CODEN:LUTEDX/(TEIE-5439)/1-67(2020)

Industrial Electrical Engineering and Automation

How Organizational Structures Affect the Implementation of Robotic Process Automation

A Practical Case Study

Joakim Härd
Max Svensson

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

LUND UNIVERSITY

How Organizational Structures Affect the Implementation of Robotic Process Automation

A Practical Case Study

Authors: Joakim Härd & Max Svensson

Thesis submitted to Lund University Faculty of Engineering for the Degree of MSc. in Mechanical Engineering

2020-06-03

Preface

This thesis was conducted during the spring semester 2020 at the Faculty of Engineering Lund University. It was written as a part of the two authors master's degree in Mechatronics, Mechanical Engineering. The thesis was conducted as an investigative practical case study in collaboration with a packaging company in Lund.

The two decentralized cases which were investigated in this study were developed and implemented by the authors of the thesis. With the exception of section 3.6 *Company structure,* section 3.7 *Background cases* and the development of the second case all parts of the thesis have been worked on equally by the two authors. The coding and development of the second case was done solely by Max Svensson while Joakim Härd simultaneously worked on the abovementioned sections.

We would like to thank the persons who helped make this thesis reality. First, we would like to thank our supervisor Gunnar Lindstedt, at the division of Industrial Electrical Engineering and Automation, who has been very helpful and amenable during the project. We would also like to thank our examiner Ulf Jeppsson, at the division of Industrial Electrical Engineering and Automation, who has given us valuable insights along the way of the thesis project.

We also want to direct a special thank you to our company supervisor Pontus Holm, Part Supply Chain and to Andreas Feldmann at European Service Administration Centre, who has spent a great deal of their time answering questions and coming with helpful insights during the process. In addition to this, they have also passionately been involved in the process and genuinely been interested in arriving at a successful completion of the project. Thank you for letting us evolve at a leading industrial company within the world of robotic process automation.

Joakim Härd
Author

Max Svensson
Author

Abstract

Title: How Organizational Structures Affect the Implementation of Robotic Process Automation

Authors: Joakim Härd och Max Svensson.

Problem description: The case company, a packaging company in Lund, Sweden, operates in an industry where high productivity is becoming increasingly crucial to remain the industry leader. One of the most rapidly growing automating technologies today is Robotic Process Automation (RPA). The case company has adopted this technology. As of now the case company uses a centralized approach in implementing RPA which means that the planning, development, and deployment is conducted from a specific department. The Parts Supply Chain (PSC) department at the case company has identified that this centralized approach leads to long lead times and a reduced development speed. To achieve a better outcome in future RPA projects, the company's PSC department wanted to investigate if there is a more efficient approach to implementing RPA. They wanted to investigate whether decentralizing their RPA-related Information Technology-structure (IT) would be beneficial.

Purpose and research questions: The thesis investigated the effect that different organizational structures have on the implementation of RPA. The purpose of the thesis was to deliver a proposal of whether the department PSC at the case company should decentralize their IT-related structure or not. Given this purpose the following research questions were formulated; 'How does the proposed decentralized structure measure against the existing centralized structure regarding RPA implementation at the PSC department?' and 'How do data from two implemented RPA pilot cases, using the proposed decentralized approach, compare to an existing RPA project which was implemented using the existing centralized approach?'

Methodology: To answer the research questions an investigative practical case study was conducted. In the study, quantitative data was gathered through automating two work processes from a decentralized approach proposed by the case company. Data was also collected from an already completed project, performed with a centralized approach at the company. The data from the decentralized cases was compared to the results of the centralized case quantitatively. Thereafter, the comparison was analyzed qualitatively with regards to the organizational structures to fulfil the purpose of the thesis.

Conclusion: From analyzing the results of the practical case study it was concluded that a decentralization of the IT-related structure would be beneficial for the PSC department. The decision lead times between process identification and development would decrease, as would the development time. As a result, the department would see a lower investment cost per project. Furthermore, from the qualitative analysis it was concluded that decentralization would lead to increased autonomy and project flexibility for the PSC department.

Keywords: Robotic Process Automation, Organizational structures, Process automation, Business processes, Project efficiency, Lead time, Case study

Table of Contents

Preface III

Abstract V

List of abbreviations IX

List of figures XI

List of tables XIII

1 Introduction 1

1.1 Background 1

1.2 Purpose and research questions 1

1.3 Significance of the thesis 2

1.4 Limitations and delimitations 2

1.5 Assumptions 3

2 Frame of reference 4

2.1 Introduction 4

2.2 Robotic process automation 5

2.2.1 Benefits and limitations 5

2.3 RPA life cycle 6

2.4 Managing organizations 7

2.5 Summary 9

3 Methodology 10

3.1 Research strategy 10

3.2 Frame of reference 10

3.3 Method for development 11

3.4 Data collection 12

3.5 Analysis of data 13

3.6 Company structure 15

3.6.1 As is centralized structure 15

3.6.2 Proposed decentralized structure 17

3.7 Background cases 18

3.7.1 Background pilot case 1 - 3PE case 18

3.7.2 Background pilot case 2 - LA lines 20

3.7.3 Background centralized case 21

4 Empirical results 29

4.1 Case 1 29

4.1.1 Automated solution 29
4.1.2 Quantitative data 30
4.2 Case 2 32
4.2.1 Automated solution 32
4.2.2 Quantitative data 34
4.3 Analysis 36
5 Discussion 45
6 Conclusions 47
7 Future work 49
8 Contribution to research 50

List of abbreviations

API	Application program interface
AmSAC	Americas services administration centre
BPA	Business process automation
BPM	Business process management
ESAC	European services administration centre
GIM	Global IT management
IPA	Intelligent process automation
IT	Information technology
MDM	Master data management
OFSP	Order fulfillment service product-team
PSC	Parts supply chain
RPA	Robotic process automation
SAP	Systems applications and products
VBA	Visual basic for applications

List of figures

Figure 2.1: Projected increase of the automation market (Stiehler & Gantori, 2020) 4
Figure 2.2: The six steps of the RPA life cycle 6
Figure 3.1: The four stages the development process was divided into 11
Figure 3.2: The steps between process identification and budget decision 15
Figure 3.3: Contact between developer and user 16
Figure 3.4: The decision timeline 18
Figure 3.5: The process simplified 19
Figure 3.6: The manual execution time 20
Figure 3.7: Illustrates the decision timeline for the admin case 21
Figure 3.8: Decision timeline for the central case 22
Figure 3.9: Process flow chart for handling orders 23
Figure 3.10: Development timeline for the central case 25
Figure 3.11: Simple payback period for the central case 28
Figure 4.1: Overview of the process sequence for case 1 30
Figure 4.2: Development timeline for case 1 30
Figure 4.3: Simple payback period for case 1 32
Figure 4.4: An overview of the process sequence for case 2 33
Figure 4.5: Development timeline for case 2 34
Figure 4.6: Simple payback period for case 2 36
Figure 4.7: Decision lead times for each case 37
Figure 4.8: Development lead times for each case 39

List of tables

Table 3.1: Total time spent per year and the respective total cost 18
Table 3.2: Total time spent per year and the respective total cost 20
Table 3.3: Lead times between each step 22
Table 3.4: Total time spent per year and the respective total cost 24
Table 3.5: Total effective time and cost spent on each task. 26
Table 3.6: Cost and time together 27
Table 3.7: Effective project time and total initial investment 27
Table 3.8: Cost before and after automation 27
Table 4.1: Time spent per task and the resulting total investment for case 1 31
Table 4.2: Total project time and investment for case 1 31
Table 4.3: Effort before and after automation as a result of success rate for case 1 31
Table 4.4: Cost before and after automation and the resulting cost savings for case 1 32
Table 4.5: Time spent per task and the resulting total investment for case 2 34
Table 4.6: Total project time and investment for case 2 35
Table 4.7: Effort before and after automation as a result of success rate for case 2 35
Table 4.8: Cost before and after automation and the resulting cost savings for case 2 35
Table 4.9: Size factor for each project 36
Table 4.10: Number of persons per task and the % of full time spent on it (central case) 40
Table 4.11: Time spent on each stage normalized – for all cases 42
Table 4.12: Effective project time and total investment normalized - for all cases 42
Table 4.13: Cost before and after automation as a result of success rate for each case, normalized 43
Table 4.14: Normalized payback period for each case, normalized 44

1 Introduction

1.1 Background

The company analyzed in this thesis is a world leading packaging & processing company based in Lund, Sweden. The company operates in an industry where high productivity is becoming increasingly crucial to remain the industry leader. This higher productivity is crucial both in production and in the back-office environment where the administration is handled. Achieving higher productivity is often directly related to the level of automation within the company, since automation enables the possibilities to achieve economies of scale. One of the most rapidly growing automation-technologies today is robotic process automation (RPA), which is a hypernym for tools that operate on a user interface in the same way a human would do (van der Aalst et al., 2018). RPA is used to automate monotone digital processes which are usually time consuming and mundane. The company has identified numerous processes within the organization where RPA is applicable.

As of now the company uses a centralized approach regarding RPA which means that the planning, development, and deployment is conducted from one department while the decision, whether it is a viable process to automate, is made by another. The Part supply Chain department (PSC) of the company has noticed that the centralized approach leads to long lead times and reduces the speed of deployment which in turn hinders potential productivity improvements. To achieve a better outcome in future RPA projects, the company's PSC department wants to investigate if there is a more efficient approach to working with RPA. They want to investigate whether decentralizing their RPA-related IT-structure would be beneficial.

1.2 Purpose and research questions

The purpose of this thesis was to deliver a proposal of whether to decentralize the RPA-related IT-structure in place at the PSC department at the company or not. To achieve the purpose of the thesis the following research questions were formulated:

- How does the proposed decentralized structure measure against the existing centralized structure regarding RPA implementation at the PSC department?
    - How do data from two implemented RPA pilot cases, using the proposed decentralized approach, compare to an existing RPA project which was implemented using the existing centralized approach?

Answering these research questions will provide the authors with a basis for their recommendation with regards to the purpose.

1.3 Significance of the thesis

The results of this study will provide a basis for the company on how to proceed in their future implementation of RPA within the organization. Furthermore, the thesis is predicted to contribute, as a foundation of knowledge, in future RPA projects at the PSC department and other departments of the company. Therefore, the target group for the study is mainly the company but could also be useful for other large organizations susceptible to RPA-related structural change as well. The results of the thesis and its content could also be useful for RPA developers and tech-interested people in general. Lastly, the thesis could also play a role in future research related to RPA.

1.4 Limitations and delimitations

The thesis project was conducted at a packaging company in Lund, starting February 2020 and concluding in June the same year. It entails an investigation regarding the decision making, planning, implementation, and deployment of decentralized RPA in the company's organizational processes in comparison to the existing centralized approach.

The cases are limited to the departments Parts Supply Chain, European Service Administration Centre, and Americas Services Administration Centre. Note that comparisons therefore are conducted for projects developed at different parts of the company. Two cases developed with a decentralized approach and one existing case, developed from the existing centralized approach, are compared. The proposed structural change presented in the thesis is limited to the PSC department at the company.

The focus in the thesis lies heavily on the factors which were easily measurable, e.g. costs, development times, and lead time for decisions. As no investigation nor comparisons between departments at the company are conducted, the thesis is limited to the pilot cases conducted in the study and how they measure against the already implemented centralized case. As such, e.g. how RPA affects employees and other soft factors are not included. Moreover, the handover and follow up of the projects are discussed but not included in the results, as this has not yet been done for the centralized case and can therefore not be compared.

Additionally, when measuring the payback period and cost savings of the RPA developed in the two pilot test cases, this report uses simplified estimations including project investment costs and the time saved in utilizing the automations. The cost savings in the thesis' automation cases are measured by subtracting the time it takes to manually finalize the part of a work process which an automation did not manage to complete, from the time it takes to perform the original manual work process. Therefore, only the manual error handling is accounted for in deriving the automations' success rates and the process costs after automation. The report does not consider other costs such as: RPA licenses, costs of operating and maintaining the automations, RPA training costs, etc. However, costs with regards to processes which take place before the development are accounted for qualitatively when comparing the cases.

Not all data is used to compare between the three cases was normalized in accordance to 3.5 Analysis of data, Size factor. Decision and development lead times were left as they were, which might skew the overall comparison between the different cases.

Lastly, although there are several RPA providers, only one is used in this thesis as a basis for development, *UiPath*. Furthermore, costs for licenses or servers are not accounted for in the comparison.

1.5 Assumptions

For the thesis, the cost per hour for an employee at the company is set to SEK 450. This number was provided by the company and it will be used as a base for all the calculations regarding cost as a factor of time.

It was also assumed that the proposed decentralized approach to IT represents a change that could be viable for the company or similar companies that use the same centralized structure.

Lastly, it was assumed that normalizing the size of the projects, with regards to manual time spent on the original processes per year, was an acceptable approximation on how they compare relatively.

2 Frame of reference

This chapter contains important background to the topics presented in the thesis. The chapter will therefore act as a theoretical framework that the reader can refer to in order to better understand the analysis and the results.

2.1 Introduction

Streamlining work through digital automation has been done for quite some time, however, the automation market is still growing. According to UBS (Stiehler & Gantori, 2020), the global automation market size is projected to increase from $179 Bn 2018 to $238 Bn in 2021, an increase of almost 33%. One of the markets that will see the biggest increase is Process Automation, which is predicted to increase with $12 Bn. This is visualized in Figure 2.1.

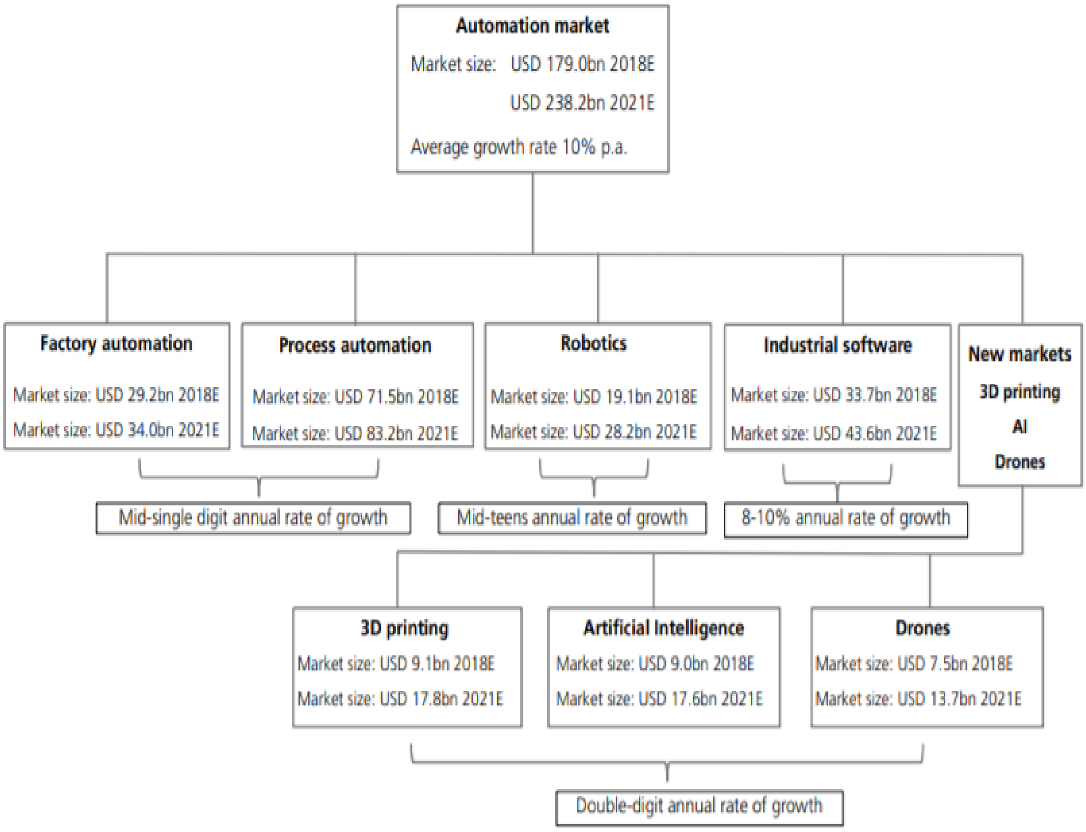

Figure 2.1: Projected increase of the automation market (Stiehler & Gantori, 2020)

Process automation or business process automation (BPA) refers to the use of digital technology to perform routine operations usually performed by a human. Automating business tasks using process automation plays a crucial role in an employee's work life. Tools such as Microsoft Azure or Google's Analytics make work more effective by automatically visualizing and compiling data which lets the employee focus on more important tasks that require strategic thinking and emotional intelligence. However, these tools are limited to their own process and cannot dynamically interact with other tools without integrated application program interfaces (API's). To account for these gaps, other tools were developed. For example, HP Sprinter is a tool that records, and when used, could mimic the actions of a user. Another solution is to create scripts using languages such as Python or Visual Basic for Applications (VBA) which can open applications and for example fetch a text field to be stored in Excel. However, these scripts or programs are restricted in being dependent on the consistency of the task at hand and often run into errors if the programs they work in are updated. They also often require IT-knowledge and some degree of coding experience to write. This could leave a skill gap if the employee who wrote the script moves on to other work or if an outsider wants to implement a simple change.

2.2 Robotic process automation

Robotic Process Automation or RPA is a process automation technology and a hypernym for tools that operate on a user interface in the same way a human would do (van der Aalst, Bichler, & Heinzl, 2018). UiPath, one of the main providers of RPA software describes the technology as "...the technology that allows anyone today to configure computer software, or a "robot" to emulate and integrate the actions of a human interacting within digital systems", i.e. a robot that mimics mundane digital tasks performed by humans in a more efficient way. It is important to note that RPA, as the quote implies, uses robots integrated into the software and not actual physical ones. It is therefore in many ways similar to how a script works. However, RPA software is built to be intuitive and requires little to no previous experience to understand. Furthermore, since most of the big RPA providers have supported plugins and frameworks for common enterprise resource planning systems such as SAP (Systems Applications and Products), updates are handled automatically. If a change is patched in one of the tools accessed by the RPA software, this will be accounted for in the supported plugins. Whereas if a script is used, coded by a local IT team or an independent employee, this bug has to be found and fixed which can be time consuming. Since RPA only communicates with other software programs via frontend as opposed to backend which most other automation tools do, it does not require any changes in the underlying IT-system to be implemented. This further explains that the technology mimics the human use of computer systems. However, replacing work done by humans also removes the important intelligence which a human possesses. RPA can therefore, although not exclusively, be described as limited to processes with deterministic outcomes, structured data, and ruled-based steps with routine characteristics (Aguirre & Rodriguez, 2017).

2.2.1 Benefits and limitations

The benefits of RPA are many and defined loosely depending on scenario or applicability on a specific process. However, the possible benefits can generally be described by three major areas. The first one being cost savings. In a case study Xchanging, a London-based business process and technology services provider, who by 2016 had deployed 27 software robots, reported an average saving of 30% on each automated process (Lacity & Willcocks, 2016). An article by Aguirre and Rodriguez (2017) further supports cost saving as a benefit. In their case study they measured cost saving on the basis of productivity improvement and the result was that RPA could increase productivity of the process by 20% compared to performing the task manually as done previously. Another benefit is the increased process speed. Willcocks, in her interview with McKinsey, argues that a software robot can outperform employees not only in speed, but also on quality and efficiency (Lhuer, 2016). This in turn shortens lead times and allows organizations to react and adapt faster than previously. As Willcocks suggests, the last major benefit is error reduction. In the previous mentioned case study with Xchanging, a process which previously took a person two hours and five minutes to do, can now be done in five minutes without any errors (Lacity & Willcocks, 2016). It can also be argued that there are other benefits that are more difficult to measure. Automating time consuming business tasks free up time which can be spent on tasks such as strategy and process improvement, thus creating value for the company. There are also suggestions that RPA can lead to enhanced freedom when organizing work and enhanced utilization of complex skills for employees (Engberg & Sördal, 2019).

As previously mentioned, RPA also has its limitations and downsides. Since RPA is limited to tasks with deterministic outcomes and makes decisions based on rule-based clauses it cannot be applied to tasks that require creative thinking. Neither can it localize and identify parts of a task that could be improved which an employee would. Even though some RPA software already has support for Intelligent Process Automation (IPA), until it becomes smart enough to adapt to changes in the workflow, automated tasks will still need some form of supervision by an employee.

Another factor which must be considered is how the employees actually perceive RPA and how susceptible they are to the change in their work. Despite Lacity and Willcocks reporting in 2015 that RPA adopters had promised their employees that automation would not result in layoffs, Asatiani and Penttinen (2016) argue that employees may still see robots as their direct competition. The authors argue that consequently, tension may be created between the employees and management, even destructively influencing the employee's morale.

2.3 RPA life cycle

There are many ways of categorizing the life cycle of an RPA implementation. The six steps described (Figure 2.2) are derived from the authors' own experience, RPA providers and educational sites, including Edureka, UiPath and Blue Prism.

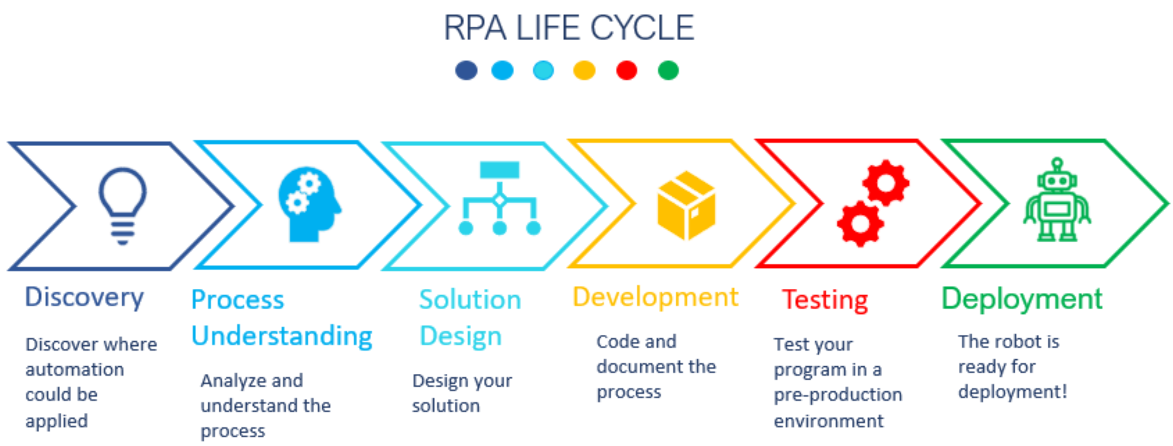

Figure 2.2: The six steps of the RPA life cycle

Since many large organizations already use some sort of software or tool to automate time consuming processes, knowledge of where automation is necessary already exists. This makes the first step of the life cycle straightforward. The client's requirements are analyzed and the benefits from automating the process is penned down. However, in order to fully understand if using RPA is the right approach requires an understanding of the technology and its limitations which makes up the second step. As previously mentioned, not every process is ideal to automate using RPA. The process must be deterministic and follow rule-based steps which the robots can imitate. Nevertheless, even though a process has these characteristics the implementation could still be time consuming and difficult if the wrong process is chosen. It is therefore imperative to investigate and get a thorough understanding of both the underlying and surrounding factors of a process before deciding to automate it using RPA. If a process has been deemed viable and the decision to begin has been made, the next step, step three, is to design a solution. The solution can be visualized in many ways, for example using simple flow chart instruments. This part of the life cycle is important to develop a solution which is smart and intuitive. If e.g. the program is developed with regards to possible changes to the process in the future, these can be easily implemented, and downtime can be avoided. This solution design could of course be subjective to change. However, if you have followed the second step of the process carefully, the change should not have a substantial impact on your solution. The fourth step of the process is developing or coding. During the development phase one of the most important parts is documenting. It is not necessarily important to document the code as much of it might not be used in the end. However, documenting the process while referring to the solution design is key for other developers to understand how the program was created. It is also important to keep in mind that smart code is not always intuitive and writing something which is easy to understand could be smarter. Although iterative testing is conducted throughout the development phase, the fifth step of the process is testing the finished program. Testing should be rigorous and cover all aspects of the process and could therefore be performed by either a designated testing team, the development team or the employees who are responsible for the task which has been automated. It can also be conducted by all of the above-mentioned. Since a lot of tasks include handling sensitive data which can affect the company, testing is usually done in a pre-production environment. When testing is completed, the final code should be documented for future developers and user manuals should be written. When all of this is done, the robot is ready for its deployment, step six. Note that the deployment phase is seldom the last step of the process, but that continuous maintenance will be needed to account for potential overlooked parameters and future changes.

To conclude, the RPA life cycle consists of six steps where focus lies in understanding the problem and process at hand. To avoid complications, the process should also be documented, and rigorous testing should be conducted before release. The life cycle of RPA is therefore not necessarily different from any other software development. Although it leans more towards the traditional project management approach often called waterfall, where the scope and requirements are highly controlled and predetermined.

2.4 Managing organizations

When implementing new technology, it is important to have a holistic view of the benefits and risks involved and consider surrounding factors which might affect the implementation. In this thesis, two different perspectives of implementation will be discussed. This chapter therefore aims to give the reader a basic understanding of how organizations could be structured and how decision-making and development could be affected.

When an organizational structure is centralized, the strategic decision making and planning within the company are concentrated to a leader or a specific department. Centralization is argued to be the tightest means of coordinating decision making in the organization (Mintzberg, 1979). In the centralized organization there is a clear chain of command. The company can focus on a unified vision and as a result give stakeholders a clear message of what it wants to achieve. However, there are downsides as well. A common error of centralization is underestimating cognitive limitations. In complex conditions, limiting decisions to a single deciding entity could lead to “information overload”, the more information the brain tries to receive, the less the total amount that actually gets through (Driver and Streufert, 1969, Mintzberg, 1979). Furthermore, executives who make decisions from a top down perspective might lose the control to oversee and supervise that their decisions get implemented in the correct way or that their strategy is aligned with managers further down in the organization (Sull et al., 2019).

A common solution to these problems would be to decentralize. An organizational structure can be called decentralized when decision making has been disaggregated into divisions, managers or subunits who make their own decisions (Siggelkow & Levinthal, 2003). It is likely that managers further down in the organization have a better understanding of the processes they manage than executives deciding on company strategy. However, it is important to consider that they might lack the holistic view which is required to make a decision which would benefit the company as a whole. Another reason to decentralize is that it could lead to increased employee motivation (Mintzberg, 1979). Motivation plays a crucial part in both managerial and professional jobs. It has been shown that employee participation and empowerment enhance efficiency, growth, and innovation. It also increases employee motivation and trust for the organization (Dobre, 2013). Furthermore, giving power to middle managers trains them in decision making so that they can help the company grow and make executive decisions in the future (Mintzberg, 1979).

The decision of decentralization however is very complex, and there are many factors that could affect the outcome of it. One of the factors that must be considered is the type of company in question. For international companies it is important to adapt to local market needs and decentralizing to geographical area units could be a solution (Egelhoff, 1988). Furthermore, the more different types of businesses an organization is involved in, the more it needs to decentralize its operations (Chandler, 1962). For multi-divisional global companies, it is nearly impossible to have a black and white approach to organizational structures and they have therefore been forced to find a balance. Gellerman and Kaestle (1990), states “One of the ways they have attained this balance is through a new look at corporate structure. They have moved away from the either/or way of thinking regarding the centralization versus decentralization debate. They have taken the best features from both types of structures and developed a third alternative, the "hybrid" type of structure” (Lentz, 1996). Lentz further explains in her article that although there are many unique perspectives on the hybrid organization and its management practices, hybrid organizations are at their core best understood as organizations where every stakeholder in the company gets to express and discuss their interests (Alexius & Furusten, 2019).

When considering these hybrid structures and the versatile approach in managing organizations they bring, it is important to understand how each separate department functions in relation to the company. The versatility of the multidimensional organization is complex and therefore brings unique organizing challenges regarding activities, structure, processes, and meanings (Battilana & Lee, 2014, Annosi et al., 2018). Breaking down and analyzing departments and their functions separately from a bottoms up perspective might help the company to decrease this organizational complexity. Even though one department within the company works decentralized in their operations they might still be affected by and must rely on decisions out of their control. Therefore, in order to understand the structural effects on a department in this situation, it might be beneficial to set apart and compare decisions from operation before analyzing the organizational approach.

The same ideas apply for managing IT. The organizational structure for IT could be heavily centralized, with a single department making the decisions or decentral where each business unit has their own IT department (Lionel & Schell, 2018). IT-related decisions are integral for a modern company and they can range from developing a new application for a specific department to a company-wide rollout of a business process management (BPM) system. Taking the aforementioned development as an example, the IT management might also have to decide whether to outsource or keep the development inhouse. Outsourcing has its benefits, e.g. it could lead to greater capacity for flexibility for purchase of rapidly developing new technologies (Altinkemer et al., 1994, Lioliou & Willcocks, 2019) especially if the expertise does not exist within the company. However, outsourcing IT also entails various risks such as increased costs and loss of distinctive capabilities or critical skills within the organization (Claver et al, 2002, Lioliou & Willcocks, 2019). The complexity of these organizational dilemmas reinforces the notion of analyzing the underlying factors of a development before making a decision.

2.5 Summary

Automation has existed for a long time within business and commerce. However, new technology leads to new opportunities and the automation market is still growing. One of the sectors which is projected to see the biggest increase in market growth is the process automation industry. Process automation or business process automation refers to the use of digital technology to perform routine operations usually done by a human. The method that will be discussed further in this thesis is RPA. The benefits of RPA can loosely be characterized by three major areas, *cost savings*: which can be measured through e.g. productivity improvement, *increased process speed* and *error reduction*. However, it can be argued that there are other benefits as well, which are more difficult to measure. Automating processes free up time which can be spent on other tasks, such as overall strategy or process improvement, thus creating value for the company. The RPA life cycle consists of six steps, *Discovery*, *Process understanding*, *Solution design*, *Development/Coding*, *Testing* and *Deployment*. In order to develop a coherent solution, it is especially important to focus on understanding the process which is being automated.

In this thesis, two different perspectives of development and implementation are discussed, centralized, and decentralized. Centralization refers to the concept for which strategic decision making and planning within a company are solely concentrated to a leader or a specific department. Decentralization on the other hand refers to an organizational structure where strategic decisions are made independently within different departments of a company. Furthermore, when comparing these structures, especially regarding IT, it is important to dive deeper into the underlying factors. Decision-making must be set apart from development and deployment so that they can be compared both separately and as a whole with regards to the overlying structure.

3 Methodology

In order to write a valid and well-grounded report, criteria were set regarding what sources to include as a basis for the study.

- The *data* was gathered mainly from primary sources. However, decision lead times, execution times for the manual processes and input data errors were estimated by employees at the company and the data gathered for the centralized case was taken from secondary sources.
- Factual *Information* that acts as a background for the reader was taken primarily from academic articles but also cited interviews with experts within respective fields.
- *Information* about the technology and how it works was also derived from guides provided by RPA-developers, such as UiPath, Blue Prism and Automation Anywhere.
- LUBsearch and Google Scholar were the main *search engines* used.

These criteria were followed throughout the study.

3.1 Research strategy

There are several approaches when conducting studies that aim to investigate and compare different organizational structures. This study was performed as an investigative practical case study which aims to answer the questions asked through comparing quantitative data and using this data as base for a qualitative analysis. Data was gathered through automating two work processes from a proposed decentralized approach in different departments of the company. These processes and the implementation strategy were then compared to an already implemented solution that was done from a centralized perspective. The gathered data and the comparison were analyzed and discussed in order to examine and validate the findings. Much of the findings were therefore based on conclusions made by the authors and even though we aim to be objective, as Gummesson (2003) suggests, all research is somehow interpreted, and possible bias could occur.

3.2 Frame of reference

As the study aims to compare different approaches for implementing RPA technology within a large organization, it is important to understand the two different organizational structures compared in the thesis and their roles on decision-making and development of IT according to previous research. It is also important to have an understanding of the technology which is being implemented; how it works, how it differs from existing technology and what effects implementing the technology could have. Therefore, the first step of the study was to conduct a literature review and write the frame of reference. The frame of reference was written with the goal of giving the reader a basic understanding of the aforementioned categories and how they are connected to the thesis.

3.3 Method for development

The development of the two pilot RPA-programs presented in the thesis was done using a decentralized approach. The developers were situated close to the employees working with the process to have quick access to information and feedback on the development. To increase measurability of the data gathered from the two cases the exact same development process was followed for both programs. This process was divided into four steps (Figure 3.1) in accordance with the RPA life cycle presented in section 2.3.

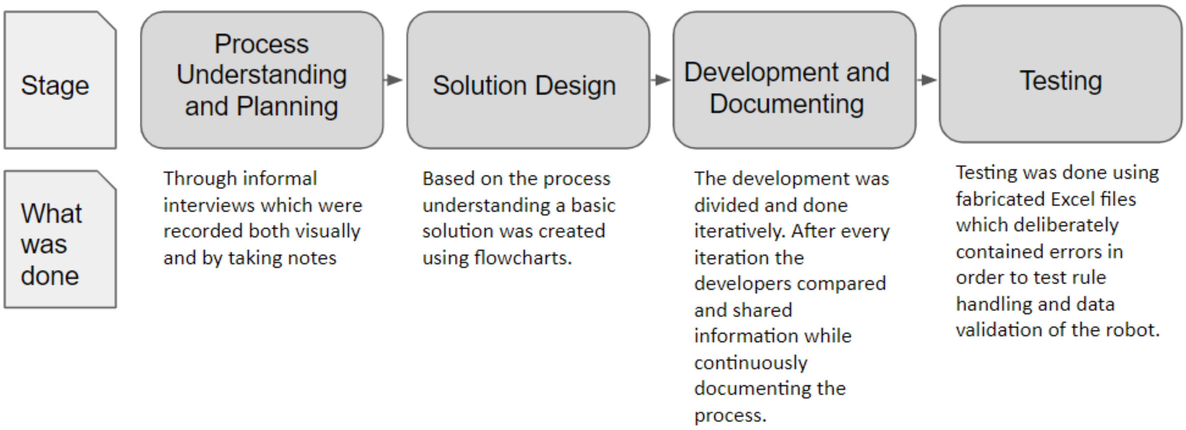

Figure 3.1: The four stages the development process was divided into

The first step was to understand the process. This was done through informal interviews where an employee at the company went through the manual work step by step while the developers took notes and recorded the screen. After this, a basic solution was designed using flowcharts. The structure of the programs was broken down into key parts which would all require their own specific function. The role of the functions, including input and output, were defined in order to create a framework which could be followed when coding. The coding and documentation of the program were divided between the two developers and information was continuously shared in order for the code to be both cohesive and coherent. Furthermore, the code was carefully documented and annotated throughout the development stage. This was done in order to deliver a solution that could be understood by others, e.g. the employees using the programs or future developers who want to implement changes. Any issues or questions that came to light during the development were immediately brought to the closest manager or employee with knowledge of the process. Although testing was done repeatedly during development the last step before deployment was testing. Testing was done using fabricated input files which deliberately contained errors in order to test rule handling and data validation of the robots. When all of the special cases that could occur had been tested, the robots were approved by the developers and managers, and development was concluded.

3.4 Data collection

Note that data collected from the two cases created with the proposed decentralized approach is primary data and the data collected from the case developed with a centralized approach is secondary.

Data from the centralized case was taken from a manager at PSC who was involved with and partly in charge of the automation request. From the manager, lead times and key figures were gathered. The other source used was a document from the affected department. This document described the "*as is process*" in detail and was therefore used as the main source in understanding the process. Since the development is still under way, no such document exists for the automated solution. Therefore, information about the solution was gathered through informal interviews with employees working with, or closely to, the robots.

The data gathered from the three cases were the following:

- Decision lead times;
- Development lead times;
- Project investment;
- Robot success rates;

The development time was gathered through careful documentation during the project. A logbook was kept in which both developers would enter what had been done and how long time they had spent on each task. Furthermore, the developers also presented a Gantt chart at the start of the thesis project which was revised and updated every week in accordance with the logbook. From this data the costs could be calculated as time spent per task (hours) times a regular hourly cost per employee at the company. The manual execution times and an estimation of number of errors in input data were taken from employees who work with the day to day execution of the process. Furthermore, the average number of articles which is processed per day for the first case, i.e. indirectly the time spent on the process per day, was derived through going through old logs of internal data.

3.5 Analysis of data

The analysis and the presentation of the results from the thesis project's three cases were divided into four different categories. *Decision lead times*, *Development lead times*, *Project investment* and *Robot success rates*. Furthermore, the payback period, which is a result of the two latter named categories, was presented as well. These results were then analyzed with regards to the different structural approaches, decentralized or centralized, the cases were implemented with. This resulted in an approach of comparison where the quantitative results derived from the cases were used as a foundation for the qualitative part of the analysis together with the structural factors.

Effort

The manual effort, in time spent to perform the process before automation, was naturally calculated as the number of jobs that are to be performed multiplied by the average time it takes to perform each job, see $\boldsymbol{Eq.}\,1$.

$$\boldsymbol{Manual\ effort_{before\ automation}} = \boldsymbol{Jobs}_{\boldsymbol{processed}} * \boldsymbol{Time}_{\boldsymbol{per\ job}} \qquad \boldsymbol{Eq.}\,1$$

Furthermore, the time manual effort left after the automation has run its course, depends on the number of jobs left undone by the automation. Therefore, this effort was derived as the number of unsuccessful jobs multiplied by the average time it takes to perform each job, see **Error! Reference source not found.**.

$$\boldsymbol{Manual\ effort_{before\ automation}} = \boldsymbol{Jobs}_{\boldsymbol{processed}} * \boldsymbol{Time}_{\boldsymbol{per\ job}} \qquad \boldsymbol{Eq.\,2}$$

Naturally, the number of unsuccessful jobs correspond to those jobs which result in errors.

Size factor

In order to make the quantitative data gathered from the three cases more measurable, a size factor was introduced. The size factor was based on the manual effort, in time spent to perform the process before automation. Since the central case had the largest amount of time spent, this case was used as a base of 1 or 100%. The other cases were then calculated as a percentage of the base. Divide 1 by the percentage for each case and get the normalizing size factor. Or the simpler way, showed in $Eq.\,2$.

$$Size\ factor_{case\ 1/case\ 2} = \frac{Manual\ effort_{before\ automation_{central}}}{Manual\ effort_{before\ automation_{case1/case2}}} \qquad Eq.\,2$$

Not all the data compared in the analysis was normalized. The data which was normalized was:

- Project investment
    - Development time and cost
- Robot success rate
    - Cost before and after automation
- Payback (as a result of the above normalized data)

The decision and development lead times were left as they were, even though the size of a project might affect these as well. Our comparisons of the resulting data for the different cases are based on the assumption that this factor is an acceptable approximation of the relative sizes.

Robot success rate

The accuracy of the implemented automations was calculated by dividing the number of successful jobs with the number of jobs processed by the automation, see $Eq. 3$.

$$Sucess\ rate = \frac{Jobs_{successful}}{Jobs_{processed}} \qquad Eq. 3$$

Time and cost savings

When calculating the time and cost savings for RPA, it was important to understand that the execution time of the automated process itself had little or no significance in measuring the net process speed increase. Therefore, the time spent on the process after automation only accounts for the time spent on error handling as a result of bad input data. This is true for unattended RPA as long as the automation can handle all incoming jobs during its runtime. For unattended automations, the net speed increase depends on the accuracy of the automation. For example, if the accuracy is a hundred percent, the net speed increase of the process will also be a hundred percent since the robot is completely freeing up the time it took to perform the original process. Therefore, in compliance to section 1.4 Limitations and delimitations, the time savings per performed process of the automations developed in the central case and in the pilot test case 1 and 2 were calculated by finding the difference between the effort before and after the automation. By multiplying this with the annual occurrence of the process, the time savings per year could be calculated according to $Eq.$ 4 and **Error! Reference source not found.**.

$$Time\ savings\ 1_{per\ year} = (Manual\ effort_{before\ automation} - Manual\ effort_{after\ automation}) * Occurrence_{per\ year} \qquad Eq.\ 4$$

$$Time\ savings\ 2_{per\ year} = Manual\ effort_{before\ automation} * Accuracy * Occurrence_{per\ year} \qquad Eq. 6$$

Furthermore, the annual cost savings were derived by multiplying the annual time savings with the company’s cost of that time according to **Error! Reference source not found.**.

$$Cost\ savings_{per\ year} = Time\ savings_{per\ year} * Cost\ of\ time \qquad Eq.\ 7$$

Simple payback

To calculate the return of investment on the automations developed in the central case and in the pilot test case 1 and 2, the simple payback method was used. It is calculated as initial investment divided by the return of investment per time unit. In the case of RPA, the return of investment was calculated as the manual effort saved by the automation per time unit. From this, **Error! Reference source not found.**, was derived.

$$Simple\ payback = \frac{Initial\ investment}{Cost\ savings_{per\ time\ unit}} \qquad Eq. 8$$

Lastly, as mentioned under the chapter description as well, the data presented in section *4.1* as well as *4.2* is primary data which was gathered during the project. Some of the data used for the analysis is secondary data. This data is presented in section *3.7 Background cases*.

3.6 Company structure

3.6.1 As is centralized structure

The company's structure regarding IT starts with deciding on a budget for each department. The budget is decided by the group executive committee every year and each department must argue and make a case for why they need a certain amount. The process from Parts Supply Chain's perspective is explained further below.

The first step of the process is that one or several processes with potential for improvements are identified and brought to PSC's global management team who prioritizes the suggestions. This usually starts as early as January since a budget is decided for the following year. Next, business cases, which include process explanations, potential improvements and cost savings are created. These are then brought to the group executive committee in the middle of June who then decide on how big the IT budget will be for PSC next year. A decision is usually made in late November or December. This budget can now be used by the PSC department freely for any IT-related changes. Note that this includes processes that were not brought to the group executive committee in the first place. When the budget is set, the PSC management now must divide it amongst themselves and decide on what projects should be prioritized. In the end, one large process could end up requiring the entire budget, leaving the rest of the department without any IT-related changes for the remainder of the year. This process is shown in Figure 3.2.

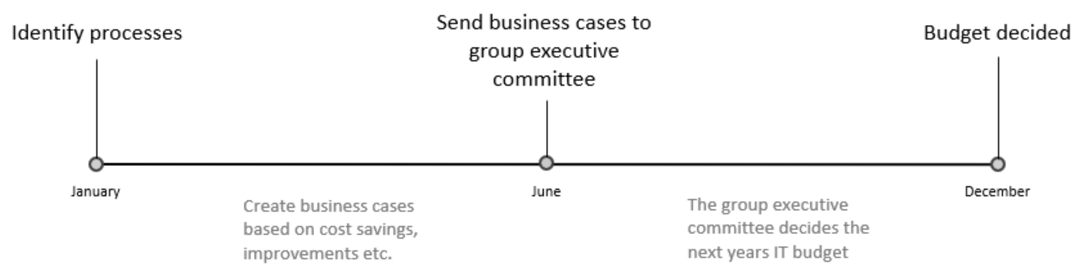

Figure 3.2: The steps between process identification and budget decision

When the budget has been set and PSC management has decided on which processes they want to prioritize, another chain of decisions and actions takes form. First of all, new business cases have to be made for the projects they want to perform. These business cases are similar to the ones brought to the group executive committee but are more focused on the operational aspects and software compliance etc. When this is completed, they are sent to the Order Fulfilment Service Product-team (OFSP) who checks if the proposed improvement is compliant with other processes or the systems in place at the company. OFSP also checks whether the change has already been implemented or is in the progress of being implemented somewhere else in the company. For example, if the proposed improvement is relatively small it could end up being put as part of a bigger case that is planned for development. This implies that the change will be pending until this larger case is implemented. These decisions are entirely up to the OFSP-team and left out of PSC's control. Here are some examples of what the OFSP-team can decide on:

- The case is not approved as it is not compliant or too difficult to implement;
- The case will be performed as a part of another case which is already under development or waiting to be developed;
- The case is approved as an independent project as the proposed improvement is compliant and as it is not a part of another pending case;

When the decision has been made that the case is compliant and that it can be developed, the OFSP-team forwards it to Global IT Management (GIM). Their job is to estimate and calculate the potential cost and project-time of the development as well as allocate the necessary resources, if PSC decide that the estimated cost is within their budget. If the required expertise needed to develop the solution does not exist internally within GIM, they can choose to outsource the project to a consultant agency. If the expertise exists, the development will most likely occur in Singapore or India since that is where most of the developers within GIM are located.

During the development phase, if the project has been outsourced, the first point of contact for the developers is GIM instead of the process users. GIM then contacts OFSP who, if they cannot help, contact the responsible process user. This process is illustrated in Figure 3.3. If the development is done inhouse, the first step of this communication chain is removed and the developers at GIM will talk to OFSP directly. It is worth noting that the process for writing business cases and process explanations at the company is performed thoroughly to reduce the need for communication between developers and process users.

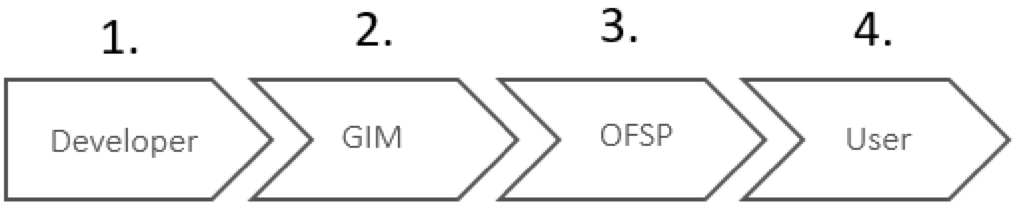

Figure 3.3: Contact between developer and user

Lastly, when the development is completed, the solution is sent to the OFSP-team. Before it can be used in the company, OFSP first has to ensure that it was developed as per instruction and that it works as intended. They once again investigate its compliance and test the solution in a testing environment. When they are satisfied with the result, they ask the final user to read through the documentation and test it in practice. OFSP then ensures that the new changed process is registered, and that future training of the process is updated accordingly. Furthermore, they also must update their tools used for reporting so that they account for the changes and lastly, they must ensure that employees using the solution are being trained properly.

This concludes the process for IT-related changes at the company from the PSC department's perspective. No follow up is being made with regards to actual cost savings or process improvement. Neither is employee satisfaction of the solution investigated nor reports whether the development took longer time or cost more than estimated.

3.6.2 Proposed decentralized structure

The proposed decentralized IT-structure shares similarities with the centralized IT-structure described above. Some things will remain the same, such as the decision chain for budget allocation. PSC will still receive their annual IT-budget every year and OFSP would still have to be counseled on compliance and would still have power on deciding whether a case gets approved or not. Therefore, the fundamental changes lie in how PSC would be allowed to use the budget if they would be able to circumvent it and the limitations that it brings.

The first proposed change to the structure regards the level of control PSC can have of their own budget and the projects that they want developed. Instead of allowing GIM to control the development and allocation of resources towards projects, PSC wants the opportunity to control this themselves. GIM would still be an integral part of the projects, and the developing or outsourcing might still be done by them. But GIM would act more as a supporting organ, than one which makes overall decisions. Allowing PSC to be an authoritarian part of this process could allow them to have the developers closer to the process if they believe it is necessary. This could speed up the lead times in the decision process. This control stretches further into the development process where decisions are being made about the solution and its features. One of the motives behind this proposed change is; if PSC (the managers or the users in charge of the process being improved) has full knowledge of the process and their goals, the lead times and effectiveness of the development could increase through eliminating complex communication structures and increasing development accuracy.

The second change regards the freedom of implementing IT-related changes outside of the designated budget. In the new proposed structure, PSC wants to be able to allocate resources from within the department to implement smaller changes outside of the designated IT-budget. First of all, in order to make this viable, UiPath for example, or other robotics related software programs, would have to be allowed to be used autonomically within the department. As of now this is regulated and not allowed. The second step is to differentiate between small, medium, and large projects and have different procedures and guidelines for each.

Medium and large processes would be run similarly to the central structure with the change mentioned in the second paragraph in place. The main difference would be that managers at PSC ideally could decide on whether to allocate their own resources outside of the budget if they deem it beneficial. As a result, another process within the department might be halted in the meantime. It is important to note that if sources from outside of the company must be used, GIM has to be consulted. While medium and large processes would be run similarly to the central structure, the big difference will be for projects that are deemed as **"light" or small**. In the current central structure, these cases fall under the same structure as large processes and are usually deprioritized or neglected. With the proposed decentralized structure in place, these cases would continuously be brought to daily management teams and then quickly sent to OFSP for approval. These smaller projects would be developed outside of any budget and resources would be allocated locally close to the process for each department. One of the goals of this is to create an environment where improvement and strategic thinking is a key part of the day to day operations. Having the ability to effectively automate smaller operational tasks would free up time for employees to think strategically, thus creating a cycle of improvement benefiting everyone at the department.

The main changes are summarized in the bullet points below:

- Increased autonomy and authority for the PSC department regarding IT development;
- Allow robotics related software to be used outside of the designated IT-budget;
- Allow PSC to allocate their own resources for development (outside of the designated IT-budget);
- Differentiate between small, medium, and large processes;
- Allow IT development of smaller cases to be a part of employees' day to day continuous improvement;

3.7 Background cases

3.7.1 Background pilot case 1 - 3PE case

This case was created by the Master Data Management (MDM) department at the company. They had acknowledged and brought to attention a set of work-routines that were done both manually and by using VBA scripts that could be improved using other automation technologies such as RPA. When done manually this process took approximately 652 hours of work per year since 150 articles are processed per week on average. This totals a yearly cost of SEK 293 400 for the company (Table 3.1). This set of work-routines were part of a bigger case that the department wanted automated. The automation request was sent to Global IM (GIM), which is the company's IT Management department, two years ago.

Table 3.1: Total time spent per year and the respective total cost

Time spent on the process per year (hours)	653
Total cost per year (SEK)	293 400

However, since this case was developed as a test from a decentralized perspective, the decision timeline (Figure 3.4) and process are explained below.

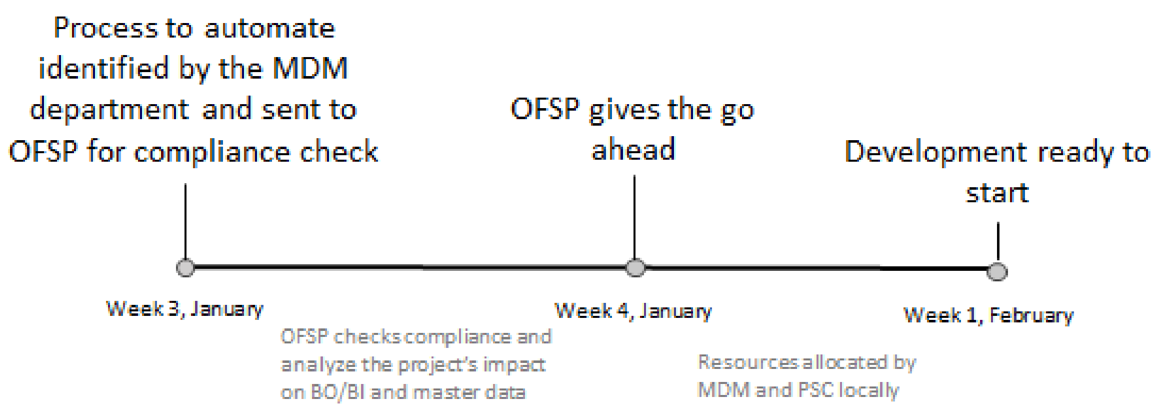

Figure 3.4: The decision timeline

The work process can be explained in three simple steps. Firstly, a supplier file is sent from the supplier and received by one of the MDM-team members. The supplier file is an Excel file that contains data on new third-party materials that are to be added to the company's product assortment. There are several different suppliers, and usually there are discrepancies in the data that they provide in their supplier-files. Since this data is later put into the company's SAP systems, it is important that the data is in the correct format. Therefore, the data first must be filtered, which is step 2. As of now, some of this filtering is done manually and some is done using VBA-scripts to remove deterministic errors. After the data is filtered, the supplier file is divided into three separate Excel files, MDM, PIR and GTS. This is also the final step. However, the MDM and PIR file needs to be supplemented with data that must be retrieved from files within the company's own systems. E.g. every article sent by the supplier needs to have a corresponding company material number. If it does not already exist, one has to be created. When this is done, the three created files can be used for the next step in the process, which is entering the data into SAP. This case, however, will only focus on what is described in the process above. The process of filtering data, checking for errors, and splitting the supplier file takes substantial time if done manually. Therefore, the VBA-scripts were created to save time. However, many of the employees continually experience errors using these scripts. Since coding in VBA is very sensitive to field names and cell values a simple update can cause the script to crash. This is not only time consuming and requires expertise to fix, but it is also frustrating to whoever experiences the crash.

Therefore, new ways to automate this process were investigated. One of these was RPA. The goal of this project was to automate everything in the process that previously had been done manually and by using scripts. The robot's job would therefore be to copy data from a specified Excel file, i.e. the supplier file, filter the data and transfer it to - and create - three new Excel files which later can be used for SAP. A simplified version of the process is shown in Figure 3.5.

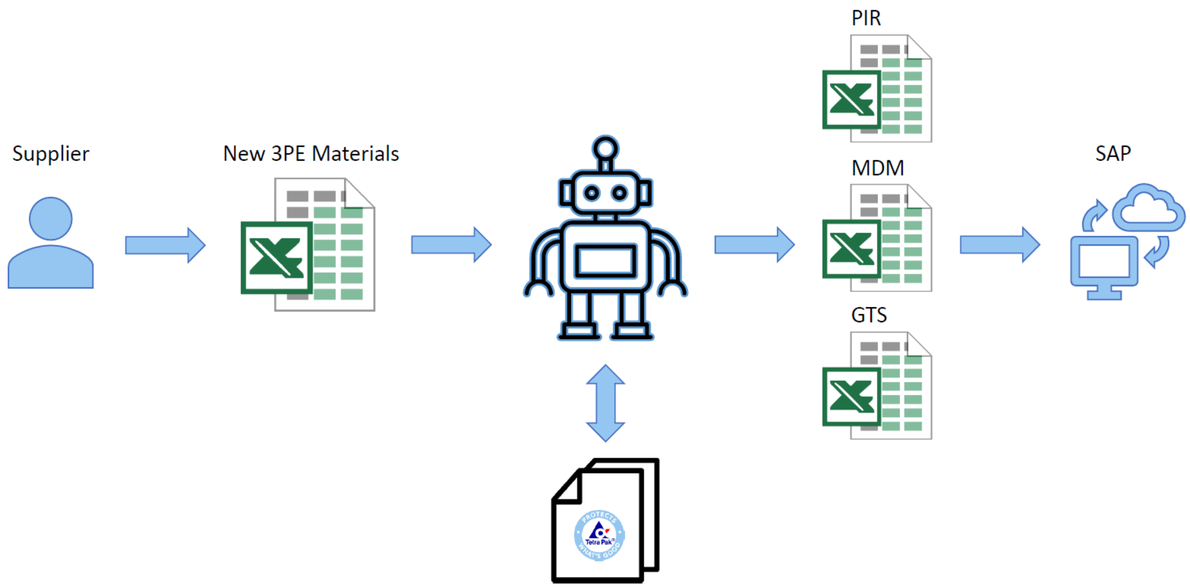

Figure 3.5: The process simplified

Manual execution time for this process is presented in Figure 3.6 and is measured in time spent per article in minutes. The estimated time to process one article has been estimated to **five minutes**.

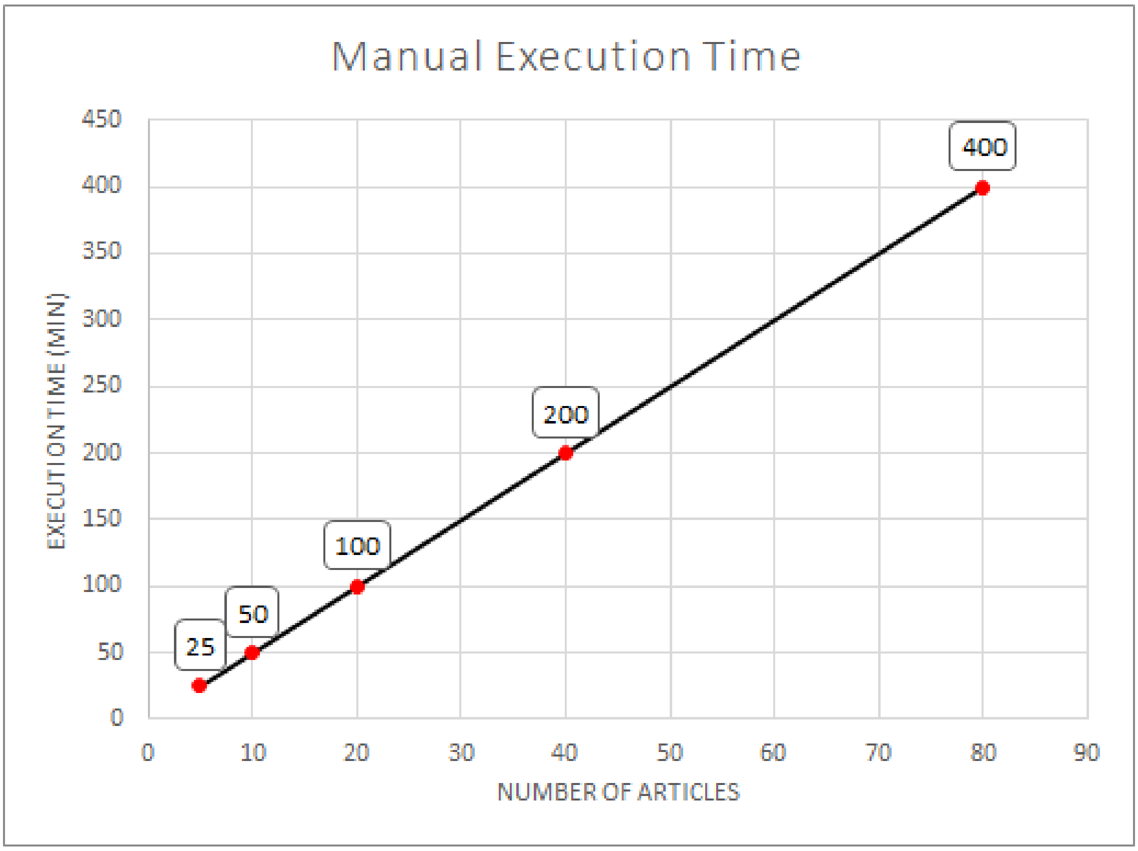

Figure 3.6: The manual execution time

3.7.2 Background pilot case 2 - LA lines

This case was created by the European Services Administration Centre at the company in response to a new EU legal demand, which requires suppliers to provide customer invoices with a dispatch date for any shipped goods. At the company, when goods were dispatched from the internal department Parts Supply Chain, the dispatch data was automatically transferred to the customers' billing document. However, for shipments from third parties e.g. Alfa Laval Kolding, the dispatch date had to be added manually prior to billing. ESAC projected that the manual process of maintaining dispatch dates on items, added 850 hours of work yearly (Table 3.2). Therefore, they wanted to pursue an automation of these data entries to drive efficiency and enhance quality. The automation request was sent to GIM, which is the company's IT Management department, two years ago.

Table 3.2: Total time spent per year and the respective total cost

Time spent on the process per year (hours)	850
Total cost per year (SEK)	382 500

As this process was developed from a proposed decentralized approach, the decision process leading up to development is visualized in Figure 3.7.

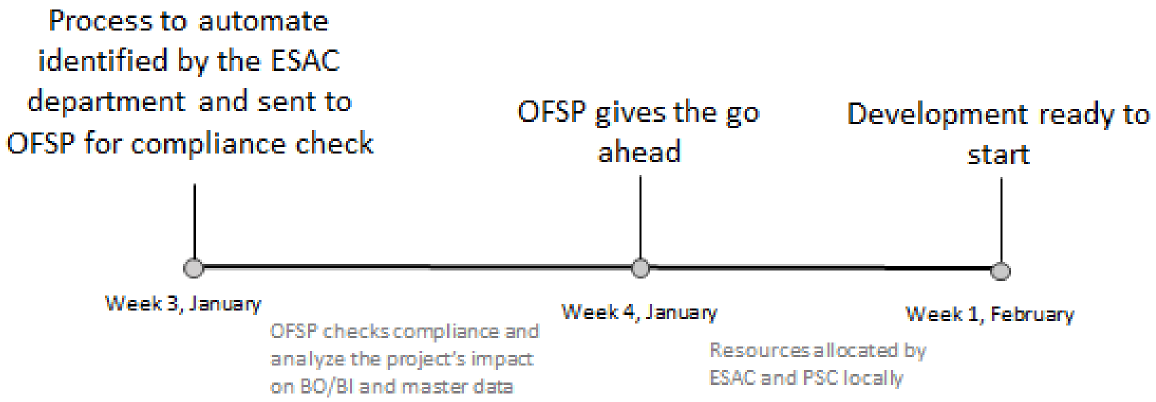

Figure 3.7: Illustrates the decision timeline for the admin case

The work process can be explained as follows. Updated dispatch dates, for the goods in a purchase order, is received by ESAC from the supplier. ESAC then locates and opens each of the corresponding purchase orders, in the company's system. Lastly, the purchase orders are updated with the dispatch dates for each of the materials contained in that order. This was a relatively simple process, but one that had to be repeated for all company purchase orders from third parties, in all of Europe. Furthermore, it had to be repeated for each material contained in each purchase order. Therefore, the time to do this manually, increased linearly with the number of orders and materials existing in a purchase order. Annually, the number of order lines totaled to approximately 30 000. To get a higher efficiency and to avoid possible errors made when performing this monotone task manually, it was decided to automate the process using RPA.

3.7.3 Background centralized case

The centralized case was developed by a consultant agency in cooperation with GIM at the company. The goal of the case was to automate the handling of orders for two of the larger customers in Brazil.

Before the development started, the project first had to go through a decision process. This process is explained generally in sub-section 3.6.1 *Company Structure* but is visualized and described briefly below with timestamps included.

When the PSC team had decided that this was one of the projects that would have priority for development, it was sent to OFSP for review and compliance checks. This was done week 3 in January 2019. During week 4 the projects impact on Business Objects and Business Intelligence software as well as master data was analyzed and added to the improvement request. The first two weeks of February were then used for meetings on what to prioritize and when the development could start. When this step was complete the project was handed over to GIM for estimates on cost and development time. Two weeks later, GIM assigned resources outside of the company to handle the programming and the development could start. The decision timeline can be seen in Figure 3.8.

Figure 3.8: Decision timeline for the central case

The lead times between each step is summarized in Table 3.3.

Table 3.3: Lead times between each step

Steps	**Time**
1 -> 2	2 weeks
2 -> 3	2 weeks
3 -> 4	2 weeks
Total	6 weeks

Process

The process for handling the orders is simple. Orders are received by email as an attached PDF-file which contains the necessary information in order to proceed. From here, the purchase order number is entered into SAP and if the order has already been processed, the customer will be notified. If the order has not been processed, a standard order is created (there are several steps when creating a standard order, but they were left out in order to keep the explanation of the process concise) and an order confirmation PDF is sent back as a reply to the customer. This process is almost the same for both customers and can be visualized in Figure 3.9 below. One discrepancy between the two is that the price is checked between the order confirmation and the original customer purchase order before the confirmation is sent. Another discrepancy is that the order attachments received by email look different, making it more difficult to automate as one solution. If there are errors, e.g. if vital information is missing in the attachment, BR PartsAmericas as well as the customer will be notified and the order will be put on hold.

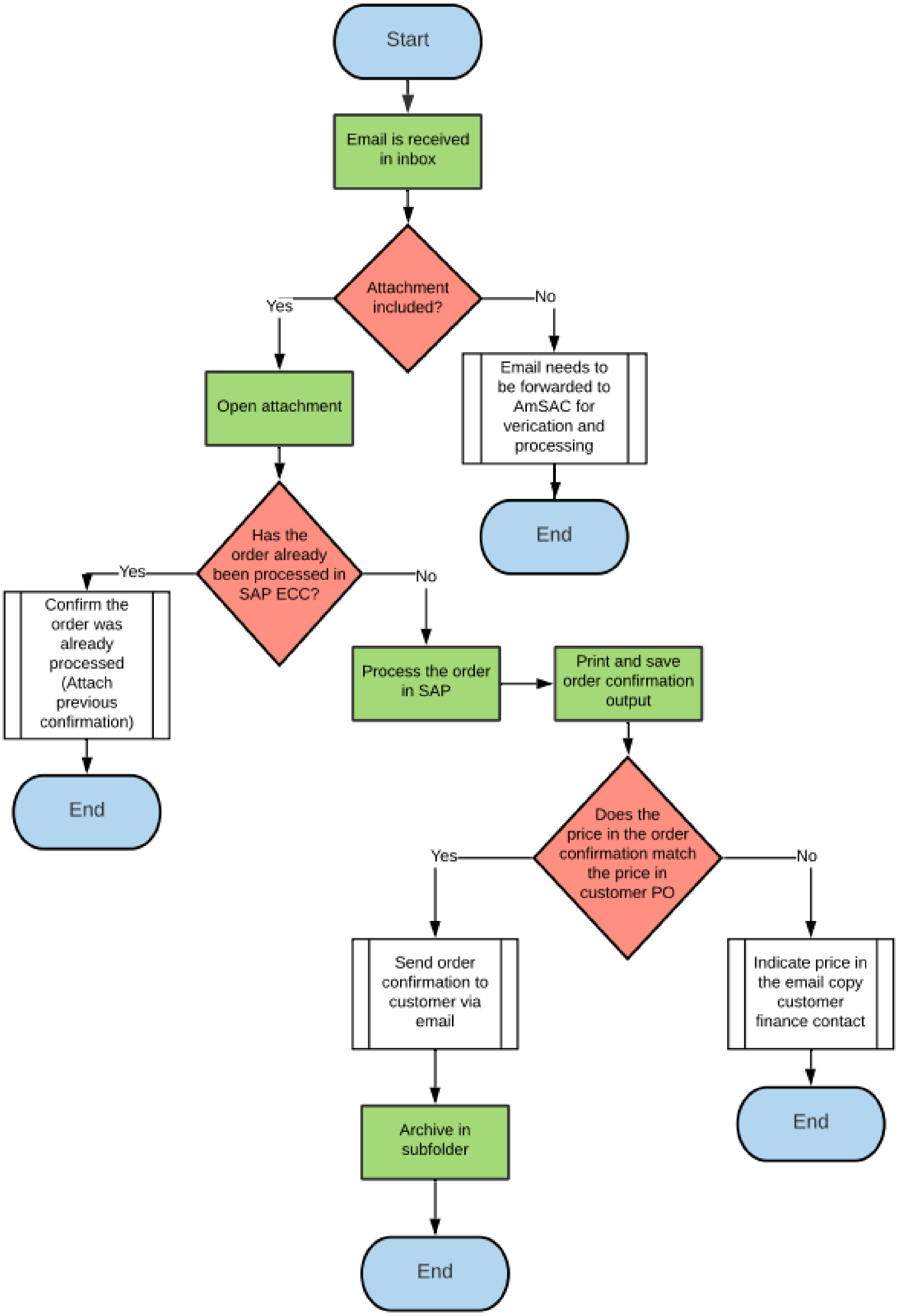

Figure 3.9: Process flow chart for handling orders

As with the other cases, this process is basic and does not require human intelligence if the data is correct. Therefore, the process takes up unnecessary time if done manually.

Estimations were made to:

- Effort in minutes per analyst/per order: 8 minutes
- Monthly volume per month: 850 orders

which totals 113,33 hours per month spent on the process. Using an average cost per employee of SEK 450 per hour costs per month were calculated to SEK 51 000 which adds up to SEK 612 000 per year as seen in Table 3.4.

Table 3.4: Total time spent per year and the respective total cost

Time spent on the process per year (hours)	1360
Total cost per year (SEK)	612 000

The automation of this process resulted in two robots which both work from a server and therefore do not need manual starting or stopping; one *Dispatcher*, which handles email-scanning and starts the job, and one *Performer* which handles the rest.

The solution process can be explained by the following steps:

- When an email has been opened the robot should identify the customer, either based on the email-address from which the email was sent or by using the VAT-number. VAT-number stands for Value Added Tax identification number.
- When the customer has been identified, the robot must match it with the company's internal database (ECC) in order to find the internal customer number.
- An order is created and saved in the company's database. If the mail and order could not be read correctly, a mail is sent to the admin team for the order to be handled manually.
- When the process is complete, an automated order confirmation is sent to the customer as a reply to the original email.

The robots were taken into production in May 2019, however, the robots only worked for one of the two big customers they were intended for. This means that for the customer which the robots worked for the project lead time added up to ca 5 months. The robots are still being developed for the second customer which means that the lead time is currently at 15 months and counting. Furthermore, for the customer that the robots work for, the success rate is 80%. This means that the robot handles 80% of the orders received by email without errors. The other 20% usually fail because of errors in master data, e.g. wrong material number. Since the robot for the second customer is still being developed, the success rate of the project in total is only 40%.

Since the project is still ongoing for one of the customers the numbers analyzed are taken from the start of March 2020. The development time totaled 1 552 hours and can be broken down into five parts. *Process Understanding*, *Solution Design*, *Development*, *Testing*, and *Problem Solving* as seen in Figure 3.10.

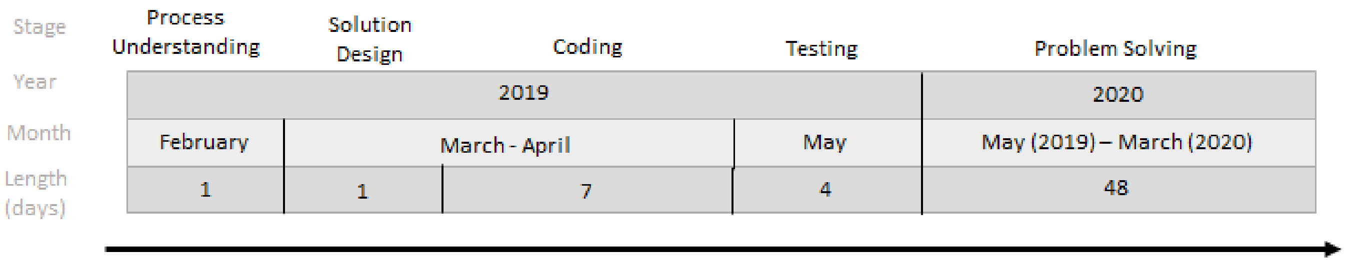

Figure 3.10: Development timeline for the central case

During these intervals, apart from process understanding, three developers from GIM and three other employees from the company were working on the development of the robots. During process understanding there were only 2 developers working at 25% from GIM. During the coding and solution design phase, the developers were working 75% of full time and the employees were working 25%. During the testing phase these numbers were 30% and 10% and for the problem-solving phase they were both 5%.

The total effective time spent on each task is summarized in Table 3.5.

Table 3.5: Total effective time and cost spent on each task.

	Department	Persons	Time spent in % of full time (per person)	Time (hours)	Cost (SEK)
Process understanding	**AmSAC**	0	0	0	0
	GIM	2	100	16	7200
Solution design	**AmSAC**	3	25	18	8100
	GIM	3	75	54	24 300
Coding	**AmSAC**	3	25	222	99 900
	GIM	3	75	666	299 700
Testing	**AmSAC**	3	10	48	21 600
	GIM	3	30	144	64 800
Problem solving	**AmSAC**	2	5	192	86 400
	GIM	2	5	192	86 400

Since full time employment has been assumed to be 8 hours/day the following time spent per task was calculated. The development cost therefore amounted to a total of SEK 698 400 (Table 3.7) and was divided between each phase as seen in Table 3.6.

Table 3.6: Cost and time together

	Time (hours)	**Cost (SEK)**
Process understanding	16	7 200
Solution design	72	32 400
Coding	888	399 600
Testing	192	86 400
Problem solving	384	172 800
Total	**1 552**	**698 400**

Table 3.7: Effective project time and total initial investment

Effective project time	**40 weeks**
Initial investment (SEK)	**698 400**

Since the execution of the robots are handled automatically the time spent on the task after the automation was calculated only with respect to error handling. Since 80% of the orders are handled without errors this leaves 20%. Using the previous estimation of a cost of SEK 612 000 per year for the manual work and dividing it by two, since the robots only work for one customer, gives us a cost of SEK 367 200 after automation. This totals a cost savings per year of SEK 244 800 which can be seen in Table 3.8.

Table 3.8: Cost before and after automation

	Before automation	**After automation**	**Cost savings**
Total cost per year (SEK)	612 000	367 200	244 800

Using the cost savings per year for the automation and the total cost of development as an initial investment, the simple payback period was calculated to **149 weeks** as seen in Figure 3.11.

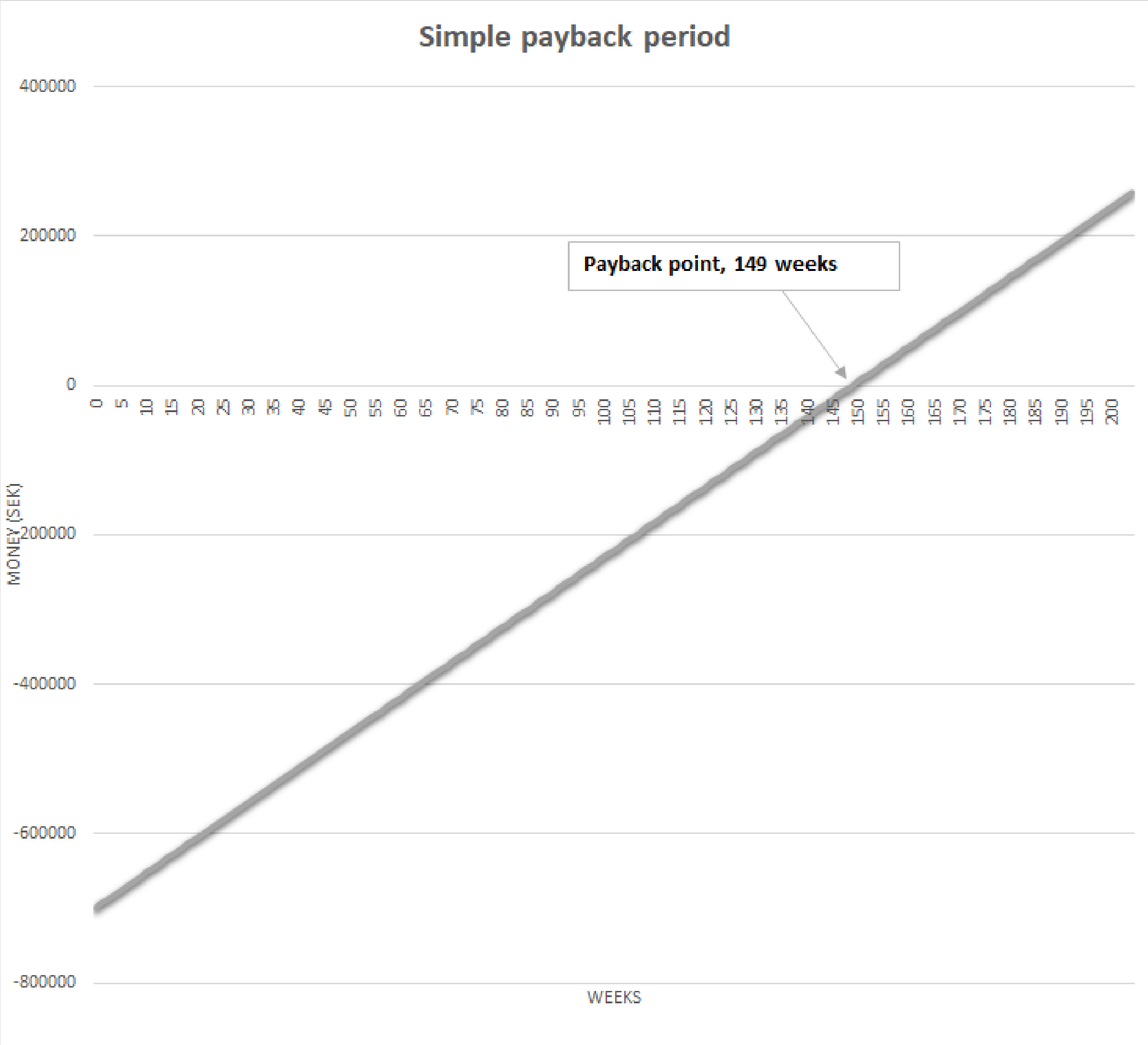

Figure 3.11: Simple payback period for the central case

To summarize and recap the result of the project. The development started at the start of March 2019 and resulted in two robots which were taken into production at the end of May 2019. However, they only worked for one of the two large customers they were intended for. The development time including the continuous problem solving has totaled 1552 hours so far which was calculated to a cost of SEK 698 400. Using the cost savings per year from the automation of SEK 244 800 the simple payback period was calculated to 149 weeks. Note that this payback period does not include the cost of future problem solving or development.

4 Empirical results

The data presented in sections 4.1 and 4.2 is primary data which was gathered during the study project. Some of the data used for the analysis is secondary data. That data was presented in the previous section, 3.7 *Background cases*.

4.1 Case 1

Note that the explanation of the solution presented in this section has been revised and shortened. The same applies for section 4.2 Case 2.

4.1.1 Automated solution

The first case was completed during a three-week period which included planning, process understanding, coding, and testing. During this period, there were two developers working full time on site. This was done using a decentralized approach where the developers were working close to the process and the manager in charge of it made the decision to automate. The case resulted in a fully working robot according to the goals set by the Master Data Management team (the robot has not yet been brought into production due to delays in handover as a result of the coronavirus epidemic, but has been tested rigorously). If there are no errors in the incoming data, the robot can perform the task 150 times faster than if done manually. However, with an estimation of at least one error per five articles and that the errors are handled manually, the success rate of the robot is 80%. The total cost of the project, the initial investment, reached SEK 129 600 and the payback period (when calculating cost savings from employees performing different tasks) is estimated to be 29 weeks. Furthermore, a user manual as well as a case report was created internally for the company.

The automated solution consists of six different functions that chronologically performs different tasks. The sequence of these functions and how they interact can be seen in Figure 4.1. The first function, Read Supplier File, reads the *supplier* or masterdata-Excel file and creates an internal copy within UiPath. The function then loops through the internal copy and validates the data according to a specific set of rules, note that it is only the internal copy that is validated and updated in this stage. If the data does not meet the requirements, it is either updated with the correct format if this is possible or updated with an error message to let the user know there is an error. Inside the first function exists two other functions which act as a support in order to validate the data. The first one retrieves the values from a specific column in the masterdata-file and uses these to look up the corresponding *material number* from the company's internal database. These material numbers are then stored with the corresponding value (from the masterdata-file) in a new data table inside UiPath. This data table will be used at a later stage. The second support function is similar but fetches a corresponding *class number* instead of a material number. These class numbers are stored inside a new data table as well. When this has been done, the Read Supplier File function overwrites the original masterdata-Excel file with data from the internal copy, so the user is notified of where in the file errors exist.

When the data has been validated and the necessary information has been retrieved, the program will start creating the Excel files. These files are created using predetermined column headers which are filled in with data retrieved from the data tables created in the previous functions.

An overview of the process sequence is presented and can further be visualized in Figure 4.1.

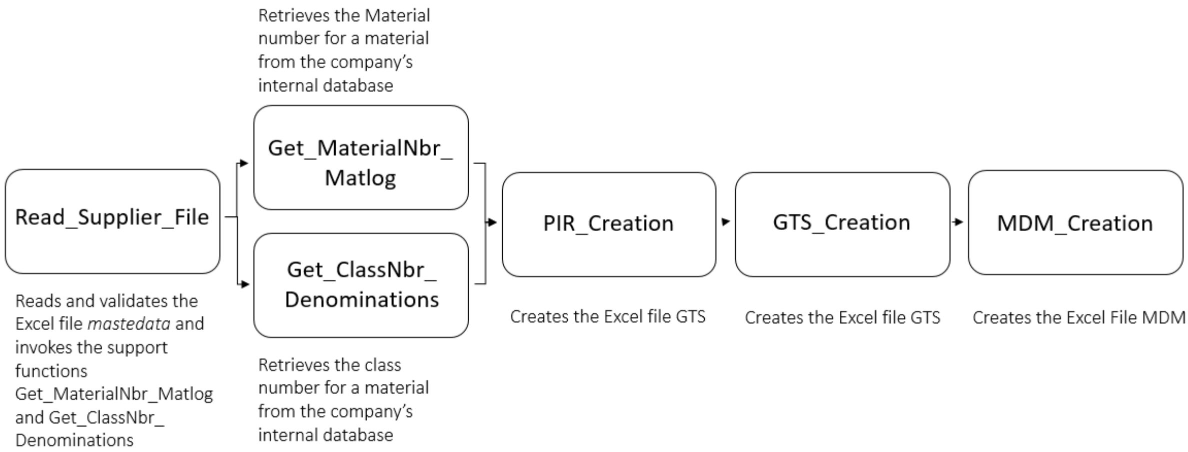

Figure 4.1: Overview of the process sequence for case 1

4.1.2 Quantitative data

Timeline development
The development timeline for case 1 is shown in Figure 4.2.

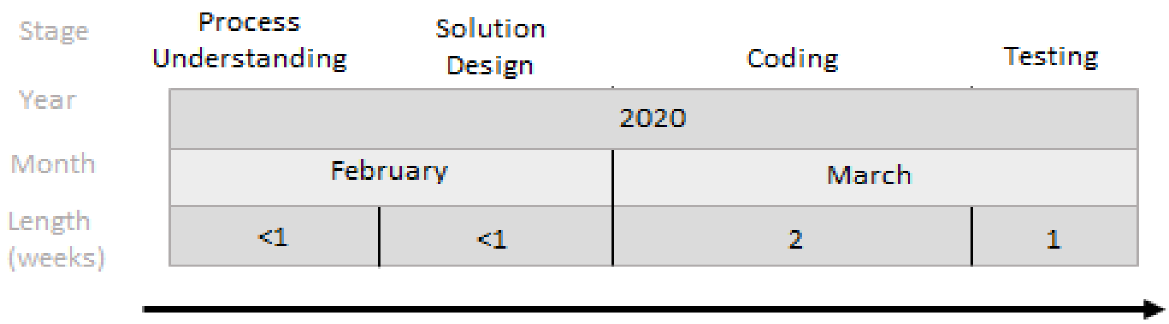

Figure 4.2: Development timeline for case 1

The development of the first program amounted to a total of 288 hours and a total cost of SEK 129 600 (Table 4.1) when using an employee cost of SEK 450 per hour. Most of this time was spent on development which accounted for 67% of the time spent and therefore also 67% of the total cost. These tasks were followed by testing which in turn accounted for 17%.

Table 4.1: Time spent per task and the resulting total investment for case 1

	Time (hours)	**Cost (SEK)**
Process understanding	32	14 400
Solution design	16	7200
Coding	192	86 400
Testing	48	21 600
Total	**288**	**129 600**

To summarize, the project was completed during a five-week period by the two developers, and the initial investment totaled SEK 129 000 which can be seen in Table 4.2.

Table 4.2: Total project time and investment for case 1

Project time	**5 weeks**
Initial investment (SEK)	**129 600**

Using the manual execution time of five minutes per article and the average number of articles processed per day, which was 30, the time spent on the process per year before and after the automation could be calculated. The numbers are presented in Table 4.3 and Table 4.4 and correlate directly to the average time spent per article. This was estimated to occur every five articles and the average time per article is therefore 1 minute.

Table 4.3: Effort before and after automation as a result of success rate for case 1

Annual effort before automation (h)	**Robot success rate (%)**	**Annual effort after automation (h)**
653	80%	131

Total cost per year was calculated using the numbers presented above for both before and after the automation.

Table 4.4: Cost before and after automation and the resulting cost savings for case 1

	Before automation	After automation	Cost savings
Total cost per year (SEK)	293 510	58 700	234 810

With a robot success rate of 80%, the cost savings were calculated as time spent on the process before the automation subtracted by time spent on the process after the automation. Savings per year totaled to SEK 234 810. Considering the initial investment for the project of SEK 129 000 and cost savings per year totaling SEK 234 810 the simple payback period was calculated to 29 weeks and is visualized in Figure 4.3.

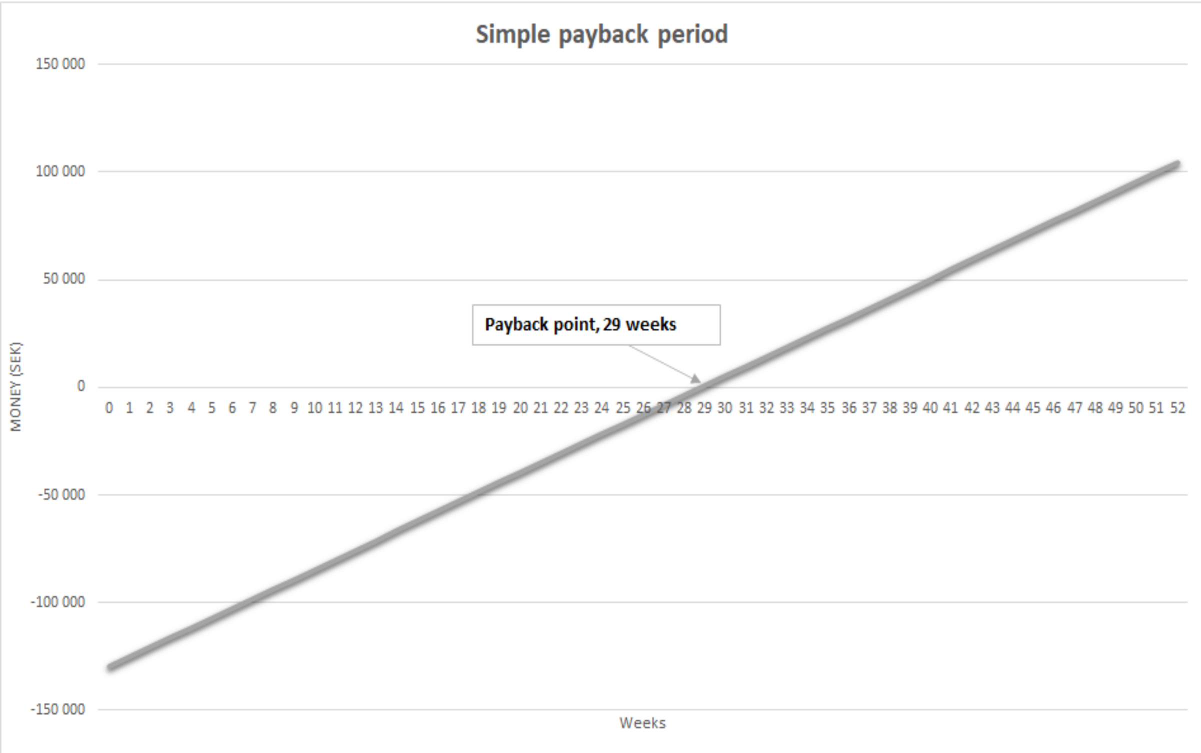

Figure 4.3: Simple payback period for case 1

4.2 Case 2

4.2.1 Automated solution

The second case was conducted by one developer during a five-week period which included process understanding, solution design, coding, and testing. The effective duration of the project was 120 hours. As in the first case, this case was conducted using a decentralized approach where the developer constantly worked close to where the business process was performed and closely to the process owner as well as the manager who made the decision to automate the business process. The case resulted in a fully functioning automation aligned with the automation requirements given by the project owner at ESAC. The automation's execution time was approximately five times faster than the time it took for a human to perform the original process. However, since the process was not a hundred percent deterministic, the practical success rate was approximately 60%. In other words, 40% of the input jobs could not be successfully completed by the robot. By registering these jobs in an output Excel file which was sent to the user, they could instead be performed manually after the automation had finished. Even though the robot did not encounter any problems during its thorough testing, successful jobs were also registered in an output Excel file to the user. This was done as a precaution since the process treats sensitive data inputs in SAP. If the robot were to fail during the execution, the resulting errors could be major as well as hard to undo efficiently. The total cost of the project, the initial investment, amounted to 54 000 SEK and the payback period was 12 weeks. Furthermore, a user manual as well as a case report was created internally for the company.

The automated solution consists of a main function which executes five smaller functions, each with its specific task. An overview of these functions and the sequence they are executed in can be seen in Figure 4.4. The first function, Init_SAP, launches the application SAP and opens the correct transaction where the data is supposed to be put in. The second function, Dictionary, creates a local data table and a dictionary from an Excel file that is created by the process user. The Excel file contains process input data purposed to be inserted into the SAP database by the robot. The created data table consists of all the purchase order numbers contained in the input data. The created dictionary consists of all the input data sorted in a structured way. It functions as an easy way of quickly fetching specific information contained in the input data. When the first two functions have been executed, the third function, PO_Open_New, is executed. This function does three major things, it opens purchase orders in SAP, then it initializes the fourth function, Materials_In_PO, and then it registers the progress in an output Excel file to the user. The fourth function checks if each material, in the current purchase order in SAP, exists in the dictionary correctly. If it does, the fifth function, Dispatch_Dates_Add, is initialized. Function five checks in the dictionary if the current material in the current purchase order in SAP, has data purposed to be inserted into SAP. If it does, it inserts this data. When function five has finished the automation jumps back to function four and checks the next SAP material in the current purchase order. This is repeated for all the materials in the current purchase order. In a similar way, when function four finishes, the automation jumps back to function three which opens the next purchase order in the local data table. This process is repeated until all purchase orders in the data table have been checked.

An overview of the process sequence is presented in Figure 4.4.

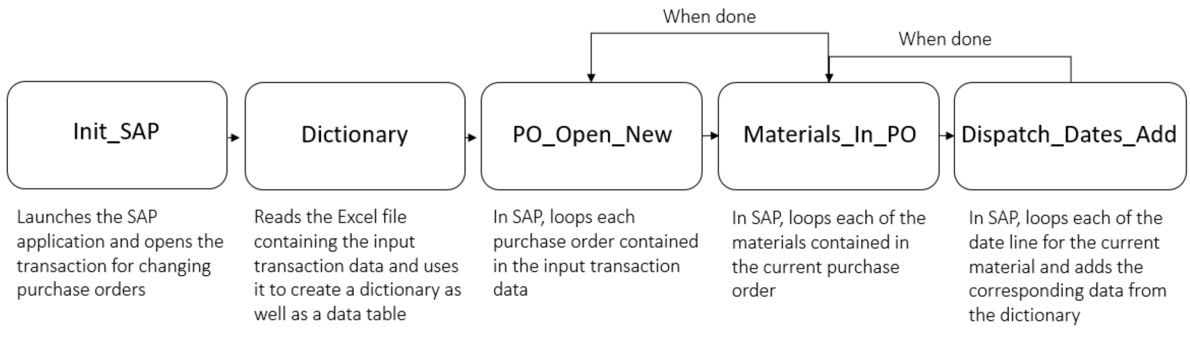

Figure 4.4: An overview of the process sequence for case 2

4.2.2 Quantitative data

Timeline development
The development timeline for case 2 can be seen in Figure 4.5.

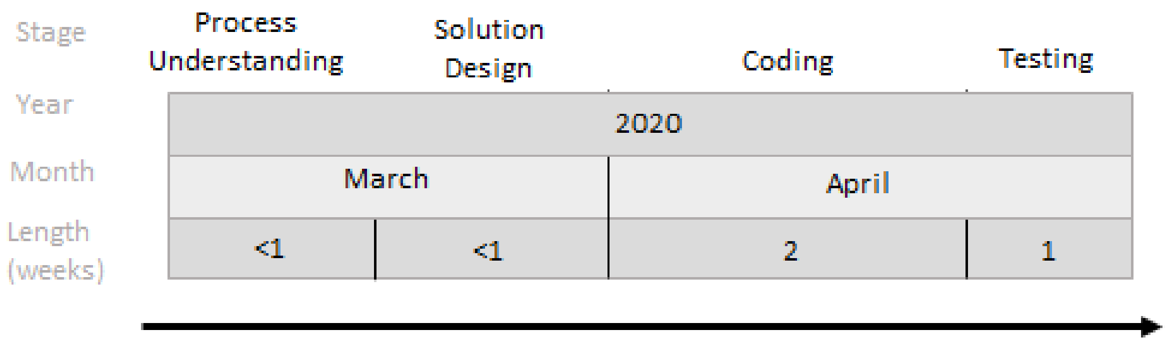

Figure 4.5: Development timeline for case 2

Development time and time spent per task
The development of the second program amounted to a total of 120 hours and a total cost of SEK 54 000 (Table 4.5) when using an employee cost of SEK 450 per hour. Most of this time was spent on development which accounted for 47% of the time spent and therefore also 57% of the total cost. This task was followed by testing which in turn accounted for 33%.

Table 4.5: Time spent per task and the resulting total investment for case 2

	Time (hours)	**Cost (SEK)**
Process understanding	16	7200
Solution design	8	3600
Coding	56	25 200
Testing	40	18 000
Total	**120**	**54 000**

To summarize, the project was completed during a five-week period for the one developer, and the initial investment totaled SEK 54 000 which can be seen in Table 4.6.

Table 4.6: Total project time and investment for case 2

Project time	**5 weeks**
Initial investment (SEK)	**54 000**

By using the annual effort before automation of 850 hours, and by combining this with the robot's success rate, the annual effort after automation was calculated. The result is shown in Table 4.7. The result directly correlates to the average time spent on handling each order line.

Table 4.7: Effort before and after automation as a result of success rate for case 2

Annual effort before automation (h)	Robot success rate (%)	Annual effort after automation (h)
850	60%	333

Total annual cost per year as well as annual cost savings were calculated by multiplying the result in Table 4.8 with the company salary of SEK 450.

Table 4.8: Cost before and after automation and the resulting cost savings for case 2

	Before automation	After automation	Cost savings
Total cost per year (SEK)	382 500	149 670	232 830

With a robot success rate of 60%, the cost savings per year totaled SEK 232 830. Using the initial investment and cost savings per year the simple payback period was calculated to 12 weeks as seen in Figure 4.6.

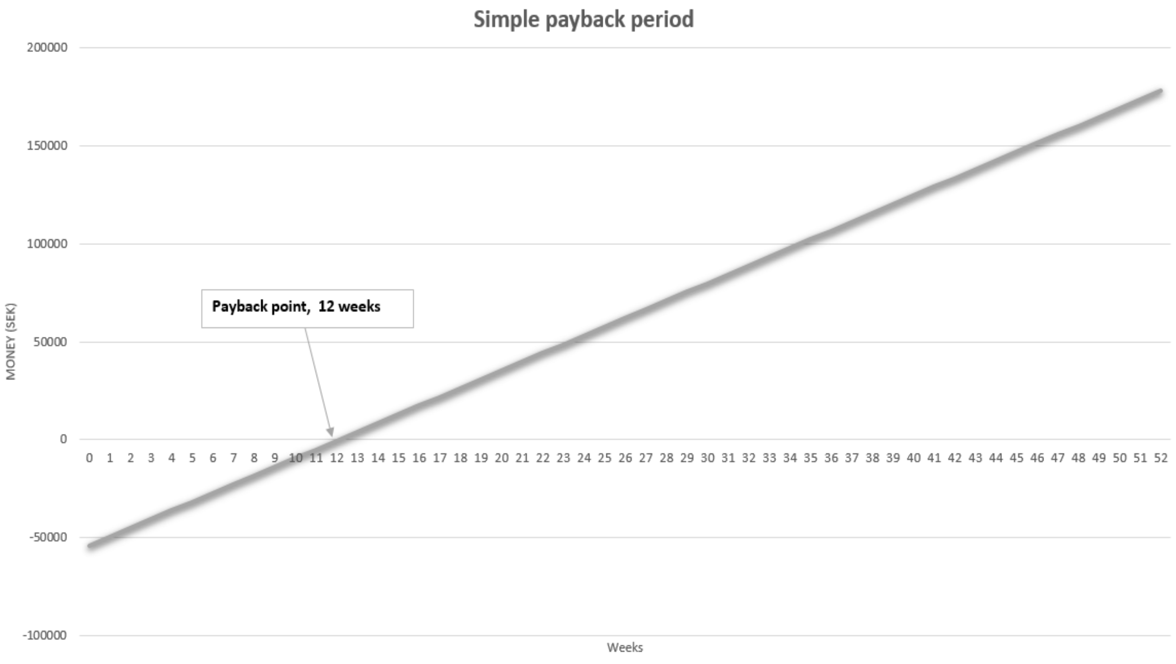

Figure 4.6: Simple payback period for case 2

4.3 Analysis

The analysis in this section will focus on the two IT-related structures presented in section 3.6 in methodology. This will be done by comparing the data presented in 3.7.3 Background centralized case with the data from 4.1 Case 1 and 4.2 Case 2 as well as the decision lead times presented in 3.7.1 and 2. In this section and throughout the rest of the report, case 1 will be referred to as MDM and case 2 – Admin. Note that the categories presented in this section are not mutually exclusive as many of them have interdependencies between one another.

Size of projects

Since the three projects are of different sizes, in order to be able to compare the cases, a normalizing factor was introduced, see *3.5 Analysis of data*. The factor for each case and how they compare against each other can be seen in Table 4.9. For the remainder of the analysis, the cases data will be normalized using each case's respective size factor. However, as mentioned in section 3.6 in methodology, this will not be done for decision and development lead times.

Table 4.9: Size factor for each project

	MDM (Case 1)	Admin (Case 2)	Central
Size factor	2.1	1.6	1

Decision lead time

The first comparison between the studies' three cases is the decision lead time. The decision lead time represents the timeline which starts when a project proposal is sent to OFSP and ends when the development starts. As can be seen in Figure 4.7, for the three cases, the central case had the longest decision lead time. Comparing step 1-2 for the three cases, the lead time was one week longer for the central case. For the central case, in step one a process had been identified and brought to the PSC management for prioritization. Given a small sized project, the proposed decentralized structure could have enabled the possibility of skipping this step by using existing resources and running the project autonomously outside PSC's budget. For the MDM and Admin cases, the projects did not require resource allocation and were therefore sent immediately to OFSP for a compliance check. Furthermore, comparing steps 2-4 for the central case with steps 2-3 for the decentral cases, the time difference is a result of the structural approach. Also here the time difference is due to the allocation of resources for the projects. In the decentral cases, this was done without involving the central IT department GIM. Resources were instead efficiently allocated locally, which reduced the lead time by three weeks.

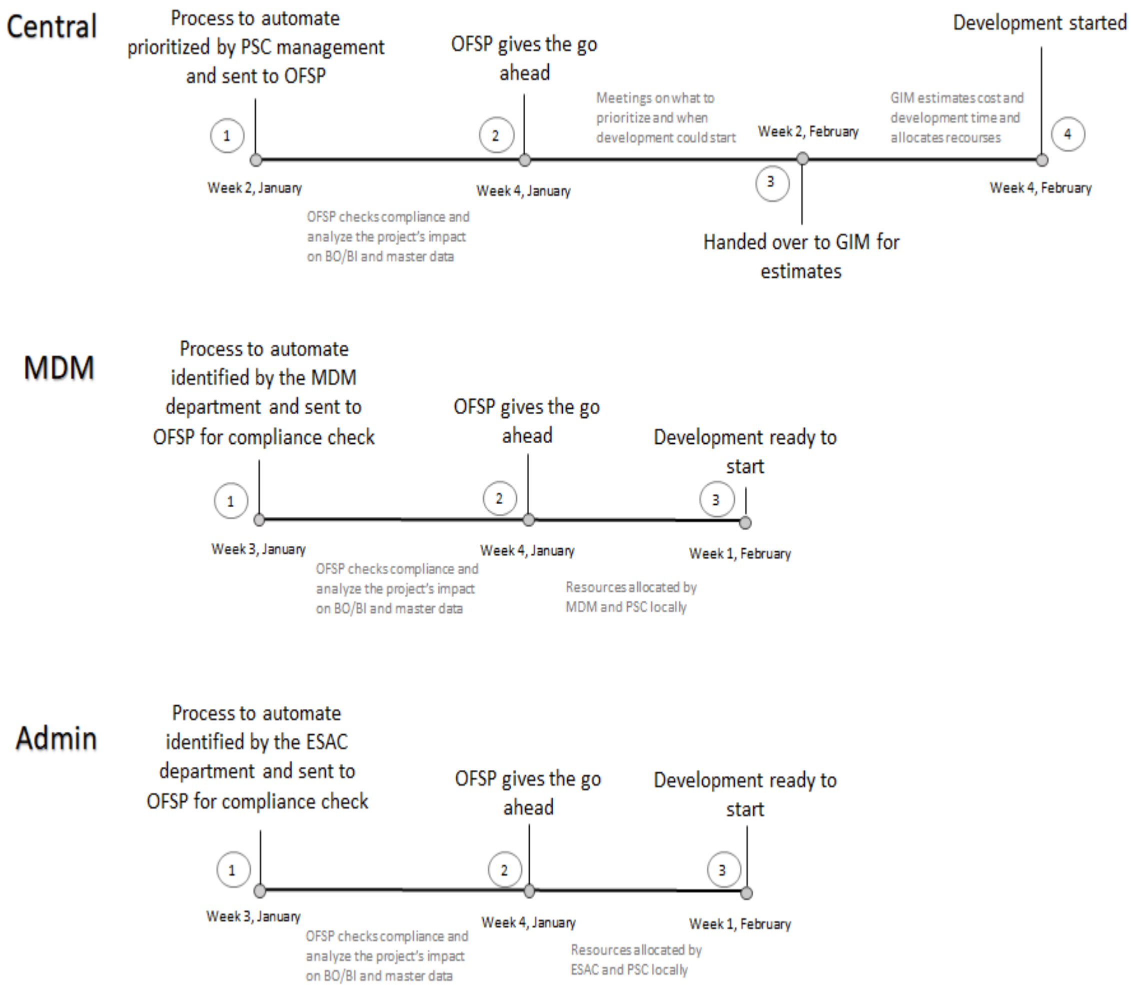

Figure 4.7: Decision lead times for each case

From the results it is clear that using a decentralized approach reduced the decision lead times. This was due to the fact that the centralized project had to be brought to the attention of the PSC management team, who only have meetings once a month, and also to GIM. For this specific case, these steps were relatively short. However, in theory if a project is unfortunately timed, these lead times could increase drastically. Comparing this with the proposed decentralized structure which includes fewer decision steps, it can be argued that the average decision lead time for these cases should be shorter. Having the opportunity to circumvent these steps by allocating resources from within the department, would partially remove or at least decrease the decision times. It can also be argued that the allocation of resources and the project cost estimation, most likely is done faster by employees familiar with the process. However, this also presupposes that the required expertise to do these estimations exists within the department. If it does not, with the proposed decentralization, GIM would still have to be asked for support. However, as mentioned in 3.5 Analysis of data, GIM would act as a supporting organ. Furthermore, it is important to note that if a decentralized structure is not handled properly, it could lead to loss of control and overall company strategy which also could affect the decision lead times.

Development lead time

In comparing how different structures affect RPA development, another important timeline to analyze is the development lead time which can be seen in Figure 4.8 below. This timeline takes place after the decision timeline, i.e. the time from the start of the automation project, the *Process understanding* stage, to the end of the last project stage, *Testing*. Note that the development lead time is not the effective time spent on each project, but it is the time span from start to end. Comparing the lead times of the studies' three cases, there are noticeable differences between the central case and the MDM and Admin cases. In total, the central case spanned over 13 months while MDM and Admin spanned over approximately 1 ½ month, respectively. The major time difference between the cases is due to an additional *Problem-solving* stage which was required in the central case. Without this stage, the central case would have spanned just over 2 months.

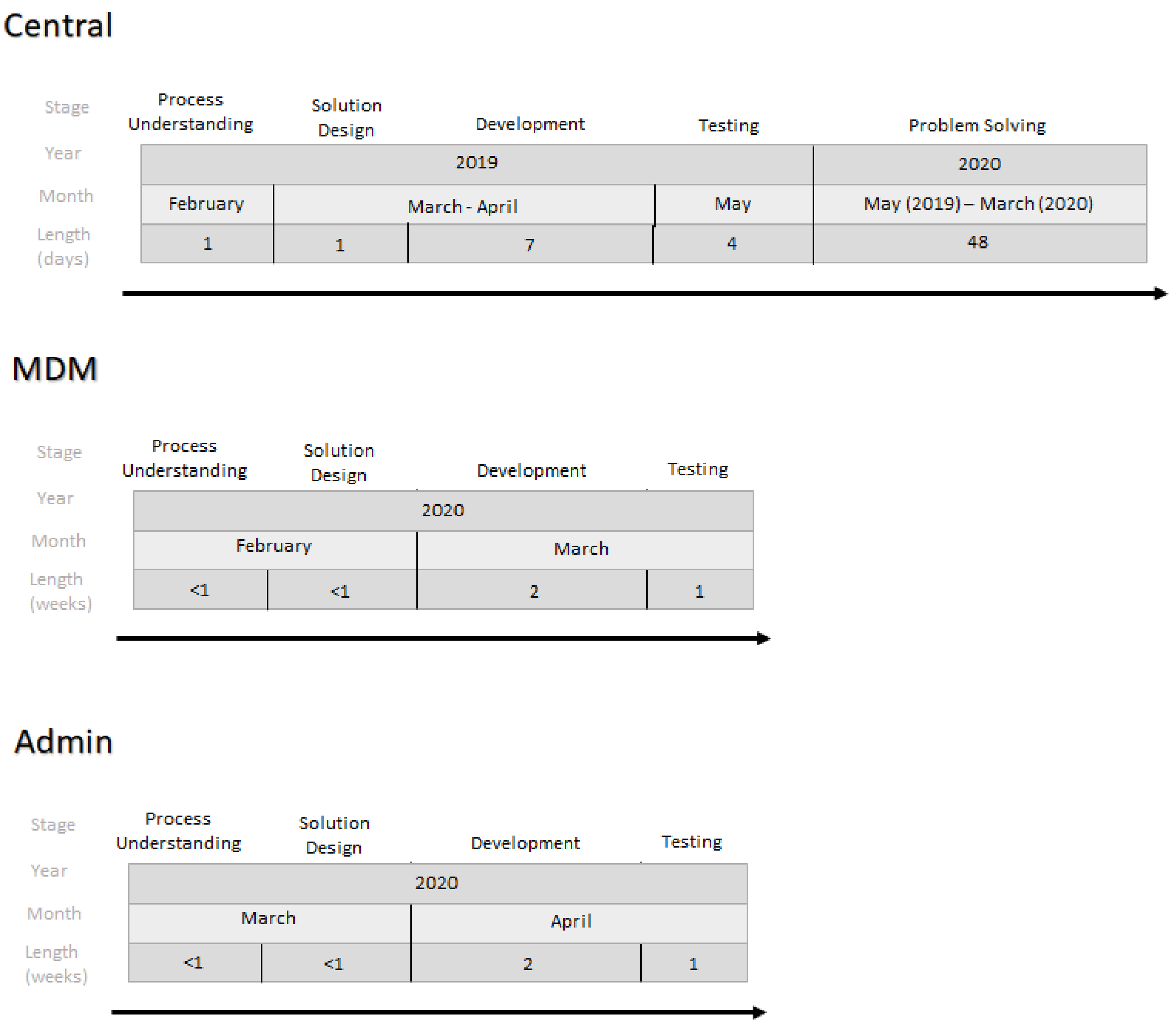

Figure 4.8: Development lead times for each case

Studying the development lead times presented in Figure 4.8, more in detail, the central case's *Problem solving* stage is further analyzed. The reason for the unproportionable length of this stage can be narrowed down to the low amount of effective work spent on it. As shown in Table 4.10, 4 people were working 5% of full time on this stage for 48 weeks. This was due to difficulties in communication which was a result of having the development far away from where the work-process was situated. Meetings had to be set up from two different time zones and both parties had to be available at the correct time. For the two decentralized cases, the *Problem solving* project stage did not occur. This could be a result of differences in complexity between the three cases. It could also be a result of the developers being situated closer to the process and therefore having a shorter and more efficient line of communication in case any uncertainties would arise. Therefore, one key takeaway from the results is that the shorter development lead times for the decentralized cases could have been a result of competences and deciding members not being separated geographically.

Moreover, a result of shorter project development times, is that the automation can be launched at an earlier time which implies an earlier first occurrence of cost savings. Furthermore, since RPA is a technology that is constantly developing, a theoretical downside with a project spread out over a longer time, is that the RPA tools used in the project might have improved greatly during this period.

Table 4.10: Number of persons per task and the % of full time spent on it (central case)

	Department	Persons	Time spent in % of full time per person
Process understanding	**AmSAC**	0	0
	GIM	2	100
Solution design	**AmSAC**	3	25
	GIM	3	75
Coding	**AmSAC**	3	25
	GIM	3	75
Testing	**AmSAC**	3	10
	GIM	3	30
Problem solving	**AmSAC**	2	5
	GIM	2	5

Project investment

As the project investment accounts for the direct spending connected to an RPA project it plays a major part in analyzing how the proposed decentralized structure measures against the central structure in place. In the study, the project investment is measured in time which is also translated into costs. Development time refers to the effective work hours spent on each case. From the results in Table 4.11 it is evident that the effective development times of the MDM and Admin cases are significantly shorter than the development time of the central case. Studying the table, it is clear that even though the three cases have been normalized using their respective size factors, the time spent on the central case still heavily outweighs the time spent on the Admin and MDM cases, by 810% and 260% respectively. As mentioned in the previous section, the most notable time difference is due to the *Problem solving*-phase in the central case. For the central case, this stage accounts for 25% of the total implementation time while it does not exist for the other two cases. However, there were also interesting findings on how much time that was spent on process understanding. As can be seen from the table, only 16 hours was spent on this stage for the central case. This is the only time where less time was spent on a part of the development process for the central case in comparison with the others. As mentioned in the frame of reference, understanding the process plays a crucial role in developing RPA programs. The extra time spent on problem solving could therefore be a result of lack of process understanding before designing the solution and starting coding. Since the two decentralized cases were developed from a method where process understanding was a key factor, the differences in process understanding might not be a result of the structural approach, but rather a difference in methodological approach. However, as touched upon earlier, the structure in place at the company and the geographic separation between the developers and the users for the central case could also have been a factor, since the development for the central case was done far away from the process, an "*as is process"* document was written by employees at AmSAC. This was the document used to understand the process which could propagate errors if written carelessly. Even if the document was written thoroughly, it could be difficult for other persons not familiar with the process to get a holistic understanding and design a fully functioning solution. Furthermore, this template looks the same whether the process is simple or complicated, thus it could be argued that it creates a stale environment which eliminates situational flexibility. Another aspect to consider is that it might also be difficult for the users writing the process template to understand what level of complexity and what language that should be used since they have no knowledge of the situational competence the readers of the document possess. This could lead to templates being written overly complex or on the other side, too simplified.

Another important aspect to analyze with regards to the two structural approaches is the number of developers that were designated to each project. From analyzing the data, it could be found that having fewer developers significantly decreased the time spent on the coding stage of the process. It could be argued that this might lead to a loss of valuable insight, however, since RPA projects aim to automate processes that are ruled based by nature, this leaves little room for innovative solutions. Thus, fewer developers could, even though at a marginal loss of creativity, lead to effective implementations. Why this is important in the structural context is because someone familiar with the process is more likely to have a grasp of how much creative complexity a process being automated would need. Outsourcing the allocation of resources solely to an independent department might lead to a loss of process understanding which is necessary in order to understand, inter alia, how many developers are needed. The proposed structural change argues that managers within PSC should have the opportunity to allocate resources and have authority over some automations themselves, especially light and medium sized cases, which might solve the issue regarding overallocation of resources. The Admin case is a good example. The potential cost savings for the admin case before normalization was SEK 382 500, which is only 1.6 times less than the central case. However, since the process was fairly simple to automate, only one developer was necessary which the managers realized. Furthermore, it is important to note that the developers in the MDM and Admin case had limited prior knowledge of implementing RPA before the project. This indicates that the coding phase might be even shorter in future decentralized cases of similar sizes.

Since the development of the central case was done far away from the process there had to be scheduled meetings held over the web. This might also be a reason for why the times differ as much as they do. Although no sources were found indicating that meetings held over the web are less effective than meetings face to face.

Table 4.11: Time spent on each stage normalized – for all cases

	Time spent on each stage (hours) normalized*		
	MDM	**Admin**	**Central**
Process understanding	67.2	25.6	16
Solution design	33.6	12.8	72
Coding	403.2	89.6	888
Testing	100.8	64	192
Problem solving	0	0	384
Total	**604.8**	**192**	**1 552**

Summarizing the data from the tables above gives us the following, Table 4.12. From this summarizing table it is more evident that even though the three cases have been normalized with regards to their size measured in manual effort, the effective project time and therefore also cost for the central case still outnumber the time spent on the other cases. The percentages on how much they differ were calculated above.

Table 4.12: Effective project time and total investment normalized - for all cases

	MDM	Admin	Central
Effective project time (weeks)	15	5	40
Initial investment (SEK)	272 160	86 400	698 400

Robot success rates

Comparing the structural approaches used in implementing the different cases, it became evident that the robots' success rates were affected. In Table 4.13, it is shown that MDM robot has the highest success rate and therefore also the lowest cost after automation, while the central robot has the lowest success rate and the highest cost after automation. The two decentralized developed robots had a combined success rate of 70%, while the central case resulted in a success rate of 40%. However, if the central case had worked as intended, for both customers involved, the success rate would have been significantly higher. Although the success rates are dependent on many factors, e.g. the quality of input data, the reason for the central case's low success rate is mostly due to the fact that one of the two implemented robots still does not work. Reiterating back to previous analysis in this section, one of the reasons for this might be that the developers had not fully understood the process and therefore designed a solution that was flawed from the start. Another reason could be the complexity of the automation which came as a result of attempting to implement one robot, which was to handle orders from two different customers. Even though these two order handling processes had major similarities, the orders handled differentiated aesthetically between the two customers. Therefore, it might have been better to implement two similar robots, each handling the two processes, respectively.

Table 4.13: Cost before and after automation as a result of success rate for each case, normalized

	MDM	**Admin**	**Central**
Cost before automation (SEK)	612 000	612 000	612 000
Success rate (%)	80	60	40
Cost after automation (SEK)	122 400	244 800	367 200

The existing central structure at the company is built for the development and implementation of large IT-related changes. This approach decreases the risk of duplicate work and reinforces the notion of having the right expertise handling the right tasks. Furthermore, as mentioned, it also leads to an increased control for the managers in charge of making the decisions as they get a clearer view of how the IT budget is being spent. Increased control could likely help managers execute the organization's strategy and fulfil long term goals. However, there are also downsides to this structure. As a result, smaller projects risk being dismissed or bundled together for the sake cost effectiveness. For the central case, although the processes of order handling for the two customers slightly differed, OFSP might have compiled these processes into one project instead of two different ones. Even though it could be more cost effective and more comprehensive to develop a solution that could benefit several processes at once, it is evident from the data that doing so can increase the project complexity and ultimately reduce cost savings.

Payback
As a result of the longer development time and lower success rate in the central case, the payback period is significantly longer than the payback period for the Admin and MDM cases, approximately 1250% and 510% longer, see Table 4.14.

Table 4.14: Normalized payback period for each case, normalized

	MDM	**Admin**	**Central**
Initial investment (SEK)	272 160	86 400	698 400
Cost savings per year	489 600	367 200	244 800
Payback period (weeks)	29	12	149

5 Discussion

Before drawing any conclusions from the analysis, it is important to reflect on the study and the chosen method since there are factors which could have affected the final result.

Firstly, a more well thought out and detailed case study could have been conducted. The case study conducted in this work is limited to three cases which differ in size and therefore most likely complexity. Conducting a larger case study on cases which are all of a similar size could have led to a more concise comparison. Furthermore, in order to make a fair comparison of the different structures presented in this work, a study comparing the development of the same process could be an alternative. However, since the structures have different pros and cons depending on the size of a process, this study should be conducted for several sizes.

Therefore, one of the factors which needs mentioning is the size factor used to normalize the three different cases to make them comparable. This factor was based solely on the manual effort the processes required and therefore lack in substance. It was mentioned under *1.5 Assumptions* that it was assumed that the factor gives an acceptable approximation of how the cases compare although they are different. However, disregarding e.g. the complexity of the processes could have led to an "unfair" estimation. Another assumption made was the cost per hour for an employee at the company. In reality this cost depends on which role the employee has at the company and might not be the same. This assumption could also have affected the final result of the analysis. Furthermore, since this thesis is limited to a certain department within the company, no knowledge or information on how the structural change might affect the company as a whole is provided. The central IT-structure in place regards the whole company and while the proposed structural change might be beneficial for the department investigated in the study it might be bad for another. This aspect has not been analyzed.

The method and data collection and the reliability of the data collected can also be discussed. Some of the data was estimated by employees and the reliability of this estimation was not investigated and/or analyzed. As these estimations have a direct effect on the results their reliability also directly affects the reliability of the study. Furthermore, the development of the two decentralized cases were divided into several steps. In order to make the comparison easier the development for the centralized case was divided into these steps as well. However, since no information was collected on how the development of this case was conducted and which steps were followed, the division could have affected the result.

There were some differences between the RPA life cycle which was presented in chapter *2 - frame of reference* and chapter *3 - methodology*. For example, the deployment phase was not analyzed and not used as a comparison between the cases. Neither was the handover (not included in the RPA life cycle) of the project a part of the analysis. A solution written with respect to future users or developers could take far longer to develop than a solution designed only based on short and effective lead times. This includes writing manuals, documenting the code etc. This was done for the decentralized cases but no information on if this was done for the centralized case could be acquired and it was therefore left out of the study.

It is also relevant to compare the findings from chapter 4 with previous research which was presented in chapter *2 - frame of reference*. The three clear mentioned benefits of RPA were cost savings, increased process speed and error reduction. The findings of this study match the findings of previous research, especially regarding cost savings. Increased process speed was also a finding but was left out of the thesis. From previous research there were indications that RPA could lead to enhanced freedom and utilization of complex skills. Furthermore, there were also indications that a tension could be created between management and employees as a consequence of RPA implementation. These factors were not researched in the thesis. The findings of this study also compare with previous research regarding organizational structures and organizational management. As found in this study, outsourcing work was in line with the risk of increased costs. Furthermore, it could also be concluded that managers closer to the process have a better understanding than executives or specific departments in charge of decisions.

It is also important to consider the roles the two authors of this thesis have had. It could be discussed if the results of this study represent a decentralized structure or merely a result of having two consultants working with implementation and development inhouse close to the process. This factor has not been analyzed. Lastly, since this study was conducted under unique circumstances, the conclusions presented in the following chapter are not general, but specific to the context.

6 Conclusions

This master-thesis set out to deliver a proposal of whether the company should decentralize parts of their IT-structure. In order to reach this purpose, the following research questions were asked:

- How does the proposed decentralized structure measure against the existing centralized structure regarding RPA implementation at the PSC department?
    - How do data from two implemented RPA pilot cases, using the proposed decentralized approach, compare to an existing RPA project which was implemented using the existing centralized approach?

These questions were answered by conducting a practical case study where an already implemented project developed with a centralized approach was compared to two projects developed with a decentralized approach. The data collected from each project was analyzed both quantitatively and qualitatively with regards to the different structural approaches. By comparing this data from the cases, the questions could be answered, and the authors of the thesis arrived at the conclusion of recommending the company to decentralize their IT-structure in accordance with the proposal presented in section *3.6 Proposed Decentralized Structure*. The basis of this recommendation and the answers to the questions can be read about in more detail below.

From the analysis it could be concluded that there are clear differences between the two different structural approaches, both positive and negative, in the investigated context. The first difference was derived through comparing the decision lead times of the project.

The results showed that through decentralizing, the decision lead times for projects would decrease. This is because PSC would be able to make the decision to allocate resources locally instead of having to go through GIM, skipping a step in the existing decision structure. However, it was also concluded that by removing GIM as a deciding organ, expertise could be lost. Although it was argued that in the proposed decentralized structure, GIM could still act as a supporting organ if any questions or problems would arise.

Another important difference between the two structural approaches is how close to the process the developers were situated during development. It was concluded from the analysis that being situated closer to the process has positive outcomes. The two most affected categories being development lead times and project investment. From the analysis it was concluded that the two projects developed from a decentralized approach had both shorter development lead times and lower effective project times (and therefore also lower project investment) than the centralized case. By being closer to the process the developers had an easier and more effective way of communicating with knowledgeable employees when questions or problems arose during development. Furthermore, since the developers in the central case were situated far away from where the process took place, they had to follow an *as is process* document written by AmSAC in order to understand the process. This process document could not only propagate errors if written carelessly, but it could also be written overly complex or overly simplified leading to misunderstandings and problems with the designed solution. It was also concluded that by being close to the process, neither deciding member nor competence must be separated geographically. These factors could also have been the reason for why the robot success rate of the central case was lower than the other two. However, since the complexity of the three cases differ and success rate also depends on input data, no conclusion could be drawn with regards to this number.

Furthermore, another aspect which affected the project investment is the amount of developers and personnel allocated to each case. It was argued in the analysis that the more knowledge about the complexity of a process you possess, the more suitable you are to make a decision regarding the allocation of resources. In the proposed decentralized structure, GIM would act as a supporting organ and PSC would be able to make the allocation decisions independently. However, it was also argued in the analysis that the difference in project investment partly could have been a result of a difference in methodical approach which was not investigated in the study.

Moreover, it was concluded from the analysis that decentralizing the IT-structure could lead to a loss of control for executives at the company and a loss of overall strategy. The central structure in place also minimizes the risk for duplicate work. However, since the proposed decentralized structure does argue for increased feedback between OFSP, GIM and the PSC department, it was concluded that duplicate work is not a result of decentralizing in this case. Furthermore, since the central structure in place has been built in order to implement large changes cost effectively it does not work well for smaller projects. In the analysis it was argued that if the automation for the two different customers were handled as two separate projects, the result could have been better. Therefore, the flexibility in handling both large and small cases makes a case for the decentralized structure.

From these conclusions with regards to the analysis and the presented data, the authors decided to propose a decentralization of the IT-structure in accordance with section *3.6 Proposed Decentralized Structure*.

7 Future work

Even though this study gave a recommendation to decentralize the RPA-related structure at the company, to fully understand the impact of decentralizing and how it measures against a centralized IT-structure, more thorough research must be conducted.

As mentioned in chapter 5 *Discussion* it would be interesting to conduct a larger study on processes which are all of the same size. It could also be interesting to conduct a study where the same process is developed from different organizational approaches.

If different types of processes would be compared as they were in this study, i.e. if the same methodology would be used to replicate a similar project, it would be interesting to use a more well thought out size factor. Instead of only using the manual effort as a base, some sort of complexity of automating the process could have been derived in order to make the comparison more correct. In future research when comparing different cases, it would also be interesting to include and compare more types of cost bearers, such as licensing costs for the automation software and cost of operating the robots (maintenance etc.). It could also be interesting to include more steps of the development process, such as follow-up and handover. For example, for the two cases developed from a decentralized approach there was a handover period where manuals and case reports were written. These steps could in future work be included as a factor for comparison.

Furthermore, a subject for future research would be to investigate more comprehensively surrounding **which** soft factors that are affected by RPA-related structural change and **how** they are affected. For example, as it was mentioned in the frame of reference, researching how employees and their motivation is affected by receiving more responsibility in their everyday work. How do they perceive the change in organizational structure with regards to RPA-implementation?

While on the subject of a purely qualitative study on structural change, it could be interesting to research a company which has undergone a similar change to the one investigated in this study and analyze the result on leading organs and which compromises and specific changes that had to be made in order to make the structural change work.

It would also be interesting to investigate the result of this thesis further. Now that the proposal to decentralize the RPA-related IT-structure has been made, is the proposal viable with regards to previous research about structural change?

8 Contribution to research

The results presented in this thesis are specific to the department and the company investigated in the study. Therefore, the research could provide a base for the department when deciding whether to decentralize their RPA-related IT-structure. Even though the results of the research are limited, the authors still feel that the methodology which was followed could be interesting for thesis workers or developers conducting similar studies in the future. It could also provide an insight on what could have been performed differently and what mistakes were made during the development of the cases.

References

Aguirre, S. & Rodriguez, A. (2017). Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study. Applied Computer Sciences in Engineering. Communications in Computer and Information Science, 742, pp. 65-71.

Alexius, S. & Furusten, S. (2019). Managing Hybrid Organizations: Governance, Professionalism and Regulation. Palgrave Macmillan, London, UK. http://search.ebscohost.com.ludwig.lub.lu.se/login.aspx?direct=true&site=edspub-live&scope=site&type=44&db=edspub&authtype=ip,guest&custid=s3912429&groupid=main&profile=eds&bquery=AN%2019659686 [2020-03-03]]

Annosi, M C., Brunetta, F., Magnusson, M. & Boccardelli, P. (2018). Learning and Innovation in Hybrid Organizations. Palgrave Macmillan, London, UK. https://link.springer.com/chapter/10.1007/978-3-319-62467-9_1 [2020-03-04]

Asatiani, A. & Penttinen, E. (2016). Turning robotic process automation into commercial success. Case OpusCapita. Journal of Information Technology Teaching Cases, 6(2), pp. 67-74.
https://doi.org/10.1057/jittc.2016.5 [2020-02-12]

Battilana, J. & Lee, M. (2014). Advancing Research on Hybrid Organizing - Insights from the Study of Social Enterprises. The Academy of Management Annals, 8(1).

Chandler, A. D. (1962). Strategy and Structure: Chapters in the History of American Industrial Enterprise, 68(1), pp. 158-160. Cambridge, Mass.: MIT Press.

Dobre, O. (2013). Employee motivation and organizational performance. Review of Applied Socio- Economic Research, 5(1), p. 53.
ftp://ftp.repec.org/opt/ReDIF/RePEc/rse/wpaper/R5_5_DobreOvidiuIliuta_p53_60.pdf [2020-03-02]

Egelhoff, W. G. (1988). Strategy and Structure in Multinational Corporations: A Revision of the Stopford and Wells Model. Strategic Management Journal, 9, pp. 1-14.

Engberg, S. & Sördal, T. (2019). Robotic Process Automation and its Effects on Job Characteristics. Managing Hybrid Organizations: Governance, Professionalism and Regulation. Springer International Publishing. p. 333.
http://lup.lub.lu.se/student-papers/record/8981946 [2020-03-17]

Gummesson, E. (2003). All research is interpretive!, Journal of Business & Industrial Marketing, 18, 6/7, pp. 482-492.

Lacity, M. C., & Willcocks, L. P. (2015). What Knowledge Workers Stand to Gain from Automation. Harvard Business Review.
https://hbr.org/2015/06/what-knowledge-workers-stand-to-gain-from-automation [2020-02-12]

Lacity, M. C. & Willcocks, L. P. (2016). A New Approach to Automating Services. MIT Sloan Management Review.
https://sloanreview.mit.edu/article/a-new-approach-to-automating-services/ [2020-03-02]

Lentz, S. S. (1996). Hybrid Organization Structures: A Path to Cost Savings and Customer Responsiveness. Human Resource Management, 35(4), pp. 453–469.

Lhuer, X. (2016). The next acronym you need to know about: RPA (robotic process automation). McKinsey.
https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-next-acronym-you-need-to-know-about-rpa [2020-02-07]

Lioliou, E. & Willcocks, L.P. (2019). The Study of Information Technology Outsourcing. In: Global Outsourcing Discourse. Technology, Work and Globalization. Palgrave Macmillan, Cham

Lionel, P. & Schell, T. (2018). IT Management: The art of managing IT based on a solid framework leveraging the company's political ecosystem, p. 7.

Mintzberg, H. (1979). The Structuring of Organizations. Prentice Hall, Englewood Cliffs, NJ. USA.
https://www.nrc.gov/docs/ML0907/ML090710600.pdf [2020-02-14]

Siggelkow, N. & Levinthal, D. A. (2003). Temporarily Divide to Conquer: Centralized, Decentralized, and Reintegrated Organizational Approaches to Exploration and Adaptation. Organization Science, 14(6), pp. 650–669.
https://pdfs.semanticscholar.org/abbc/37ec5f24b8cde261102c2da6a881c5638d53.pdf?_ga=2.65734958.1803208352.1588937925-1677774284.1588937925 [2020-03-02]

Stiehler, A. & Gantori, S. (2020). Longer Term Investments: Automation and robotics. Chief Investment Office GWM, UBS.
https://www.ubs.com/content/dam/WealthManagementAmericas/documents/automation-and-robotics-lti-report.pdf [2020-02-14]

Sull, D. Sull, C. & Yoder, J (2018). No One Knows Your Strategy - Not Even Your Top Leaders, MIT Sloan Management Review.
https://sloanreview.mit.edu/article/no-one-knows-your-strategy-not-even-your-top-leaders/ [2020-03-02]

van der Aalst, W. M. P. Bichler, M. & Heinzl, A. (2018). Robotic Process Automation. Business & Information Systems Engineering, 60, pp. 269–272.
https://doi.org/10.1007/s12599-018-0542-4

UiPath (2020). Intelligent Process Automation: Automation enabled by context-aware robots.
https://www.uipath.com/rpa/intelligent-process-automation [2020-02-12]

UBS (2020).
https://www.ubs.com/content/dam/WealthManagementAmericas/documents/automation-and-robotics-lti-report.pdf [2020-02-06]