properly improve the productivity of knowledge-based work, too. With this prospect now at hand, companies need to figure a way to effectively and efficiently implement RPA to their processes. In knowledge-based work, automation solutions need to be tailored and customized to adapt to the needs for each target process. The RPA literature and project literature offer the building blocks to construct a model for implementing RPA solution to a target process. There are multiple factors that need to be taken into account to make a RPA project successful.

5.1 Combining project life-cycle and project success with RPA

This study began with introducing three research questions: What elements would be crucial in different IT projects (RQ1), what stages have been identified to form a project life-cycle (RQ2) and how a process formed of these elements would perform in a real-life setting (RQ3). Based on these questions, first literature was reviewed to understand what the fundamentals for establishing answers to questions RQ1 and RQ2. Then, using these perspectives and the context of RPA, synthesize the findings to form a RPA implementation process. This process is visualized in Figure 16, which shows the two main facets or areas of the literature that were synthesized together using the context of RPA as a filter and a mould. Next, this model was tested in a case study, by developing a RPA solution for one selected target process. Findings from the case were then used to improve on the model. The findings help to further understand the forces in this context, providing an answer to RQ3.

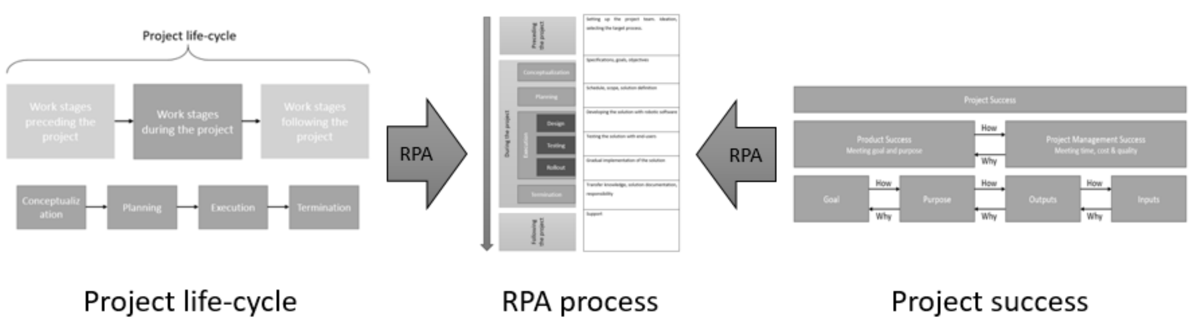


Figure 16. Summary of the progress and logic of this study.

From a research point of view, the main intrigue of the focal phenomenon was to discover how RPA is different from other sorts of IT projects when it comes to their implementation. Finding the answer to this question was at the core of the study. Based on the RPA literature, there were multiple differences found, but two major distinctions stand out. Firstly, RPA as a technology is lighter than most traditional automation tools. This aspect shows in both resources needed and the agility of the approach itself; RPA has a very high return on investment, and the benefits are realized in a very short time frame. The second significant difference was the needed skillset of the developers and users operating RPA. Much like other electronic tools such as spreadsheet or word editors, the logic in using the tools is self-evident enough to not require vast amounts of experience before users can begin to harness the benefits. These differences make it clear that RPA solutions must be introduced with an implementation process designed just for them.

In terms of project management, RPA projects were found to be partially in line with other IT projects, which translated to project management literature being, to a degree, applicable to RPA. Synthesizing the literature, the project life-cycle consists of six major steps that cover the entire timeline of a project. These steps provided a solid foundation for building the model for the implementation process for RPA. Critical Success Factors are factors that, if fulfilled, will make the success of a project more likely. There are plenty of such factors, but some of them are universal and more common in literature. These factors were used as building blocks to further populate and shape the implementation process for RPA.

Findings from testing the process model in action with a case study validate the assumptions made earlier: RPA projects are indeed similar enough with other IT projects that transferring concepts from literature to RPA project model was a valid approach. The project life-cycle was suitable and almost all of the steps took place just as planned in the model. The success factors identified were relevant and proper attention was focused on each of them. Furthermore, using project success measures, the case project was a success, which suggests the project model performed adequately and catered the needs dictated by the nature of RPA projects.

The main logic in this study was to first synthesize literature to form a RPA implementation model, and then modify this model based on the findings from the case study. The findings, though, show that the initial model performed more than adequately and that there only needed to be slight modifications – such alterations that do not fundamentally change the logic behind the process. These changes also are RPA specific, converting the already RPA-tailored process model to an even more RPA customized state. For example, increasing the significance of the process documentation and elevating it to a proper step in the implementation process was something only RPA projects would need. Another good example would be the proper introduction of RPA to the end-users at the start of the project, an approach that is only needed due to the novelty of RPA, meaning most end-users have never heard or seen RPA before.

The underlying lesson from the findings is clear: To have an impactful RPA implementation, one needs to consider both project CSFs and project life-cycle in the model, as well as the context of RPA. The literature offered the initial template that worked. The findings from the case project allow for further modifications to tailor the process model to the nature of RPA.

5.2 Theoretical implications

RPA literature in general, due to the technology's novelty, is not very comprehensive or mature as of yet. The existing literature is focused mostly on presenting what RPA is (e.g. Lacity et al. 2015, Lacity & Willcocks, 2015, Willcocks et al. 2015A), instead of diving further down to nuanced connections. Therefore, there are not too many theories to ground the findings from this thesis to. Instead, this study contributes to the existing literature by binding different established concepts together to form new knowledge. By doing this, the study produces new insights by applying RPA as a context to concepts from already established disciplines, mainly the project management. This produces multiple new connections and linkages between different areas of study.

First of all, using the context of RPA as a glue, literature about project life-cycles and project success were fused together to create an implementation process specifically tailored for RPA projects. This produced insights on how well previous studies on project life-cycles could be adapted to RPA, and highlighted both the similarities and differences of the projects for RPA implementation and traditional projects. The field of project management is already well saturated, for example the literature surrounding project life-cycle spans multiple decades and has been the focus of many studies (e.g. Pinto & Slevin, 1988; Parr & Shanks, 2000; Hartman & Ashrafi, 2002, Artto et al., 2006). This means its concepts and theories have already been well-tested and validated; the only new component introduced in this study is the RPA as the context, to provide a new viewpoint.

Secondly, in creating the RPA process model, project CSFs were closely and meticulously examined and adapted to specifically fit RPA projects. This established which CSFs are actually important in RPA projects and what can be left out. Again, as Wateridge (1995) suggests, the theory about CSFs is established and tested by previous studies, but for each context and setting the CSFs need to be defined and agreed on, separately. This study contributes to this discussion by using RPA as a viewpoint, forming a list of CSFs that are critical in RPA projects. This RPA-tailored list of CSFs can be used as a basis for further projects or research, as RPA project success factors.

The test case project in this study helps in establishing the validity of the aforementioned findings. The findings show that RPA is indeed different from traditional IT projects and that it needs a tailor-made solution for its implementations.

5.3 Managerial implications

One of the motivations behind this study was to help organizations begin to harness the full potential of RPA in their business processes. The current RPA literature does not explicitly offer a concrete enough approach on how RPA projects should be structured. It was established that a new approach, an implementation process tailor-made for RPA would be required to achieve the full potential of the promises RPA offers. From a managerial viewpoint, the main contribution of this study is establishing a template for this RPA process model, enabling proper RPA implementation projects. Looking back at the research design, these implications also act as design propositions from the CIMO logic, being essentially guidelines on RPA project design.

Indicated by the success of the case project, the RPA implementation process model proved to be properly made, successful and suitable for RPA. If managers want to replicate this success, they should use this model and follow its steps in the life-cycle, which should provide a clear path towards project success. However, since this process was only tested in one case and one context, managers should adapt the process to each project individually, taking care the differences in context for each case are taken into account.

To help make RPA projects successful, the study introduces multiple critical success factors that need to be fulfilled; by concentrating sufficient resources on each of these factors managers can ensure successful RPA implementation projects. In total, 12 of these aspects were fused into the RPA process model. It is imperative that all of such factors get their share of attention, however, there are some factors that can be regarded as the most important ones. Mostly the more fine-tuned prioritizing of the factors is case specific, and project managers must pay attention in selecting the right combination for each scenario. In general, one of the commonly important factors is communication, which is crucial to any project. On top of being a critical factor itself, it works as an enabler to other factors, too. If communication works, often many other problems are immediately mitigated.

As a new technology RPA is still mostly void of best practices. With its results, this study aims to provide first steps in laying the foundation for conventions that help managers focus their efforts on the right areas, until more common, industry-wide best practices can be formed. Therefore, caution should be used when applying the results of this study in the future, as new discoveries related to RPA can easily revolutionize the entire field of robotic automation.

RPA projects can be treated as isolated and still be successful, but by following the established set of CSFs, there is more to gain than just success in singular projects. By introducing RPA as a continuous way of improvement in the organization, managers can encourage business to become the drivers of RPA themselves. By having good experiences and letting the end-users themselves first-hand benefit from the upsides of RPA, a positive feedback loop can be constructed, where the business themselves wants to take RPA into use in more processes. Not only does this drive allow future projects to be more easily successes, but it also lowers the resistance to change.

For the case company (where the test case took place), a few improvement suggestions can be made based on the findings of this study. There are positive aspects in their current RPA approach: the implementation process is flexible and easy to tailor for each target process. The flipside of the coin here, unfortunately, is that projects are very loosely defined; the process lacks focus, which manifests in absence of tangible targets for the projects. This study can provide some help in tackling these problems: the introduced RPA process offers a good template for the project life-cycle, which can help with the structure of the projects. Like in the test case, without a good roadmap, project members can feel lost and not know what is expected from them and, therefore, be not able to fully contribute to the project. Furthermore, the critical success factors establish good targets to focus resources on, further improving focus in the projects. All in all, the company can learn a lot from this study, but they should not completely forget their current approach. Instead, they should adapt a way of continuous improvement, picking best performing components from both the study and their current process, testing and re-designing everything with every RPA project.

5.4 Limitations and further research

No study is without limitations when it comes to the results, and neither is this one. All in all, the research design for this study has been well-constructed, but as with all case studies, they share the aspect that there is only limited generalizability. All the findings and conclusions apply only in the context of this study, and any generalization outside the established setting should be done with caution. If a more comprehensive understanding of RPA projects in different contexts is needed, then further research must be conducted.

Another point to keep in mind when interpreting the results from this study is that in its core, the research is centred on forming, analysing and criticizing a process model. It is possible, that this approach, because of its focused nature, narrows the scope too much, leaving some important aspects completely outside the selection. It could be, that some aspects are therefore completely ignored in the study. The impact of this limitation on projects can be mitigated by being open and attuned to suggestions for improvements in the process flow, essentially treating this established process as a solid template to further build on.

RPA as a technology is still very new and the range of diversity of literature available is not too wide. It is likely that because of this, the contents of the RPA literature review are not as comprehensive as they could be. As the technology evolves, further studies should be conducted, to keep up-to-date with the state of RPA and how this context could change in the future. However, due to the test case being successful, the introduced process that was extracted from the literature is applicable and exhaustive enough to serve as a basis for the process model.

Another matter to consider, caused by the novelty of RPA, is the pace at which the technology is still advancing. Most of the sources for RPA literature are only a couple of years old, and with the completion of this thesis taking more than half a year in total, it is possible that during this timeframe there were some significant findings or advances in the field. To this end, it is worth noting that it is possible the findings of this thesis are already slightly outdated due to this fast pace. Again, further studies could help in understanding how and why RPA is different and how it has evolved.

The limitations of this study bring up prospects for future studies. Conducting studies with multiple cases across different business units and industries would allow for a more comprehensive analysis that could highlight differences and solidify which elements are crucial in all RPA projects and which aspects shift in terms of importance depending on the context. With the rapid advance of RPA technology, there is a rather unique possibility to study changes in RPA implementations across time. It is likely that the elements considered important in RPA project now will change, as with more maturity the technology will establish itself and the crucial elements to success will also change. Conducting studies in succession would enable this sort of analysis in a considerably shorter timeframe than with technologies that no longer evolve with such a rapid pace.

6 References

Applegate, L. M., McFarlan, F. W., & McKenney, J. L. (1996). Corporate information systems management: The issues facing senior executives. Irwin, Chicago.

Artto, K. A., Martinsuo, M., & Kujala, J. (2006). Projektiliiketoiminta (Vol. 2). WSOY.

Bals, L., Daum, A., & Tate, W. (2015). From offshoring to rightshoring: focus on the backshoring phenomenon. *AIB Insights*, 15(4), 3.

Beath, C. M. (1991). Supporting the information technology champion. *MIS quarterly*, 355-372.

Cooper, R. B., & Zmud, R. W. (1990). Information technology implementation research: a technological diffusion approach. *Management science*, 36(2), 123-139.

Corbin, J., & Strauss, A. (2008). Basics of qualitative research. 2008.

Davis, N. (2013). Driving quality improvement and reducing technical debt with the definition of done. In *Agile Conference (AGILE), August 2013* (pp. 164-168). IEEE.

Denyer, D., Tranfield, D., & Van Aken, J. E. (2008). Developing design propositions through research synthesis. *Organization studies*, 29(3), 393-413.

Grover, V., Jeong, S. R., Kettinger, W. J., & Teng, J. T. (1995). The implementation of business process reengineering. *Journal of Management Information Systems*, 12(1), 109-144.

Hartman, F., & Ashrafi, R. (2002). Project management in the information systems and information technologies. *Project Management Journal*, 33(3), 5-15.

Jarvenpaa, S. L., & Ives, B. (1991). Executive involvement and participation in the management of information technology. *MIS quarterly*, 205-227.

Lacity, M., Willcocks, L. P., & Craig, A. (2015). Robotic process automation at Telefonica O2.

Lacity, M. C., & Willcocks, L. P. (2015). What knowledge workers stand to gain from automation. *Harvard Business Review*, 19.

Lacity, M. C., & Willcocks, L. P. (2016). A new approach to automating services. *MIT Sloan Management Review, 58*(1), 41.

Leidecker, J. K., & Bruno, A. V. (1984). Identifying and using critical success factors. *Long range planning*, 17(1), 23-32.

Nelson, R. R., & Cheney, P. H. (1987). Training end users: an exploratory study. *MIS quarterly*, 547-559.

Parr, A. N., Shanks, G., & Darke, P. (1999). Identification of necessary factors for successful implementation of ERP systems. *New information technologies in organizational processes* (pp. 99-119). Springer, Boston, MA.

Parr, A., & Shanks, G. (2000). A model of ERP project implementation. *Journal of information Technology*, 15(4), 289-303.

Pinto, J. K., & Slevin, D. P. (1988). 20. Critical Success Factors in Effective Project implementation. *Project management handbook*, 479.

Slaby, J. R. (2012). Robotic Automation Emerges as a Threat to Traditional Low-Cost Outsourcing. HfS Research Ltd.

Somers, T. M., & Nelson, K. (2001). The impact of critical success factors across the stages of enterprise resource planning implementations. In *System Sciences, 2001, January. Proceedings of the 34th Annual Hawaii International Conference* on (pp. 10-pp). IEEE.

Wateridge, J. (1995). IT projects: a basis for success. *International journal of project management*, 13(3), 169-172.

Willcocks, L. P., Lacity, M., & Craig, A. (2015A). Robotic process automation at Xchanging.

Willcocks, L. P., Lacity, M., & Craig, A. (2015B). The IT function and robotic process automation.

Yin, R. K. (2003). Case study research: design and methods (3rd ed.). Thousand Oaks: Sage Publications.

Zairi, M. (1997). Business process management: a boundaryless approach to modern competitiveness. *Business Process Management Journal*, 3(1), 64-80.

Appendix 1. Process documentation

This is a translated and re-formatted version. Parts in italics are additions or modifications that were made to the first version of the documentation.

1. Log in to the travel expense system
2. Open the Excel file.
3. Get employee's name for a specified cell
4. Copy name to ”Find person…” field in the system
5. Press enter
6. Select “New Invoice”
7. Select “Foreign travel” as the type from a dropdown menu
8. ”Travel Destination”, list of airports e.g. “HEL-PRG-CPH-ARN”
9. ”Invoice Description“ same as ”Travel Destination”
10. *If the first line does not originate from ”HEL”, the first trip is incomplete and the beginning of the trip is searched from the last month's file*
11. ”Departure Time” is copied from Excel under ”Activities”, the first flight that has ”HEL” as the departure code in column E and something else than ”HEL” as the arrival code in G-column. B-column has the departure date (input format DDMMYYYY) and C-column has time (input format HHMM).
12. ”Return Time” is looked up in the Excel in the row where next time G-column has “HEL”, the date is in B-column and time is in I-column (if I-column is empty, use H-column), similar to the departure time.
13. ”Country of first arrival” is taken from the same row as departure time, from column G. The country corresponding to this airport code is then searched for (e.g. Google, Wikipedia), for example PRG=Prague=Czech Republic
14. ”Arrival to country” time is from the same row as arrival country, date from column B and time from column H. If time has “+1” at the end, e.g. 0054+1, add extra day to date.
15. Select “Continue”.
16. *If the first arrival to a country is more than 24 hours after the start time of the entire trip, the system will ask for a departure time from the origin country here.*
17. Using “Add” button select “Country Arrival”

18. “Arrival to country” time is looked up from B-column (date) and H-column (time), from the next row as the trip start time. If the G-column on the row has the same country as the previous country, then move to next row.
19. “Country of Arrival” information comes from the same row from column G, if it is “HEL” then the system already suggests “Finland” by default.
20. Select “Save”.
21. Add country arrivals until G-column is again “HEL”.
22. *If the last row in the file has G-column as other than “HEL”, the trip is incomplete. This trip is then skipped, it will be added in its entirety next month.*
23. Move to “5. Confirm” and select “Send to approval”.

Appendix 2. Instructions

Robot instructions

General

The folder for robot to work in is set as:

[Folder path to robot here]

The files are read in from subfolder called "Input". The folder "Archive" has all old files that have already been handled. "Working" folder is used for those files which are currently used by the robot. This, for example, has all the files for mapping different inputs.

The status of the process will be reported by mail, the mail address has been set as [support email].

Flow of the work

1. The user puts the arrived data files to the Input-folder

2. The robot has been set to automatically start working at 00:30 every night, so the files will be processed automatically.

3. The robot will send email based on the status of the process. Errors will be automatically reported. A successful run of the process will also be reported.

Notice that the files in "Archive" are needed if journeys next month begin with an incomplete trip. Then, the beginning of such trip is fetched from the last month's file. Contrary to previous tradition, the robot does not save a month's last trip at all if it is incomplete. Instead, such trips are handled in their entirety at the beginning of the next month. Please do not delete any files from the archive!

Processing one data file takes about 10 minutes. With, for example, 30 files, a full run of the process would take five hours.

Common errors

Airport code is missing. The robot can react to a missing code, but the users must add this code to the mapping file in the "Working" folder. A single error message can include multiple erroneous codes. The robot will re-run automatically next night, so no further actions are required from the users.

The match for an airport code (target country) does not match with any of those in the target system. The robot will check that all destinations exist in the system. If, for example, a destination has been misspelled, the robot will not be able to find it and will report this by email. Again, all erroneous destinations will be listed in a single mail, and the user must simply check the writing of each of them in the mapping file in the “Working” folder.

Invalid file. The robot will notify the users via mail of an erroneous client data. There are two cases here: an empty file, or problem with interpreting the journey data.

1. The users asked for the information of an empty file to be conveyed to them, so that they can make sure the file actually should be empty. In case of a problem, the user should put the correct file to the “Input” folder and delete the empty one. The robot will restart by itself next night, so no further actions are needed.

2. Invalid trip data: If a file begins with a non-finished journey, the robot will try to find the beginning of said trip from the last month’s data. In some cases, the beginning will not be found, so the robot will need help sorting such cases.

It is possible, that one row has been entered trouble and is causing trouble (the last trip last month was actually completed already, but the first row is on this month’s file as incomplete). In these cases, the solution is to simply delete the erroneous row and moving the file back to “Input” folder. Such issues must be evaluated case-by-case.

Other errors. In case of any other errors, the robot will again send email. They should be handled case-by-case, but more difficult cases can be forwarded to the robot support for the robot developers to handle.