UNIVERSITY OF LJUBLJANA
SCHOOL OF ECONOMICS AND BUSINESS

MASTER'S THESIS

IMPROVING ROBOTIC PROCESS AUTOMATION PERFORMANCE WITH DATA ANALYTICS

Ljubljana, March 2024 VALERIJA KONESKA

AUTHORSHIP STATEMENT

The undersigned Valerija Koneska, a student at the University of Ljubljana, School of Economics and Business, (hereafter: SEB LU), author of this written final work of studies with the title Improving robotic process automation performance with data analytics, prepared under the supervision of red. Prof. Dr. Jurij Jaklič.

D E C L A R E

1. this written final work of studies to be based on the results of my own research;

2. the printed form of this written final work of studies to be identical to its electronic form;

3. the text of this written final work of studies to be language-edited and technically in adherence with the SEB LU's Technical Guidelines for Written Works, which means that I cited and / or quoted works and opinions of other authors in this written final work of studies in accordance with the SEB LU's Technical Guidelines for Written Works;

4. to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Criminal Code of the Republic of Slovenia;

5. to be aware of the consequences a proven plagiarism charge based on the this written final work could have for my status at the SEB LU in accordance with the relevant SEB LU Rules;

6. to have obtained all the necessary permits to use the data and works of other authors which are (in written or graphical form) referred to in this written final work of studies and to have clearly marked them;

7. to have acted in accordance with ethical principles during the preparation of this written final work of studies and to have, where necessary, obtained permission of the Ethics Committee;

8. my consent to use the electronic form of this written final work of studies for the detection of content similarity with other written works, using similarity detection software that is connected with the SEB LU Study Information System;

9. to transfer to the University of Ljubljana free of charge, non-exclusively, geographically and time-wise unlimited the right of saving this written final work of studies in the electronic form, the right of its reproduction, as well as the right of making this written final work of studies available to the public on the World Wide Web via the Repository of the University of Ljubljana;

10. my consent to publication of my personal data that are included in this written final work of studies and in this declaration, when this written final work of studies is published.

11. that I have verified the authenticity of the information derived from the records using artificial intelligence tools.

Ljubljana, March 12th, 2024 Author's signature: _____

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

RPA – Robotic Process Automation

UI – User Interface

IPA – Intelligent Process Automation

ML – Machine Learning

BPM – Business Process Management

KPI – Key Performance Indicator

RCA – Root Cause Analysis

PDD – Process Definition Document

CRISP - DM – Cross-Industry Standard Proces for Data Mining

IAM – Identity and Access Management

VDI – Virtual Desktop Infrastructure

1 INTRODUCTION

One of the things that different-sized businesses from all industries have in common is the necessity to establish effective business processes that would help achieve strategic alignment and create value for the company. Data entry, extraction, and processing are often critical initial steps in various business processes; with these steps, the company should have data that is appropriately standardized, consistent, and reliable, which can further create knowledge of both the company's internal processes as well as external factors such as customers and competitors – all crucial for decision-making and efficiency enhancement. A holistic view of the overall business process, including the initial steps, reveals that many tasks are repetitive and structured, meaning that they consume valuable time and resources during their manual execution (Leshob et al., 2020). To stay competitive, businesses can introduce Robotic Process Automation (abbreviated: RPA) to automate these kinds of tasks.

RPA is a software technology that can, locally or through a virtual machine, automate predictable, repetitive tasks and operate applications following the person's steps in front of a computer screen (Alberth & Mattern, 2017). The global RPA market was valued at 2.3 billion USD in the year 2022; this value is expected to grow at a CAGR of 39.9% by the year 2030 (Grand View Research, 2022), which is understandable as a survey of 500 company executives reveals that while 83% of them use automation, 78% are likely to invest more in automation especially because of labor shortages (UiPath, 2022). When a business has its processes optimized and documented, RPA can be easier to implement compared to other technologies (Axmann & Harmoko, 2020) firstly because it interacts with the User Interface (abbreviated: UI) by following pre-set rules and business logic; therefore, the technology can follow and improve the manual execution of the process without having to make changes or switches to the software it interacts with. This also makes it easier to confirm which processes are suitable for RPA based on requirements. Secondly, smooth implementation is possible because RPA can be incorporated into Business Process Management in a bottom-up approach, meaning that it can integrate itself into the processes and is scalable according to changing needs (Capgemini Consulting, 2016).

While its advantages are favorable and RPA has a lot of potential, robots are not without their faults, the dominant problem being their likelihood for frequent failure. The downside of RPA robots relying on pre-set logic is that independently, they do not have intelligence and self-learning capabilities; therefore, if the logic is not set properly, the steps change, or the user interface changes frequently, then the robots will fail and need human intervention to resolve the failure. Ernst and Young's (2019) research states that 30-50% of initial RPA projects fail to produce the anticipated return, meaning that aside from their implementation, the individual process needs to fit within RPA, IT, and business capabilities. Like with any other software, a high robot failure percentage creates additional expenses for the development and business consulting teams. The development team needs to allocate

resources to bug-fixing and error handling (Hamill & Goseva-Popstojanova, 2017), while the business consulting team needs to improve the process that the robot emulates if the errors persist, as selecting the right processes to automate is one method to maximize ROI (Deloitte & Blue Prism, 2023).

Data Analytics is one decisive way to monitor RPA robots, as using different techniques to create visualization can help identify patterns, anomalies, and deviations from the standard (Abu Sulayman & Ouda, 2018). The operational metrics of the robots, such as their productivity, average handling time, and process throughput, can be some of the key metrics for successful analysis (UiPath, 2020). After this point, the analytics combining automation through BI&A visualization tools and manual human intervention for data interpretation are labeled as semi-automated data analytics. Furthermore, RPA activities have recently been extended with the introduction of Intelligent Process Automation (abbreviated: IPA) through machine learning conjunction. Machine learning (abbreviated: ML) allows for the robots to gradually redirect from performing only repetitive, structured tasks successfully to more complex, knowledge-intensive, and value-adding tasks (Ivančić et al., 2019). Because of its complementary link, and its possibility for failure prevention through monitoring for deviations and analyzing root causes**,** ML can be used as another valuable approach, which is labeled as automated data analytics. While implementing RPA yields considerable benefits, addressing its primary failure disadvantage can prove to be challenging. Additionally, understanding how to exploit data analytics and ML as appropriate preventive measures and performance improvement tools is paramount to fixing the disadvantage and achieving improved process accuracy.

The purpose of this master's thesis is to provide companies with guidelines for leveraging data analytics to identify and minimize the causes of RPA process failure, thereby enhancing RPA performance. First, this thesis aims to examine the most common reasons for robot failures within the company context through documentation analysis and to identify the benefits of process monitoring. Second, it is to document data analytics implementation and assess the potential of using it for RPA's performance improvement by utilizing two different approaches for analyzing the data: a semi-automated approach and an automated ML model approach. By conducting a comparative study, an assessment is made about which approach is more effective in identifying potential anomalies and errors in RPA processes. Third, the objective is to use the study results to provide insights into robot improvement, specifically for preventing or mitigating the same failures in the future.

The thesis intends to answer the following research questions:

1. How can data analytics be used to forecast RPA failures, and what are the benefits and limitations of implementing it in the company's RPA processes?
2. Which approach, semi-automated or machine learning model, is more effective for detecting RPA process anomalies?

3. How can the approach output be leveraged for long-term RPA process optimization and overall improved performance?

The thesis is done in collaboration with a department of dmTECH GmbH. Specific information concerning company structure and RPA processes in the production environment is omitted from the thesis for confidentiality reasons.

The thesis consists of a theoretical and an empirical part. It employs different chapters gathered as primary data and secondary data. The theoretical part uses distinct research resources, referring to them in the process of straightening the arguments and defining the used concepts. Information in the literature review part is gathered from various studies and meta-analyses; those include academic journals, reports, books, technology-related publications, and company frameworks. With the aim to answer the research questions, this part provides logical cases about RPA automation, including its benefits and disadvantages. It further assesses data analytics and machine learning involvement in RPA, specifically for process improvement through error prevention. It focuses on answering how each approach can be implemented and on evaluating their application's results.

The empirical part is realized by collecting and analyzing primary data. Interviews with RPA experts in the company were conducted regarding the gravity of continuously monitoring parameters to ensure process improvement. Interviewees shared insights into the data types that produced the most valuable results in data analytics for automation monitoring and identified the factors that predominantly cause failures based on their experience. Within the realm of RPA, a data extraction robot was developed to collect historical data from automation management tools and test case robots to produce performance data, necessary for compiling training and testing datasets. Python and PowerBI were used for descriptive analysis to create interactive data dashboards and reports that facilitate data exploration and metrics tracking. To assess the preferred approach, semi-automated or automated, both were compared based on the ease of implementation, the need for human interpretation, the likelihood of accurate results, stability, robustness, real-time prediction, and flexibility. The comparative analysis provides insights into the research questions concerning each approach's limitations, benefits, and practical usage in reducing RPA robot failures. Once a superior approach was chosen, the empirical study determines how to leverage its outputs long-term and what the savings are in different aspects from the company’s initial situation.

2 THEORETICAL FRAMEWORK

2.1 Robotic Process Automation

2.1.1 Definition

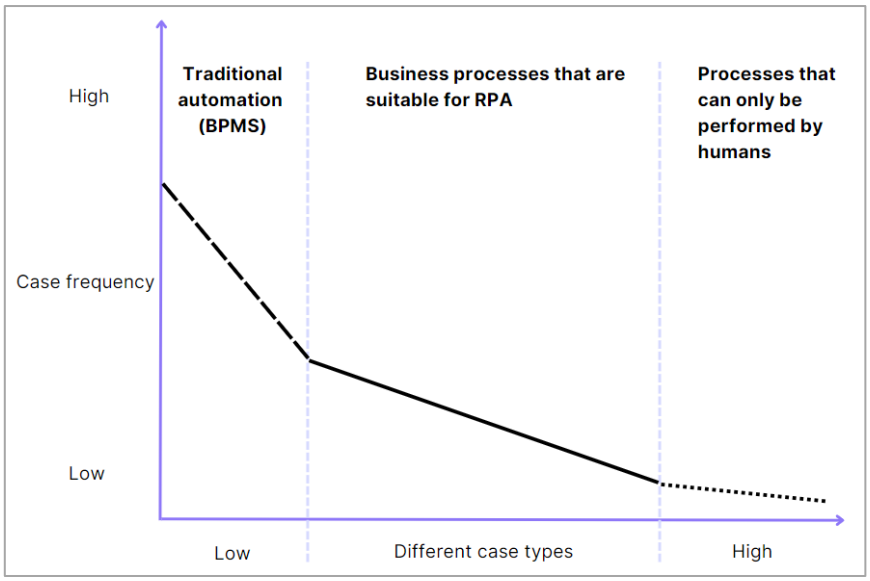
Leshob et al. (2020) explain that RPA is a technology that uses algorithms and software robots to perform tasks across multiple business applications through a graphic user interface without having to alter the existing infrastructure and systems. Automation is primarily based on defined business rules, often expressed in an if-then statement (Lacity & Willcocks, 2016). Activities like entering data, acquiring, and processing data from online sources, sending emails with attachments, uploading files, and calculating Excel ranges are just some that allow software robots to execute tasks successfully by following the process step-by-step as a human would do it. This means that the robots are configured mainly to process transactions, manipulate data, and communicate with other software systems (Leshob et al., 2018).

For a company to capture the essence of RPA, an introduction toward enabling productive automation, it is essential to highlight the type of tasks RPA automates, the type of data it uses, and the rules it is based on. The mentioned activity types and systematic literature research conducted by Ivančić et al. (2019) show that in a business sense, RPA best automates "high volume, repetitive, monotonous, well-structured and standardized tasks, where there is no need for subjective judgment, creativity or interpretation skills". Conducted case studies show that companies that use full-time equivalent (FTE) savings from automation have reported, among other percentages, three of the highest as follows: 50% of responders redeployed employees within the work unit, 49% redeployed employees to another work unit in the company and 43% took on more work with the same number of employees. This means that automation takes over tedious and repetitive tasks, allowing employees to focus on tasks such as solving problems, thinking creatively, and building relationships (Lacity & Willcocks, 2018). By minimizing the burden of these tasks for employees, especially noting their repetitive nature, they can easily have more time available to focus on higher-value, higher-satisfaction tasks within their expertise (Accenture, 2016). Huff (2021) references a survey conducted by Forbes where 302 executives shared their RPA implementation experience. The results show that 92% specified an increased employee satisfaction, with 52% of them specifying a higher satisfaction by 15%.

The Pareto distribution, shown in Figure 1, effectively explains the usefulness of RPA automation by depicting a more significant portion of tasks that can be automated. It illustrates the correlation between RPA and traditional automation, where the y-axis shows the case frequency and the x-axis the different case types. According to the distribution, 80% of the cases can be explained by 20% of the case types, indicating that the relevance of traditional automation is financially viable for frequent tasks and that the most feasible RPA

automation is for low to medium complexity, repetitive tasks that are being executed manually. It becomes evident that RPA is immensely beneficial in automating tasks that are time-consuming but do not happen often enough for traditional automation to be appropriate (Flechsig et al., 2019).

Figure 1: Positioning of BPM and RPA



Source: Flechsig et al. (2019).

There are two types of software robots: attended and unattended. As the names suggest, a human controller on their local machine triggers and tracks the attended robot. There are adequate reasons for using attended robots, particularly in case the robot cannot directly access a system or tool, which can be due to not being authorized for login credentials saved as robot assets. An additional reason is if the employee must perform complex tasks concurrent with the robot's execution. On the other side, unattended robots can have a scheduled trigger or be triggered remotely and run without human interaction on a dedicated machine (UiPath, 2023). The choice of robot type depends on the automated process, client requirements, and the logging, scheduling, and tracking system necessary for the development team. While attended robots can be considered less risky since their execution is supervised in real-time by the employee, unattended robots can also have triggers, timely notifications, and logs, allowing for quick actions when appropriate.

2.1.2 Benefits and Challenges of Robotic Process Automation

Understanding the benefits and challenges of a certain technology and how they influence the existing workflows can guarantee its long-term potential in an organization. By evaluating the two opposites, management can make informed decisions throughout the whole lifecycle – from introducing and controlling the technology to transforming it, if

necessary, to successfully serve its purpose and improve the performance/outcome of those workflows.

RPA assures advantageous data consistency and reliability through human error elimination and data validation before processing, improving process efficiency and effectiveness. From the depicted research about RPA implementation logic criteria, where the time to complete a task was analyzed based on the number of resources, the results suggest maximizing productivity where the teams need to prioritize automating tasks that have common patterns and require higher manual labor (El-Gharib & Amyot, 2022). Client-published studies about RPA implementation suggest that, with fitting governance, it results in FTE cost savings, better service quality, and service delivery speed (Lacity & Willcocks, 2018), which indicates the potential to add additional value due to saved time, improved monitoring, and agility. For example, in a communication process between systems, the automation business logic can find root cause issues and instantly give responses to the client about the reasons behind them.

Before suggesting a roadmap of important implementation steps into the company's business processes, that provide a holistic view of RPA within an organization, it is essential to emphasize the primary benefits associated with its implementation, which can be assessed from a **technological** and **cost-saving** perspective.

From a **technological** perspective, the robot mimics the employee's steps on the presentation layer and within the same interface as the employee, which eliminates the need for its integration, making RPA a non-invasive technology for accessing other organizational systems (Syed et al., 2020). This allows the robot to work 24/7, at a faster rate, across different systems/interfaces, and within different industries/departments, confirming the benefits of not interrupting the existing systems. User-friendliness is critical because of its layout, reusable modules, drag-and-drop activities, and properties, as responsible employees do not need advanced coding abilities. The need for IT consultation is also minimized because complicated application adjustments and configurations are limited, highlighting the low technological barrier to entry.

From a **cost-saving** perspective, RPA is a less expensive automation option to use compared to alternatives, taking into consideration its upfront investment costs (internal resources, robot provider, machine, service provider), license fees, governing fees, and scalability where its increase corresponds to an insignificant increase in costs (Alberth & Mattern, 2017). RPA requires a low-cost initial investment, with findings disclosing that the cost of this solution can be one-fifth of the price of a full-time employee executing the same task manually (Hindel et al., 2020). A database to store transactional data is not required, reducing any ongoing costs of managing a dedicated database (Ivančić et al., 2019). The quantitative savings further depict reduced costs due to defective processing, especially with optimized RPA development and shorter service level agreements with clients, allowing for improved retention and new revenue streams (Alberth & Mattern, 2017). Aside from the direct or tangible savings due to its implementation, many ancillary ones serve as a plus to

other benefits and carry the same value depending on the use case. For example, with the RPA benefit of work consistency, increased precision is expected, reducing processing time, labor costs, and cost of impact (Moreira et al., 2023).
However, it is imperative to mention that even though savings stand on the positive end compared to other automation, RPA costs should not be underestimated. This and other factors make it clear that RPA is not immune to challenges. To validate the cost-effectiveness of implementing RPA, proof of concept should be built to demonstrate the value for the organization in terms of development, integration, licensing, maintenance, and infrastructure because costs can outweigh savings if a non-suitable process is automated (Hindel et al., 2020). The percentage of savings generated through RPA is substantial, necessitating a comparison of the future savings against the initial development expenses (Koch & Fedtke, 2020). Existing automation maturity models can help the organization plan for technological evolution. Syed et al. (2020) organizational characteristics automation state that maturity means the company has the required resource experience, equipment, people, and funding to support RPA, making it easier to fit with the dynamic factors for optimized automation.

Other challenges the company should consider in the landscape of adopting automation fall into the **social/implementation** and **technological** categories (Moreira et al., 2023).

The **social/implementation** category focuses on the role of sound communication, documentation, and managing of the stakeholders in trusting the technology prospects. Management needs to establish governance, continuously communicate, and potentially educate employees on usage aspects within the company and organize the RPA development process by following standardized organizational methods and procedures (e.g., Documentation plan for comprehensively capturing activities and processes). This type of transparency can help improve a larger organizational goal, trust in the automation purpose, and how it relates to job security. Expert interviews confirm that employees can oppose automation due to fear of it causing their position or mistrust that the selected part of their position can be effectively automated (Hindel et al., 2020), further depicting the critical role of leadership. Communication is also vital within the RPA team because it facilitates knowledge sharing, informed decision-making, and swifter problem-solving and ensures a clear understanding of the objectives and goals of the whole team. While discussing the need to keep the teams involved and improve real-time transparency, the in-house RPA consultant stated, “The current challenge is that we lead the communication usually through email or teams in a personal message to explain the issue. It often happens that a colleague is informed about something additional that the rest of the team missed. Keeping up with the documentation is also important, where data tables and opened process tickets need to be updated regularly.”

When combined with the pool of other documentation and useful historical data, the **technological** category shows how crucial it is to assess which processes are suitable for automatization. It is more about how structured and standardized the processes to be automated are, fitting RPA with the complexity of the environment due to its limited

cognitive capabilities, the technological readiness of the organization, and the implementation methodologies (Moreira et al., 2023). Different examples fall under each mentioned point, such as more extended development than planned and increased ongoing costs due to a lack of technological readiness/knowledge. If RPA is already implemented, there is a risk that without process documentation, employees can eventually forget how the process was executed manually and, with that, any necessary access rights, or expertise. Documentation of the AS-IS process scenario, as well as the BPMN format of the steps that the robot will take, are therefore crucial for successful automatization. Constructive documentation offers a more extensive range of valuable advantages; it aids the development team in error handling and provides guidelines for when any intelligent technology is integrated with RPA.

2.1.3 Robot Failures Challenge

As the introduction highlights, the likelihood of robot failures poses a key challenge. Identifying the causes of these failures can be difficult due to various possible underlying factors. Therefore, recognizing the most common factors is crucial to ensure robust processes and maintain trust among current users.

In the **social/implementation** disadvantages category, there are additional important exceptions to consider that can lead to failure. This is why change management being deployed more extensively is essential for stable robot execution. Change management is the procedure of transitioning to a future state in different company aspects, from technologies to structures. A useful model for creating a change management plan is Kotter's 8-step model, which addresses communicating the vision properly and dealing with the company's structure that would bring about the incentive to change, among other things (DeDavis, 2022). It is more difficult to manage robot failures if change management in the company does not incorporate RPA, as any modification in the automated applications requires verification to determine potential impacts. The deviations from the ideal path should be considered in the process planning phase because only thinking about the successful path and not about an inevitable exception and how to handle it heightens the risk of failure. During the interview with T. Endt, a Pre-Sales technical consultant in UiPath who supports both the business and technical aspects of RPA process insights, he stated, "Tools like task mining, where the process owner records many times the execution of the process, could unavoidably portray the exceptions from the ideal path which can happen often."

Scalability involves numerous change management activities, transitioning RPA processes beyond implementation. With an increasing number of processes, adoption, and scaling become a continuous cycle of RPA support processes (Herm et al., 2022). Several factors affected by scalability can impact the stability of RPA and potentially lead to process failures. Firstly, it is vital to govern if the company adopts a fixed-term or continuous automation approach, which determines if resource capacity increases with every new

process, crucial for development, support, and other tasks. Secondly, whether on-premises or cloud-based, the chosen deployment model must align with the desired scaling capacity (Asatiani et al., 2022). Lastly, regarding infrastructure, it needs to remain robust and adaptable. If the infrastructure changes are not adequately managed, it can introduce vulnerabilities or incompatibilities that directly contribute to RPA failures. This is why early IT involvement needs to provide support for complying with IT security and configured infrastructure (Lacity & Willcocks, 2016).

From the **technological** disadvantages category of RPA, which shows that suitable process implementation is an ongoing initiative aside from the costs recognized when a non-suitable process is automated, automating the wrong process also increases inefficiency and failure speed (Santos et al., 2019). It is evident that **process/data supervision** and **security measures** play a role in preventing robot failure. **Process/data supervision** shows a risk of robots working with out-of-date business rules or older test data that is not updated to the new system interface, leading to faulty outputs. RPA cannot handle deviations or adapt to changes without human intervention, or the utilization of data analytics technology equipped with sufficient monitoring instructions, which can inevitably lead to inaccuracies in its performance over time (Syed et al., 2020).

Since failures can mostly occur due to changes after the automation has been finished, not all potential causes can be eliminated in the design phase. Therefore, recognizing the importance of finding efficient ways of error handling to reduce failures in the productive environment is crucial for fully leveraging the benefits of RPA automation. Different examples of actions assist in achieving this purpose, such as proactive monitoring and supervision so that the process stays in focus or taking security measures for the process to fit within certain organizational regulations. Similarly, to minimize the time spent on eventually updating the process activities, a leaner and more dynamic way of designing the bot with reusable modules, arguments and by following coding guidelines (e.g., Activity naming, frameworks, error handling rules, versioning, log tracking) is vital. T.Endt and the in-house RPA consultant both concurred that “running automated tests on the robots regularly within the development setting can verify the automation’s functionality or detect issues well before they arise in production. Alerts regarding the test cases with a faulty status indicate necessary adjustments.” This can highlight the interconnection of different issues, such as the team's capacity to update test data and respond to alerts with a rework and the dedicated robot infrastructure for consistent testing without tying up machines required for sporadic tests during process development. Before starting constant automated testing on all RPA processes, all interconnections should be considered, therefore recognizing failures.

Based on regulatory compliance, **security measures** are reviewed because of the impact robot implementation makes on the structure of the company’s business model (Moreira et al., 2023). For departments that have sensitive data, like finance or HR, stricter regulations should be considered for data protection, robot user access controls, and permissions.

Pertaining to the permission rights, process failure can happen due to a lack of rights for the technical user. In a discussion about common causes for RPA failure, an in-house RPA expert developer stated, “While issues like system updates, maintenance, network, or authorization problems that make the systems unavailable for the robot can be unpredictable, user permission setup is an area we can control better. This can be done by having a list of users and information when something may change, such as permission expiration.” Workshops in different companies have concluded the point that with IT support, the robot or technical user needs to have defined permission rights, the same as the process owner (Herm et al., 2022).

2.1.4 Robotic Process Automation Roadmap

Building upon the foundation of thorough documentation and process suitability assessment, this subsection leverages Business Process Management (abbreviated: BPM) as a strategic framework. BPM is a structured approach that serves to understand and improve the company’s business processes. More precisely, it is defined as a body of methods that help discover, analyze, redesign, execute, and monitor business processes (Dumas et al., 2013). These BPM lifecycle steps also change the current state toward an improved TO-BE model. Its foundation brings IT and business specialists together through a common language. It manages activities and decisions that add value to the company and its clients and helps align the processes with the strategic goals and performance objectives. Its definition aids in clarifying BPM as a holistic approach striving for an end-to-end improvement of the processes in possible efficiency, functioning, processing, and other aspects. Due to different systems carrying out functions that assist in the holistic approach, one of the main BPM components is process automation, where an IT system (RPA in this instance) is included in the process redesign serving as a tool to automate activities fully or provide automation support after modeling is used to understand the process relations or resources (Dumas et al., 2013).

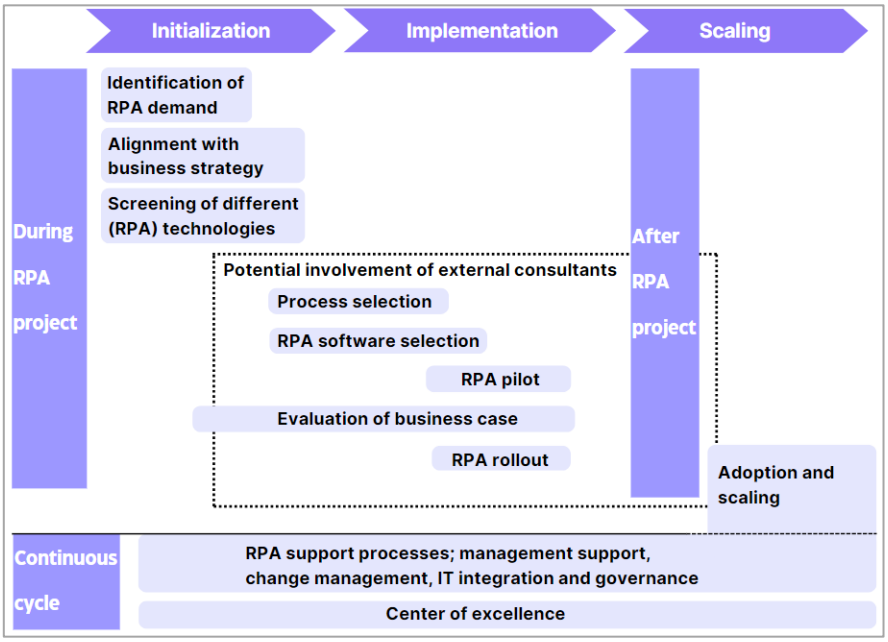
To fully convey the benefits of the implemented RPA tools and to comprehend how improving the BPM approach impacts them, it is essential to understand the relationship between RPA and BPM. It is also important to note that while BPM is used to automate an entire business process, initial RPA robots automate sub-processes or activities/tasks within a process (Wanner et al., 2019). Flechsig et al. (2019) inspect ways to explain the relationship by first focusing on the distinguishing characteristics between the two technology concepts, listed in the following criterion: area of application, procedure of automation, method of integration, implementation personnel, implementation effort, introduction phase, and dissemination.

Taking as an example the method of integration criterion, BPM uses application programming interfaces to access third-party systems and integration happens in a “top-down” movement because processes are standardized and because BPM operates on a

macro-level, transforming practices in that direction. While RPA is task-based on existing interfaces and systems, making its integration in a "bottom-up" movement because its processes do not change the system logic. Finally, while there are similarities to consider, it is also clear that the technology concepts of RPA and BPM are independent and distinguishable. However, simultaneous usage of RPA as a part of BPM can bring the company to realize the full RPA potential (Flechsig et al., 2019). Since the RPA automation steps follow the documented process, their redundancies transfer onto the newly designed robot process flows if BPM is not combined to help standardize and optimize them (Flechsig et al., 2019). This means that with strategic deployment and synergizing, optimized or yield immense benefits that bring the company, which has existing functional BPM systems, forward. This can proclaim the RPA tool as a part of the entire BPM umbrella, confirming its complementor role and not one striving to replace the approach.

By leveraging the BPM and RPA combination in a supportive environment, companies can successfully coordinate employees and robots, therefore offering initiatives to improve business processes and, more importantly, create a solid platform that supports end-to-end digital automation. To achieve successful end-to-end automation, RPA plays a role in steps of an existing process that are rule-based, increasing process accuracy with the processing of big data, and repetitive tasks. Meanwhile, some of the activities BPM accomplishes are orchestrating employees to follow business rules, warranting ordered interaction with system integrations, and encompassing the process workflow design that will undoubtedly positively influence the monitoring and analytics of RPA processes. Some different companywide models and concepts ensure proper relation to the RPA benefits and its seamless integration with BPM and business rules, one of them is displayed in Figure 2.

Figure 2: Framework for RPA Implementation



Source: Herm et al. (2022).

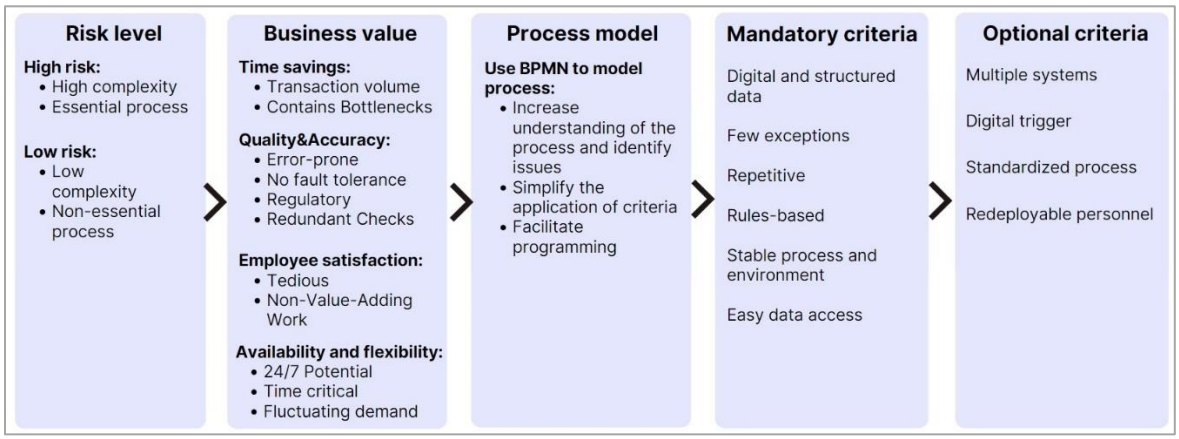
In a company's context, the models provide valuable guidelines and support in designing a roadmap with essential steps. Even though Figure 2 displays RPA implementation steps, to focus on a maturity level where the RPA automation readiness is already established, continuous and improved-suitability steps from the framework example are highlighted focusing on ultimately keeping a successful synergy between RPA and BPM.

As with any changing technology, each of the implementation steps can eventually be re-evaluated. This entails the procedure items of assessing risks, project effort, management plan, quality requirements, and timeline, as well as the method or success factor items of project planning warrant re-evaluation (Krakau et al., 2021). For example, transforming the process selection criteria can initiate a chain reaction affecting subsequent steps. Further, considering the RPA software selection criteria, if the company uses one of the leading providers UiPath, Automation Anywhere, or Blue Prism from more than 50 other substitutes (Alberth & Mattern, 2017) belonging to challengers, niche players, or visionaries groups, conducts an analysis of the platforms and recognizes that another provider offers capabilities that are more useful for its structure and process stages focus this is also an aspect that can change. For example, a platform overview of these three software can elaborate that Blue Prism has a vertical market strategy, flexible tools, and secure systems. Moreover, UiPath is recognized as the leader of Gartner's Magic Quadrant's fourth consecutive year (Gartner, 2022), and Automation Anywhere includes process discovery, analytics, and marketplace integration tools (De Moraes et al., 2022). As noted, an analysis can clarify if in fact the additional benefits that the new software offers exceed the costs to make the transfer, causing the need to review all indicators that initially served as a basis for management approval.

Fitting with the BPM initialization phase, when a company has already defined the suitability and adapted RPA, the RPA **initialization** steps can be assessed for individual processes since they differ depending on their type or the technology expertise required and can therefore be regarded in a continuous cycle (Herm et al., 2022). In identifying RPA demand criteria, the dedicated team needs to focus on two main activities: identifying automation opportunities and discovering if a manual process should be automated. (Herm et al., 2022). The first activity increases awareness for automation opportunities; by analyzing the awareness of the value-adding technology with each process iteration, the team can gather which actions give the best results in helping the company understand and potentially trust RPA. It can also motivate interest in automation in different departments, as employees become aware of its capabilities through workshops, training, or presentations of successful processes.

The second activity involves strategizing if the process should be automated, based on criteria that reflect the type of tasks most suitable for RPA, some of which have been previously outlined in subsection 2.1.2 on benefits and challenges. Many different frameworks and techniques can help summarize the suitability of RPA tasks; one of them is portrayed in Figure 3.

Figure 3: RPA Business Process Suitability Framework



Source: Agaton & Swedberg (2018).

Each of the mentioned criteria is iterative and requires an ongoing evaluation. The technology has regular updates that can advance its capabilities, and feedback gathered from stakeholders can also identify new insights. Since a complex company displays a variety of characteristics that influence its automation potential, it is challenging to implement RPA with a one-size-fits-all philosophy (Wanner et al., 2019). A thorough understanding of the business processes confirms that RPA-applicable process selection requires a situational analysis and adaptation (Lacity & Willcocks, 2016). Santos et al. (2019) explain that the outcome of the process lies in following the criteria mentioned, ranging from having clear rules and processing voluminous transactions to minimizing changes in the user interface relative to the underlying data structures so that the robot is not reconfigured as often. They further specify that once a process that meets most of the criteria is identified, a tactical evaluation on implementing the automation should be conducted, aligning with the implementation roadmap.

Within the **continuous cycle**, the RPA support processes are principal for overseeing other steps, not reliant on the RPA's business model being conservative, strategic, or achieving efficiency improvements (Moreira et al., 2023). The management support and governance processes serve as guidelines that preserve the awareness of RPA capabilities and assure RPA processes are continuously optimized and consistent in business strategy alignment. Considering the defined RPA characteristics and current process criteria, particular focus should be added to three key criteria that play a crucial role during the transition from the documentation to the development phase of the end-to-end process. These are time savings (measured in effort days), cost savings (measured in effort/saving ratio), and documentation criteria. Proper documentation, in line with the documentation methods, delivers a deeper understanding of the processes and guides the adaptation of best practices for process optimization (Koch & Fedtke, 2020).

2.2 Data Analytics in Robotic Process Automation Monitoring

2.2.1 Role of Data Analytics

RPA robot execution provides data with each process step, which can be classified into datasets useful for recognizing business goals, procedures, and environments. Recommendations based on historical data and metadata can also be collected if the robot is incorporated with suitable applications that track it. Data analytics is a broader field that combines computer science, artificial intelligence, machine learning, statistics, mathematics, and business domains. It is defined as the process of sorting raw data, enabling the creation of methods that help to evaluate historical trends and predict future ones (Cuesta & Kumar, 2016, p.6). The value of its defining characteristics is evident from the steps of collecting, cleaning, preprocessing, and analyzing the data available, to its interpretation and visualization.

After the company identifies traits to become more data-driven from being process-driven, the benefits of these insights come to light during the final step of the analysis process, where the model output results are presented using visualization tools. The tools serve as an overview for asking questions, providing explanations, and testing hypotheses based on logical and analytical methods (Cuesta & Kumar, 2016, p.6). Different questions can be answered depending on the analytics conducted of the four key types, each serving a different purpose in providing control over the information. Descriptive analytics considers the historical data to answer the question “What happened?” diagnostic analytics uses methods to give root-cause details on “Why it happened?” predictive analytics uses patterns and data correlations to suggest “What will happen?” and, additionally, to oversee future outcomes, prescriptive analytics also oversees the action that should be taken, answering “What should we do next?” (Sharda et al., 2018). Descriptive data analytics essentially uses forms and structures to recognize previously unknown patterns, clusters, and associations within datasets. This approach helps to interpret the data in a structured manner and generate valuable information used to communicate and gain a deeper understanding of complex data relationships (Sheikh, 2013).

By incorporating semi-automated and automated data analytics as explained in the introduction, companies can gain a sound understanding of the company's workflows and processes. This enables them to accurately spot opportunities for improvement, thereby adding notable value to the business. RPA offers significant automation efficiencies, and since it is based on BPM workflows, the efficiencies are only amplified when data analytics is applied to different phases of the robot’s implementation to measure, support, and align the automation with strategic outcomes in mind. In addition to RPA’s advantage of automating data entry and facilitating the migration between enterprise applications, it also supports aggregation and monitoring of data sources. This preparation is crucial for data analytics, improving analytics capabilities, and enabling the use of machine learning.

In the field of decision-making and problem-solving, analytics and decision automation are apprehended as potent tools. Davenport (2009) states that companies build analytical capabilities by strategically and tactically embracing analytics. He adds, "Analytics are even more effective when they have been embedded in automated systems, which can make many decisions virtually in real-time". Implementing data analytics in the RPA phases offers several other benefits, including increased operational visibility, anomaly detection through pattern identification, improved governance to meet business requirements, and increased performance (UiPath, 2020).

During the robot's construction phase, aside from process understanding and validation, model analytics are crucial in refining the parameters that control the tasks's execution. In the production phase, gathered metrics become essential for performance monitoring and real-time operational efficiency regulation. The efficiency of a process is determined by its performance in relation to predefined indicators, some of the performance indicators that can be visualized include success rate across the automation, time to execute a task, or cost per task (Chakraborti et al., 2020). Finally, metrics from the monitoring phase of the implemented robots aid in parameter refinement which is a basis for addressing potential shortcomings (Quille et al., 2023). The provided RPA benefits and general usefulness of analytics integration made it simple to recognize the logic of the statement that "We need to think less about human-machine interfaces and more about the design of human-machine teamwork, to take advantage of the synergies afforded by the combination of people plus tools." (Norman, 2017).

2.2.2 Benefits and Challenges of Machine Learning Inclusion

ML optimizes performance criteria by utilizing example data or past experience. A model is characterized by its parameters, and the learning process involves optimizing the model's parameters through training data or experience, so that it may make predictions, gain knowledge, or both (Alpaydin, 2014, p. 4). All classification or prediction models aim to predict the value of the target variable whose outcome is unknown. The ML model can approach and quickly analyze large datasets, even specifying what elements produce different kinds of outcomes, depending on the company's goals. Cuesta and Kumar (2016, p.7) clarify that depending on how they are training there are three groups of algorithms: supervised learning, unsupervised learning, and reinforcement learning.

Having looked at the leading RPA software platforms, along with members of the other segments, it is evident that almost all facilitate the integration of ML applications, such as Python and its libraries, for the extension of the rule-based nature of RPA and improvement of the configuration, monitoring, and security capabilities (De Moraes et al., 2022). Expanding upon this foundation, IPA is defined as combining RPA with advanced technologies beyond task automation. Incorporating cognitive capabilities from ML and artificial intelligence marks the transition toward IPA. The advancements enable the system

to adapt to new data, recognize patterns, and learn from outcomes, enhancing the automation scope and overcoming the obstacles faced by RPA alone (Siderska et al., 2023). The integration is further facilitated by the ability to work with third-party cognitive technologies, pre-trained automation, and open-source ML libraries that can be incorporated as a plug-in component, making the integration process less demanding. Moreover, when ML is customized for a company's particular needs, it typically requires the aggregation of data from different systems and business processes to create a training model, which is one of the factors that make the necessary effort noticeable.

Key features for recognizing the IPA scope are data availability, type of data that can be processed, input-output relationships, exception rate, and decision-making input required from a human or technology that has cognitive abilities (Quille et al., 2023). Analysis and RPA estimation involve measuring insights, defining cost-benefit analysis, and Key Performance Indicators (abbreviated: KPIs), which are measurable metrics dedicated to showcasing the company's effectiveness in reaching its key objectives. Data quality can be enhanced through accurate data annotations during the preprocessing phase and the key features' accurate selection and weighting. Defining a suitable solution architecture acknowledges the importance of optimizing databases and data repositories and that the integrated abilities match with the company's scheme (Lievano-Martínez et al., 2022). The implementation journey to reaching desirable outcomes, depending on the context, can be realized more easily by using the guidelines generated by Lacity and Willcocks's (2021) research for intelligent automation in early adapters. There are 39 action principles worth analyzing distributed across nine sections: strategy, sourcing, program management, process selection, tool selection, shareholder buy-in, design-build, and test, run, and maturity.

This subsection focuses on the benefits and challenges associated with the management of IPA for improving RPA process execution, specifically through the integration of ML. The goal is to understand ML's impact better on improving the robot's decision logic, engines, or assurance across company boundaries to solve applicable RPA limitations.

Most benefits closely resemble those offered by RPA, as IPA is built upon the foundational technology of RPA. The integration of the technologies also leverages ML benefits, enabling better control, management, and improvement of business processes over time (Feio & Santos, 2022). By combining these technologies, processes can learn from outputs, reduce operational risks, create groups, and identify patterns. Technological challenges can be minimized by creating an extensive operational plan, choosing the right tools, and documenting processes that can be realistically automated, among others. Some of the challenges are cost-related, as the technology requires data preparation, feature engineering, and potential changes to business processes and data handling. The increased costs associated with IPA, being higher, indicate that a greater ROI or reasoning is necessary to justify the business case, compared to RPA (Chakraborti et al., 2020).

Siderska et al. (2023) elaborate that the promise of real-time monitoring and performance analytics with IPA additionally serves to improve the RPA processes. The data-driven insights lead to more accurate decision-making, improving the performance of RPA processes, minimizing the existing shortcomings, and improving customer relationships. The unpredictability of the outcome is also fundamental; RPA business logic can be easily tested, but the company cannot predict the user's interaction with the cognitive learning-based automation tool with certainty. This helps to clarify that delivering business value from the ML integration relies strongly on management practices that serve to match capabilities across tools, platforms, and attributes, scaling the technical effort required as well (Lacity & Willcocks, 2018). The company's process categorization and maintenance aspects are explored for an overview suitable for management needs. The inclusion of ML offers a multitude of advantages, encompassing triple-win benefits in increasing enterprise/shareholder, customer, and employee value. Each benefit constitutes significant subfactors, bringing forth company success (Lacity & Willcocks, 2018). Ng et al. (2021) research elaborates on other noteworthy benefits, such that IPA enables the enhancement of process transparency where there is better visibility of the robot's performance.

The challenges centered around management preparedness to achieve anticipated business goals or lead the integration process are also discussed as true for introducing any new technology in the company, focusing on appropriately assessing the risks, and fitting the capabilities with existing business workflows. This requires improving the existing roadmap and creating an adaptable strategy that ensures new capabilities will be useful in the company's processes (Siderska et al., 2023). Strong change management support for the new ML technology also belongs to the strategy, including system infrastructure, IT support, and employee training. Unpredictable changes to business activities or process complexity require context awareness, as mentioned in subsection 2.1.3. Along with reengineering legacy systems and preparing a comprehensive data governance framework, they play a complex but vital role in utilizing intelligent automation better. The reengineering confirms the consistency and quality of activity logs, event triggers, and data within business processes. At the same time, the data governance framework reduces adoption risks and improves the analytical performance of intelligent automation (Ng et al., 2021).

2.2.3 Applications and Requirements

The adoption of innovations, such as data analytics, does not imply their appropriate usage. Their incorporation can fail if the importance is not acknowledged as useful by the department and company as a whole (Hazen et al., 2012). As this thesis places emphasis on the RPA robot failure key factor, which will be analyzed in greater detail specifically for dmTECH during the company's case study, it is vital to explore the potential involvement of semi-automated and automated data analytics in mitigating this factor and to ascertain their application across the company.

The analysis of RPA automation and its associated challenges made leading factors for robot failure evident. One of the factors is the robot's reliance on following out-of-date business rules in an environment that supports business logic changes. A way to reduce such failures is to apply appropriate control mechanisms, new monitoring approaches for the health of the bots, and proactive adaptation to changes in business rules (Syed et al., 2020). Accenture (2016) emphasizes that scaling automation efforts across the company with several diverse processes can amplify the failure risk. To address this, a formal structure and long-term roadmap, potentially with technological integrations and employee task organization in mind, are vital for failure mitigation. Adequate management support is also critical for identifying RPA-suitable tasks from a technical feasibility and business value perspective. Notably, in line with other research, Feio and Santos's (2022) framework supports the importance of monitoring. It states that ongoing governance, applications, and workflow platforms can help monitor key indicators and collect robot data to digitalize processes. This enables the identification of IPA opportunities and the visualization of implementation results that interest the company, ensuring long-term functionality.

The monitoring data outputs make recognizing the process improvement metric easier, as they provide enough outputs to analyze the reasons behind RPA failures. These metrics help with data analytics as they represent the root causes and assist the companies in detecting a deviation from a typical performance behavior or changes in the environment that might lead to failure, leaving the company to prioritize tasks and updates based on the potential failure impact. Root cause analysis (abbreviated: RCA) is a class of techniques used for identifying the faults that can potentially lead to system failure (Lokrantz et al., 2018). During the preprocessing phase, if the dataset is not already fully prepared, it is necessary to transform it for root cause analysis, as it requires binary or categorical attributes (Pohlmeyer et al., 2022). A rule-based RCA uses a predefined set of rules, often derived from expert knowledge or historical analysis. Hanemann (2006) explains the process as searching for rules that are then applied to the data to identify the root causes. Using specific scenarios, he further explains that the rule-based reasoning component contains a set of rules designed to match service events to events on the resource level. If this possibility has not been considered previously and no appropriate rule exists, there is no rule match, and the problem would transfer for further analysis.

By incorporating the ML approach, automated data analytics explores and solves the failure factor with its potential prediction capabilities. It concludes the intelligent automation involvement with exception handling, for instance, using fault prediction algorithms or intelligent fault management automation, so that correct environment information is obtained. Accordingly, this prompts operational efficiency, error reduction, reduced invalid decisions, and improved process quality over time. (Ng et al., 2021). Using the data that RPA processes can provide precise accuracy levels, which increase as the learning model improves. The process enables the company to control upcoming events progressively, with the process improvement metric in mind (Lievano-Martínez et al., 2022).

However, to enhance the accuracy and handle the inevitable occurrence of unexpected data analytics results, special attention should be given to the highlighted aspects of data quality, model selection, and ongoing monitoring, as exemplified within the context of RPA. Examples of unexpected outcomes can come from overfitting the model and producing poor generalization with new data. Another example can result from false positives, both during the operational phase of the robot and the interpretation of the model's results. Differentiating between the terms fault, error, and failure in the context of failure and false positive prediction is significant. A fault is a hypothesized cause of an error; an error is a deviation that occurs, and failure is when the system is not delivering the expected outcome. Notably, the error is the earliest detectable manifestation in these kinds of predictions (Borkowski et al., 2019). There are unexpected factors when it comes to balancing the two broad fields that can appear if one fails to provide the intended results. The examples help to acknowledge that to avoid unexpected situations, even when models have undergone comprehensive training for intelligent automation, the company needs to prepare for human intervention to provide proper updates, parameter improvement, or retraining (De Moraes et al., 2022). Davenport (2009) additionally argues that human intuition and judgment as a backup should always be accessible to revise the criteria or algorithm if the automation no longer works.

2.2.4 Data Sources

To achieve the goal of rule-based RCA, log messages serve as a main source of information for identifying system failures. According to UiPath's research (2020), operational data produced from the robot execution and valuable for the failure classification model can be gathered from the RPA automation management tool and divided into three groups: robots deployed, processes, and queues. A deeper evaluation of the first group investigates the productivity, capacity, utilization, and error measures. The second group focuses on the throughput, success rate, duration, and exceptions rate. Finally, the third group focuses on processing transactions, average handling time, and service level agreements.

However, a challenge with the log file data is their generally unstructured nature. Therefore, formatting the data into a uniform structure is necessary to extract the relevant information for analysis. An investigation of the log messages reveals a pattern of error occurrence before a system failure (Gurumdimma & Bisandu, 2018). Automation management tools that contain real-time monitoring, search, filter, alert, and other useful features for providing operational audit trails are beneficial in this case as they can provide information for analytics purposes. Process analytics tools can merge and clean raw data, like the log messages, from other various data sources as well, such as Process Design Documents (abbreviated: PDD) documentation, which illustrates the steps of the business process flow and error handling reports used by the development team to record and organize diagnostic information for different processes.

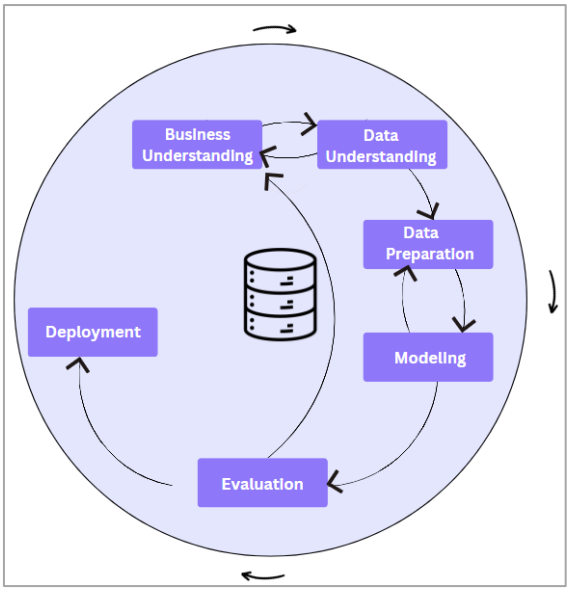
3 RESEARCH METHODOLOGY

3.1 CRISP-DM Methodology

The general methodological approach is case analysis, employing a real-life context of the defined research problem. A case analysis is elaborated as one aspect of a historical event chosen as suitable for internal examination (George & Bennett, 2005). For this analysis, the case chosen is the implementation of data analytics in RPA processes, underscoring the success and challenges accompanied by it in the context of automation technology. A case analysis approach was selected due to its usefulness in providing a foundation for theoretical development and its capacity to reveal a greater understanding of individual cases (George & Bennett, 2005).

This thesis adopts the Cross-Industry Standard Process for Data Mining (abbreviated: CRISP-DM) methodology, technology, and industry-independent data mining process model that provides a framework to enhance the reliability of the project and guide the project execution steps (Schröer et al., 2021). The methodology lifecycle consists of six phases illustrated in Figure 4. The decision to adopt the CRISP-DM phases for the case analysis was driven by its ability to provide a manageable structure throughout the entire process, enabling a more organized approach to expand on specific aspects in the future. Moreover, its flexibility ensures that it can be adapted to meet the requirements of this research successfully and that the research objectives are met.

Figure 4: CRISP-DM Methodology



Source: Stirrup & Ramos (2017).

The first phase is dedicated to defining the business objectives, involving problem clarification, project goal setting, and designing a plan to achieve the objectives (Wirth &

Hipp, 2000). First, it is important to understand the link between business and data context. Secondly, the methodology phases are not linear, meaning that the direction can be refined, the earlier phases can be revisited, and the subsequent phases can be decided based on the objectives determined, thereby amplifying process agility.

The data is collected and explored in the data understanding phase to gather insights. The succeeding data preparation phase entails cleaning and transforming the data, ensuring it is prepared for analysis (Stirrup & Ramos, 2017).

The fourth phase focuses on applying appropriate modeling techniques. A model is built and evaluated against specific criteria to ensure it meets the most favorable outcomes (Wirth & Hipp, 2000). All the necessary technique choices are closely linked to the type of project and data available.

During the evaluation phase, the model's results are checked about the predefined business objectives (Schröer et al., 2021). The final phase involves deploying the model along with necessary action steps or a user guide to ensure its usefulness for future applications. This can be presented in a report, or, for a more complex solution, a repeatable process can be administered (Wirth & Hipp, 2000). The deployment of the model does not denote the project's conclusion; ongoing maintenance is required as new requirements emerge, triggering the need for answers and modification.

3.2 Interviews

The selection of qualitative research aligns best with the thesis's aim, as it is explorative and manages to uncover a deeper understanding of detailed topics. The description of the qualitative data collection methods used, such as semi-structured interviews, assists in evaluating the overall quality of the research. In preparation for the interviews, the productive RPA infrastructure was researched, the department's process documentation was reviewed, and a list of questions was crafted. The interview questions were designed based on theoretical research about RPA challenges and best practices aimed at process optimization, in combination with findings from the process documentation on related topics. The questions were designed to align closely with the participant's job role, aiming to gain a deeper understanding of error categorization, team communication dynamics, the enhancement of data analytics utilization, and the current and future role of ML in RPA processes. The interview questions are listed in Appendix 2.

Six semi-structured interviews were conducted remotely via a video call, ensuring detailed notes were taken during the process. Before beginning the interview, the thesis's topic was explained to each participant, outlining the significant role of their input in answering the research questions. Even with a level of preparedness, the interview format and open-ended nature of the questions also allowed for flexibility and spontaneity, reaching a more profound understanding as the participant's narrative unfolded (Galletta & Cross, 2013). During the

interview, it is important to listen attentively, clarify the meaning when necessary, and note any points to return to for further elaboration that may have been reformed during the progress of the discussion (Galletta & Cross, 2013). In Table 1, an overview of the interview schedule details is provided.

Table 1: Interview Schedule Details

Job position	Time	Duration
Technical Consultant at UiPath	14.09.2023	1 hour
RPA Developer	14.09.2023	1 hour
RPA Developer	15.09.2023	1 hour
RPA Consultant	19.09.2023	1 hour
RPA Consultant	22.09.2023	1 hour
RPA Developer	27.09.2023	1 hour

Source: Own work

The participant's experience plays a crucial role in the interviews as their knowledge of the specialized information highlights how decisions are made in practice and what might be the optimal way. They may even successfully anticipate possible events in their field. The external expert participant, a Pre-Sales Technical Consultant at UiPath with over 18 years of experience, brought extensive technical knowledge to the discussion, regarding current and future RPA developments, the UiPath platform, the implementation of artificial intelligence within the existing platform tools, and the refinement of monitoring insights and dashboard creation. The titles of internal RPA team participants comprised two RPA consultants and three RPA expert developers, each having great experience in their field and a pivotal detailed understanding of the process specifics within the department.

The participants shared valuable details of the data types that produce the most valuable results in data analytics for automation monitoring and identified the factors that predominantly cause failures based on their experiences. The discussions generated a detailed understanding of the RPA processes, the systems used, and their application. Test case scenarios were developed for the thesis using the gathered error categories. These scenarios are directly informed by the types of failures identified by the experts, ensuring they closely mirror the real-world challenges faced by the RPA team and are likely to yield the anticipated outcomes. The purpose of the interviews is explained in larger detail in the subsequent subsection 4.1 which is used to elaborate further the creation of error categories from experiences each interviewee shared.

3.3 Data Collection Techniques

Data sources include an automation monitoring tool, an analytics engine, a software package, an identity management system, and internal Excel used by the team. Of the mentioned leading software companies that provide RPA solutions and offer extensive automation capabilities, dmTECH uses UiPath. This is why the referral to the test case RPA processes as well as the data extraction process will portray figures from the UiPath Studio and UiPath Orchestrator tool, the modules of the UiPath platform. UiPath Studio is a tool where workflows are designed, modeled, and executed ensuring the transfer of packages to the automation monitoring tool Orchestrator (Ribeiro et al., 2021). Orchestrator is a centralized platform for managing, monitoring, and controlling UiPath robots (UiPath, n.d.) and is where the unattended robots are triggered. The analytics engine used for storing log records is linked to Elastic Search, which is built on the Apache license search engine, with a Kibana data visualization plugin (Ribeiro et al., 2021). Robot dependencies are retrieved from the robot's GitLab repository, and the duration of robot roles is extracted from the company's Identity and Access Management (abbreviated: IAM) system. The IAM system prompts the automation of user administration workflows by aggregating permissions into roles. Roles are assigned to employees or, in the case of RPA, to robot users, which are identified through credentials under which the robot operates within the environment called Windows identity. Mandatory models within the IAM system contain a digital identity, account, and permissions assigned to the account to grant IT resources. This setup reduces administrative effort and, more importantly, enhances security measures by monitoring and restricting what the robot users can access (Kunz et al., 2019).

Consequently, the dataset was framed from Orchestrator job data (status, timeframe, machine name…), Elastic logs data, IAM user role data, Gitlab dependencies data, and internal Excel file data (documented updates).

To align with the interview findings, the test case robots used to collect process data were explicitly designed to match activities that display the highest failure rates. The selection was aimed at delving deeper into the underlying causes of these failures. The decision to create the test cases was further motivated by the need to maintain confidentiality, ensuring that no sensitive data from the production environment would be disclosed. A new robot user was created to facilitate this, and the necessary access rights that the user needs were reviewed and ordered in cooperation with one of the RPA consultants.

Given the necessity for the data collected to span for a longer period, the timeframe offered by the initial test cases proved insufficient. Additional earlier data was incorporated from several processes in the testing environment to address this issue. The testing environment is primarily used to evaluate the robots before integrating them into production - the inclusion of the additional data allowed for the expansion of the dataset to reflect seven months. To fit confidentiality purposes, the names of added processes, their Windows

identities, and the machine names used for testing have been altered through data anonymization.

After the dataset was framed using the extraction robot, data preprocessing was done using Python programming language in the Jupyter Notebook platform. This involves merging, sorting, standardizing headers, formatting, and handling missing data to ensure uniformity. Subsequently, text classification techniques were applied to categorize errors, where categories are defined in a dictionary with associated keywords for each category, and rule-based RCA was conducted to correlate error categories and their respective failure reasons.

Visualizing the data for descriptive analysis allows for the creation of insights into the most commonly faced robot failures. To assess the preferred approach, semi-automated or automated, both are compared based on the ease of implementation, the need for human interpretation, the likelihood of accurate results, stability, robustness, real-time prediction, and flexibility. Once a superior approach is chosen, from the comparative study, the empirical study determines how to leverage its outputs long-term and what the savings are in different aspects from the company’s initial situation.

4 RETAIL COMPANY ANALYSIS

4.1 Overview of the Initial Situation

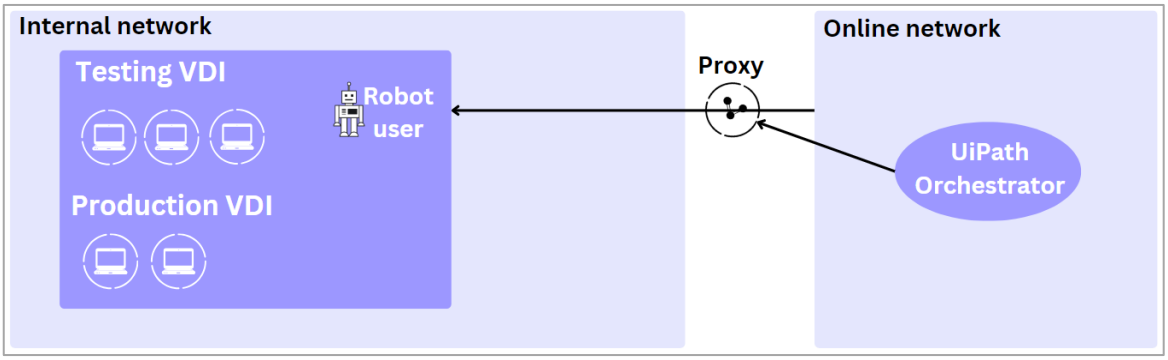
The prominence of maintaining effective workflows and enhancing operational efficiency by minimizing the failure amount was addressed in subsection 2.1.3, and in this chapter, the practical aspects are detailed. Likely scenarios featuring integrating new robot users or machines, managing system updates and maintenance while preserving quality interconnections, and ensuring that processes run individually on the machines, underline the importance of perceiving the initial situation of a company's RPA environment. The initial situation information was collected from the sources described in subsection 3.2. Specifically, the details about the steps were gathered from the process's internal documentation, while insights into the team dynamics and grouped error categories, were gathered through interviews.

The RPA journey begins when a team, currently performing a process manually (referred to as the functional team) identifies the need for automation and communicates this to the RPA team. The RPA team then meticulously records and documents the manual process steps. During the process evaluation, following the business process suitability framework criteria of an applicable RPA robot, as explained in roadmap subsection 2.1.4, establishes if the process is suitable for RPA. This decision-making process considers the key requirements, such as the effort-to-savings ratio, the amortization period, and other details related to the process lifecycle, including development time, complexity, and execution frequency. Suppose it is concluded that it is suitable. In that case, the RPA team creates implementation

documentation capturing important authorization information, such as the tools the robot uses and the necessary access rights.

When the UiPath Studio development process is finished, the robot is deployed to the Orchestrator's testing environment. The connection from an internal network, such as the Virtual Desktop Infrastructure (abbreviated: VDI) environment, to an external online network like Orchestrator, is facilitated through a proxy server that acts as an intermediary. For streamlined organization, both testing and production environments are represented in Orchestrator by their machine name, which is explored in detail during the CRISP-DM phases.VDI warrants users to remotely connect to the data center employing solutions such as Citrix or VMware Horizon View, providing access to the operating system, data, and applications as if operating on a local machine (Sheikholeslami & Graffi, 2015). Similarly, the proxy selectively allows traffic to enter or leave a network, processes it, and forwards it, effectively enhancing security and control (Tracy et al., 2007). The connectivity setup is illustrated in Figure 5. This figure was created in consultation with one of the RPA experts, offering additional context to the process.

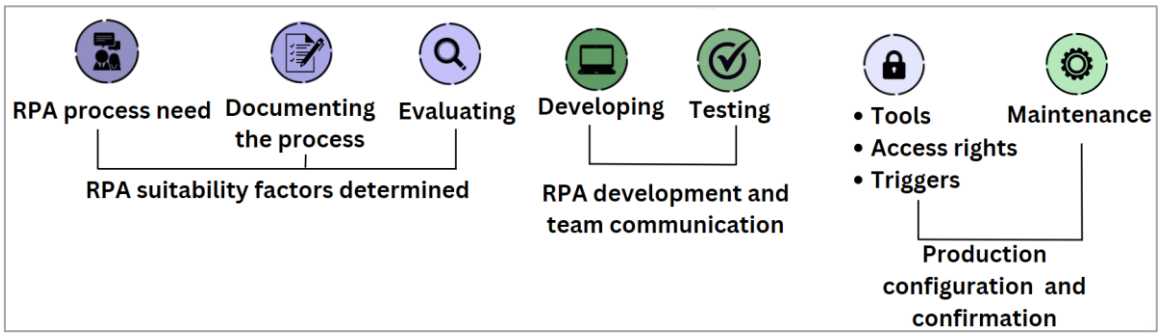
Figure 5: Connection Through Proxy Server



Source: Own work

The RPA responsible team manages and monitors robot utilization in the testing environment. This entails first ensuring that the testing VDI, designated for robot execution, is equipped with all the necessary applications, as outlined in the initial documentation. Second, asserting that the user linked to the robot has all the required permissions set up in the IAM system. Third, monitoring through Orchestrator allows the team to observe logs and verify successful job execution. Developers can delve into intricate details via connected Elastic logs, finalize the robot's code on GitLab, and thoroughly document the execution steps to streamline future maintenance. Finally, the latest version of the robot is uploaded to the Orchestrator production environment, where the unattended robot user and applications are configured on the production VDI. The robot's operational parameters, such as start times and maximum run duration, are set according to agreements with the functional team. The entire process is depicted in Figure 6.

Figure 6: RPA End-to-End Process



Source: Internal documentation (2023).

Having explored the RPA process and infrastructure, and acknowledged the team's need for resources in bug-fixing and error handling, it is insightful to examine the team's structure and its effectiveness in overcoming challenges. This understanding provides a solid foundation before delving into the common errors encountered by the RPA team. The team consists of expert developers, student developers, and RPA consultants. Through the interviews conducted with each team member, it was calculated that expert developers typically spend an average of 2 hours per week on bug fixing, student developers allocate approximately 3 hours weekly, and RPA consultants also spend 2 hours. One of the RPA consultants explained, “Each of us must monitor the robot email outputs to track if everything is functioning correctly. While solving a problem, we prioritize maintaining open communication between all involved parties. This ensures that the developer responsible analyzes the technical issue and that the functional team is up to date. Depending on the complexity of the problem, we inform on the estimated time needed to solve the problem and follow up on the progress.”

The current situation acknowledges the importance of monitoring and maintenance to improve robot performance; however, it needs to fully follow the theoretical research recommendations as it lacks an in-depth analysis of the root causes. To address this discrepancy, including regular robot performance assessments with visualizations could reduce the need for rework and the time spent on recovery mechanisms. Analyzing historical data and past activities provides the team with tools to understand the robot's performance. Historical data tells the team what they did while analyzing the results, allowing for focused improvements based on past outcomes (Sheikh, 2013).

To reach the goal of operational efficiency by reducing failures, the focal point in the initial situation is related to the most common errors encountered, which can eventually help identify their underlying causes and failure reasons. The grouped error categories mentioned below, which also guided the selection of test case activities, reflect the activities most prone to failure, as identified in the research methodology. This means that if the team encounters a specific category, it is more likely to experience associated robot failures. The interviews

conducted aimed to identify the most common errors encountered by each developer, serving as a foundational element in forming the error category groups.

As illustrated in Table 2, each macro group contains connected categories, and the text descriptions provide a clearer understanding of what each group encompasses. Certain errors are outside of the team's immediate influence or control. For instance, within the proxy server macro group, challenges such as the robot user being unable to access dedicated libraries and packages - due to a change in the cloud storage location or temporary unavailability of the VDI environment - are not problems that the RPA team can influence directly. The most common Excel error is the unexpected processing error. The application group is expected to have the highest frequency of errors, it contains cases such as system maintenance period incapacitating RPA activities. Additionally, other errors placed in this group are related to user access rights, such as cases where a user cannot access a system or transaction.

The Orchestrator group contains cases where potentially not all steps have been taken to prepare the robot user, leading to the failure of Orchestrator or Office 365 activities. Such issue examples include the robot user not being correctly added to the folder, missing robot assets compared to the configuration file, and SharePoint activity errors happening. The malfunctioning code group addresses non-compliance with coding standards while creating the process. Examples here include using UiPath legacy language or improper error handling within the process structure. The ideal structure adapts to a standardized framework that guides more stable automation. All prerequisites are regarded in it, from closing open applications before job execution to standardizing the catching of business and system exceptions, setting up subtasks correctly, and precisely processing transactions in line with business logic and the configuration file (Potturu, 2023). Potturu (2023) further states that adhering to consistent design patterns within a framework principle leads to better quality solutions in a transparent environment. This is why the malfunctioning code group review issues in the framework processing methodology. The last two macro groups, UI navigation, and Browser, relate to errors resulting from UI changes, such as browser extension issues, page loading delays, or web element updates. Given the rule-based nature of RPA struggles with unexpected UI changes, tracking error occurrences within these groups is important for achieving the optimization goal.

This grouping was grounded in the team's collective experiences, focusing primarily on the frequently used tools and the infrastructure setup, which are central to the process workflow. The establishment of these error category groups serves multiple purposes. Firstly, it provides a clear framework to pinpoint the most prevalent errors within the team. Secondly, it lays the groundwork for a deeper diagnostic process, where the identified errors act as indicators, guiding further investigation into the underlying causes of each failure. Lastly, and significantly, this categorization enhances team communication and retrospective analysis. It offers a structured platform for the team to reflect on recent challenges, suggest additional frequent errors for inclusion, and collectively enhance their understanding and

monitoring of ongoing and potential issues in improving the monitoring step. The integration the monitoring step is possible at various stages of the RPA end-to-end process depicted in Figure 6, thereby enhancing overall team synergy.

Table 2: Error Category Groups

Proxy	Orchestrator	Browser	UI Navigation
• Proxy package access • Proxy VDI unavailable • VDI network access • VDI updates	• Orchestrator assets • Orchestrator user setup • Orchestrator SharePoint activities	• Browser extension	• Dynamic web elements • Selector setup
Application	**Malfunctioning code**	**Excel**	
• SAP maintenance • User access rights • SAP transaction access rights • SAP system access rights	• File exists • Null reference • Cancellation request	• Excel Error	

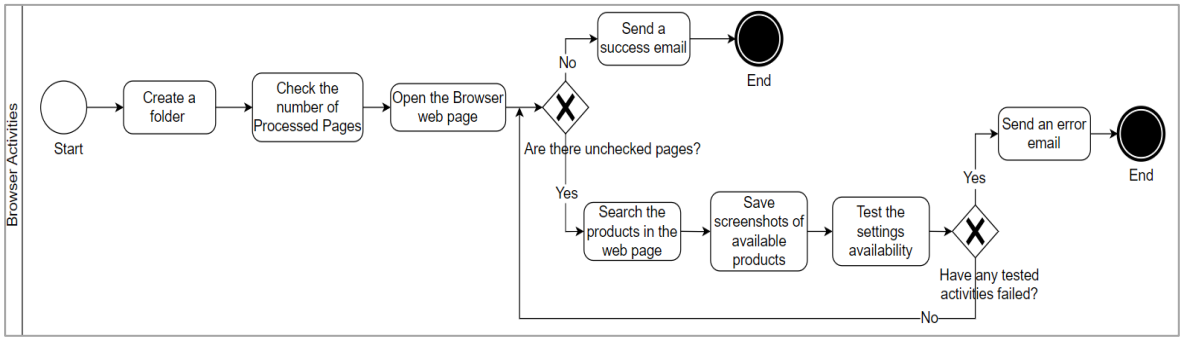
Source: Own work

4.2 Data Analysis with CRISP-DM

4.2.1 Business Understanding

The business goal, derived from section 4.1 of this chapter, focuses on identifying the most common failure reasons and the factors contributing to them, addressing the business problem of RPA robot failures. To directly influence this problem and build on the significance of PDD documentation elaborated in subsection 2.2.4, test cases were formulated, and their corresponding PDD flowcharts were developed. The purpose of these tools was to categorize the errors encountered in Table 2 and to collect the operational outputs. The process for browser activities, depicted in Figure 7, focuses on UI navigation, such as clicking and typing.

Figure 7: Browser Activities Test Case

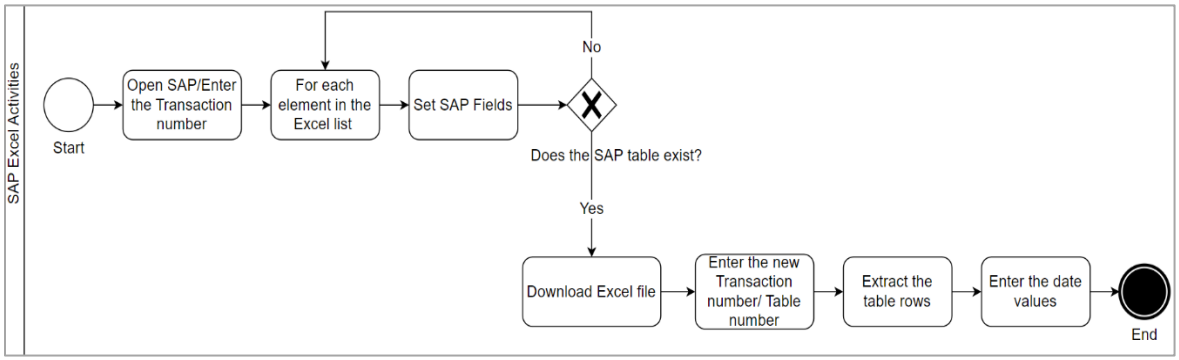


Source: Own work

Changes in the UI consequently lead to process errors, as the robot, which relies on stability, may not be able to determine how to proceed. The process first checks whether the browser connection is established.

Figure 8 depicts a combination of SAP and Excel operations for the processing of downloaded files. The integration of RPA processes stems from the expected stability of the SAP interface and the repetitive nature of data input.

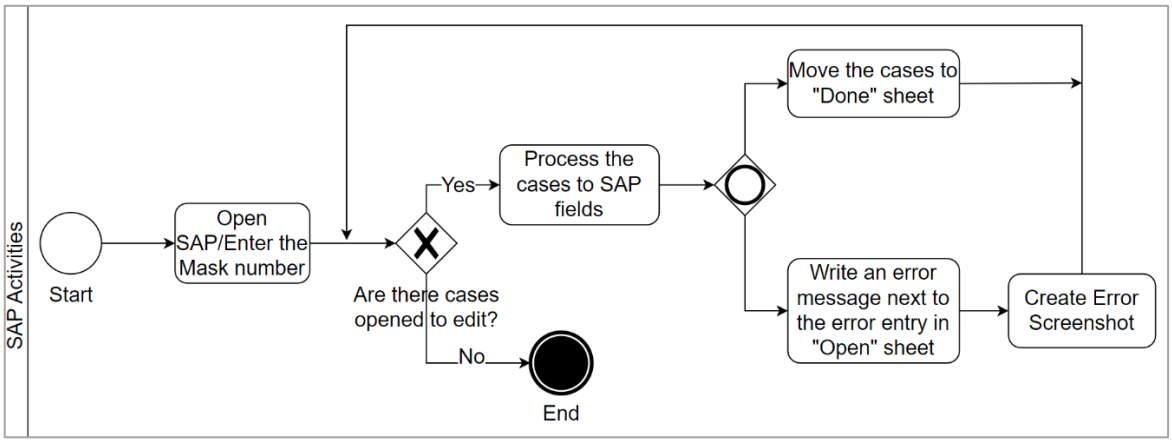
Figure 8: SAP Excel Activities Test Case



Source: Own work

This emphasis arises from leveraging SAP's capabilities for managing details surrounding business operations as an enterprise resource planning system. The portrayed process is centered on testing the stability of SAP-related tasks, similar to the SAP activities process shown in Figure 9. These processes include activities such as launching the SAP application, entering transaction numbers, confirming that the user has the necessary permissions, populating SAP fields, and proceeding based on conditional logic to the following transactions.

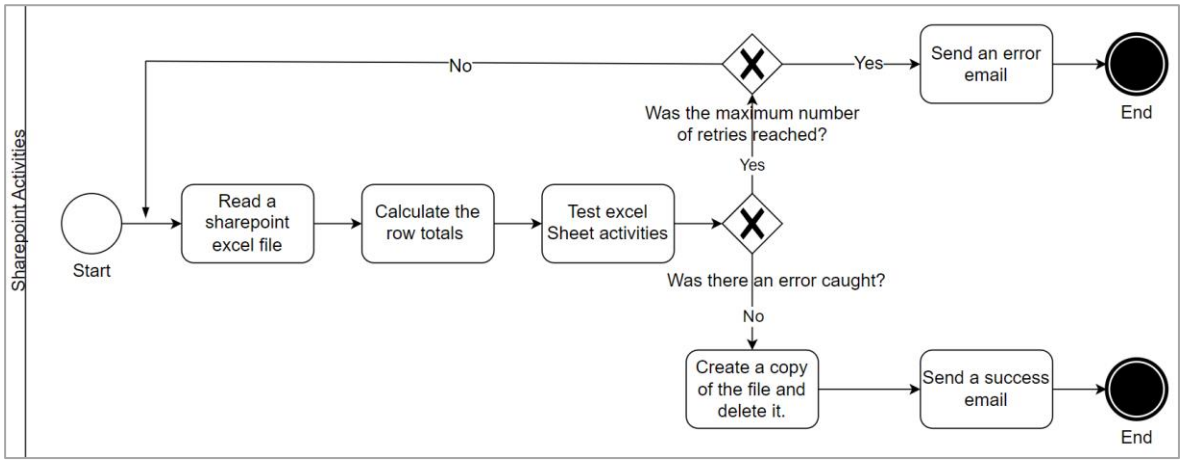
Figure 9: SAP Activities Test Case



Source: Own work

Finally, SharePoint activities processes, whose flowcharts are presented in Figure 10 and Figure 11, are centered on Microsoft 365 operations, highlighting the connection properties, assets, and correct activity usage. The processes test frequently used Microsoft 365 activities, including creating, populating, and deleting from folders, renaming files, and sending an email using the previously established SharePoint connection, to name a few.

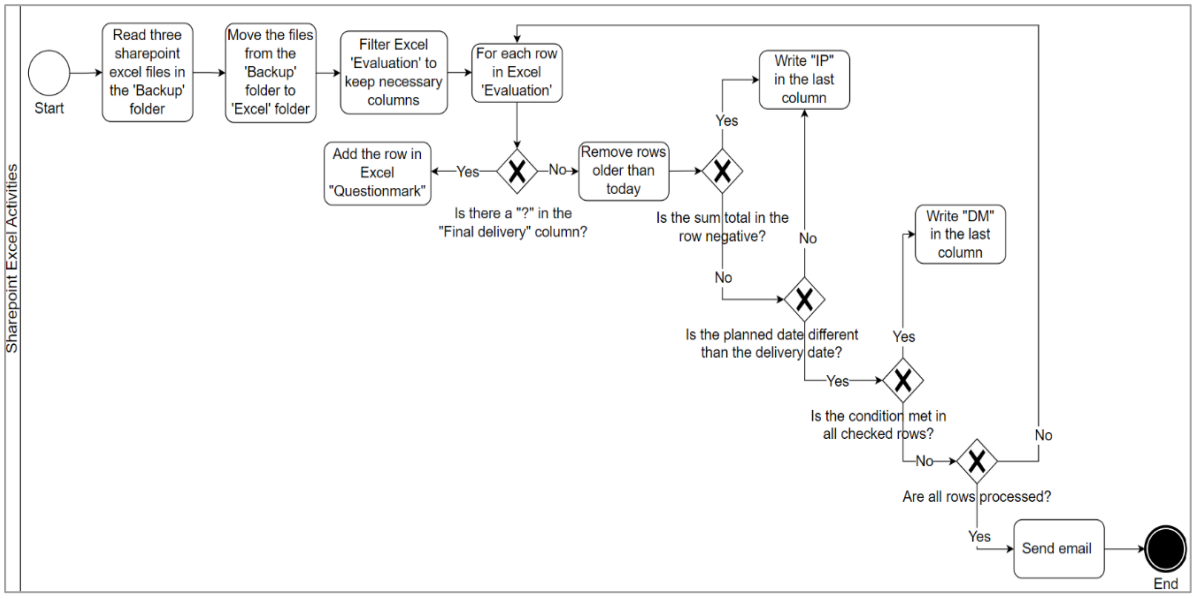
Figure 10: SharePoint Activities Test Case



Source: Own work

The SharePoint Excel activities process, illustrated in Figure 11, further integrates Excel operations to ensure proper processing of columns and the execution of string inputs in expected rows based on conditional statements. Many existing processes use these tools and encounter potential issues, such as incorrect date inputs.

Figure 11: SharePoint Excel Activities Test Case



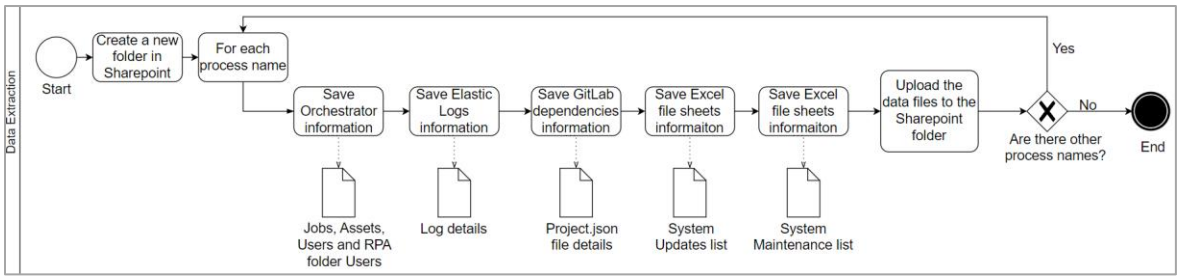
Source: Own work

The business assumptions made at this point are that the data extracted by the RPA robots will be consistent with format and quality and that all relevant errors will be logged with sufficient detail to be effectively categorized. Other related assumptions are that the error categories identified during team interviews comprehensively cover the most common errors and that the addition to the checks within these categories will be stable and straightforward. The final assumption is that the RPA department is equipped to manage the changes introduced by implementing additional data analytics methods. To address these assumptions, which pose potential risks, aside from ongoing RPA monitoring, and continuous stakeholder engagement (RPA developers, RPA consultants, operations managers, functional team, etc.), regular reviews of the rule-based RCA logic should be conducted. The rule-based RCA, as outlined in subsection 2.2.3 involves identifying the root causes based on a predefined set of rules, which in this case are the underlying 'failure_reasons' that consider both the identified error category and supplementary information from a reference file. Such reviews should focus on updating the keyword dictionary and the predefined criteria validations against the reference files. These steps can ensure the successful mitigation of assumptions.

4.2.2 Data Understanding

In Chapter 3, the dataset was described, including the sources of data collection. The dataset is gathered using an RPA data extraction robot, which collects, appropriately names, and stores it in a designated SharePoint folder used for subsequent processing. The process flowchart of the extraction robot is detailed in Figure 12.

Figure 12: Data Extraction Activities



Source: Own work

The robot is designed to save Orchestrator job files according to the process names specified in an input file, storing each file in a ‘Jobs’ folder. Additionally, it also captures assets and user data by process name from the same application but stores these in a separate ‘Assets and Users’ folder to maintain organization. The ‘RPA users’ folder user data from Orchestrator, which is not process-specific, is categorized by machine name. This categorization indicates if the data is extracted from the testing or production environment, with each file marked accordingly. Furthermore, the robot extracts Elastic log details and Gitlab dependencies, organizing them by process name. Elastic log details are stored in the

‘Logs’ folder, while Gitlab dependencies are compiled into a single ‘Dependencies’ file. In addition to these tasks, user roles and SAP maintenance data are retrieved from the IAM system, using the machine name. They are then saved in the ‘System maintenance list’ and ‘IAM entitlements’ files, respectively. Lastly, the ‘System updates list’ file undergoes a verification process to ensure that all update dates are current and up-to-date.

Certain tasks still require manual execution in this process. The tasks are as follows:

- **Dependencies**: The latest dependencies available must be manually acquired from UiPath Studio and added to the prepared file.
- **System updates:** Tracking and recording of the dates of system updates must be manually done. Updates occur in a consistent sequence each month.
- **Input file:** The process names that should be extracted, as well as the machine names for each process, need to be manually specified in the input file.
- **Validation:** After the data preparation phase, the output prepared for analysis requires a manual review. The review can consist of checking if the file is correct and meets the expected standards, especially regarding the consistency of date columns.

4.2.3 Data Preparation

The purpose of the dataset, detailed in this section, is to enable a descriptive analysis aimed at understanding patterns and factors influencing robot errors, which are later reported. To achieve this, following the findings from the data understanding phase, it becomes clear that the data preparation phase requires data preprocessing and exploratory data analysis. Data preprocessing is an essential step for preparing the dataset suitable for analysis, specifying the procedures needed to reach the final version. The purpose of the exploratory data analysis is to examine and understand the main characteristics of the data before proceeding with modeling.

4.2.3.1 Data Preprocessing

After extracting the data, the next crucial step involved preprocessing the reference Excel files, which would serve as a foundation for creating failure reasons based on error categories. Oracle's guide (2018) outlines that reference data is employed for organizing and maintaining information by defining and tracking various categories. The specific details tracked depend on the structure or format of the reference data. The data preprocessing of these files was multifaceted; the data cleaning part entailed the removal of any empty cells and duplicates, in addition to replacing unwanted characters to preserve data quality and avoid inaccuracies in subsequent analyses.

Furthermore, an effort was concentrated on retaining only the relevant features, which reduced the dimensionality and complexity. Data normalization thereafter ensured the

standardization of column names across the files to maintain uniformity. Finally, a critical step involved fixing formatting inconsistencies, such as structuring the files coherently by splitting the dataset into multiple columns where necessary. For example, the 'log_date' and 'log_datetime' columns were created by splitting the timestamp, providing a clearer and more organized overview of the data. A significant part of this process was standardizing date formats across the datasets and sorting the data based on the 'log_date' column. Instances of Python commands used for preprocessing the reference files are documented in Appendix 3.

To merge jobs and logs datasets into a single dataset, which is the basis for the descriptive analysis, a shared key identifier 'job_log_id' was introduced. This identifier uniquely represents each process job by combining the process name with a sequence number. A left outer join was employed for the merging process. The Python commands used are presented in Appendix 4.

In the data preprocessing pipeline for the merged dataset, feature engineering techniques were utilized to further clarify the data. Feature engineering is the process of generating new variables from existing data, which are represented as columns in a dataset. This process is achieved by processing or combining two or more existing columns to create a single new column (Chicco et al., 2022). Applied feature engineering techniques include **feature construction** and **feature extraction**. Feature construction generates new features that offer more meaningful information in a single feature, reducing the number required. On the other hand, feature extraction transforms raw data into a structured set of features that convey more significant insights (Lensen et al., 2016). Lensen et al. (2016) demonstrate that feature extraction can involve generating histograms or gradients within an image to capture patterns. In this analysis, the extraction technique transforms combinations of log messages and information from reference files into structured columns.

Several new columns were created to enrich the dataset through the process of **feature construction**, each serving a specific analytical purpose in RPA job performance. These columns include: 'newly_added', indicating whether a process has been recently integrated within the selected Orchestrator tenant project version; 'retry_number' and 'maximum_retry_number', which track the number of consecutive transaction retries; and 'CPU_usage', which monitors system performance. Further enrichments include 'log_exception_type', 'secondary_causes', 'false_success', 'false_failure', 'increment', 'old_UiPath_activities', and 'simultaneous_process'. Detailed descriptions of these columns are presented in Appendix 5.

The **feature extraction** process involves creating the 'error_categories' column from log messages by identifying specific patterns in the logs, through dictionaries that contain a list of keywords associated with common error groups, as outlined in Table 2. This method allows for structured analysis of otherwise unstructured text data by categorizing log messages into meaningful groups. The rule-based RCA implementation then examines various causes behind each error. For instance, errors related to orchestrator user setup, if

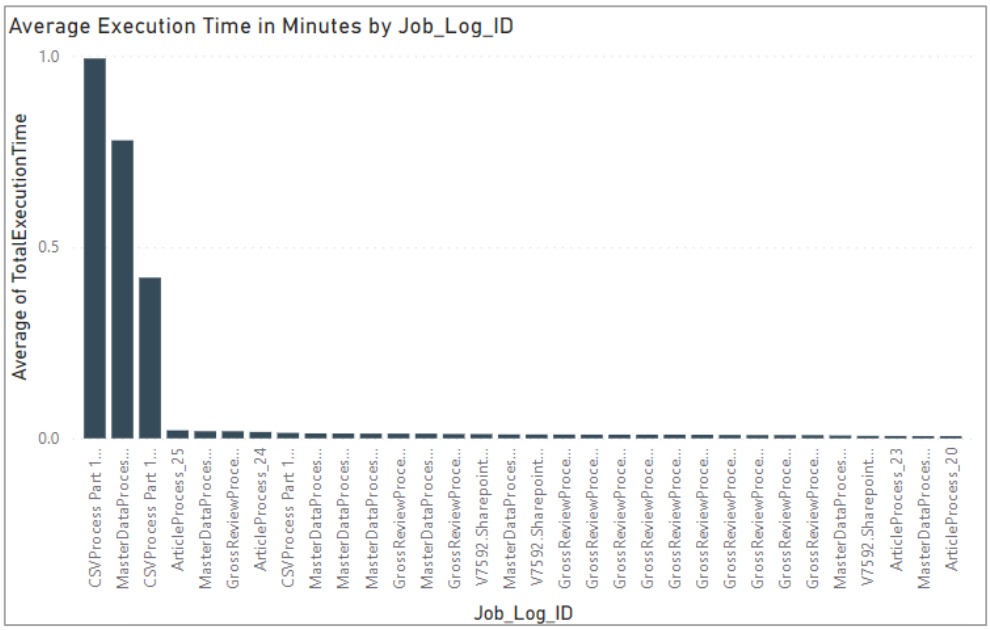
identified within the log message patterns, are investigated through a series of systematic assessments, such as verifying the entry of Windows identity and machine name, comparing project version between Gitlab and Orchestrator, ensuring the robot role's currency, and determining an output if the mentioned assessments do not produce any results. The Python commands utilized for this analysis, along with the derived failure reasons are detailed in Table 6, found in Appendix 6. This structured approach is saved within functions and mirrored across other error categories, such as SharePoint errors, where assessments include validating user roles and the existence of necessary IDs, along with analyzing specific patterns in log messages pertinent to that category. The Python commands used for these failure reasons are detailed in Table 7 of Appendix 6. Creating the 'error_categories' and 'failure_reasons' columns enriches the dataset, providing an understanding of the origins of identified errors through targeted assessments gathered from prior experience with such errors, thereby clarifying the underlying reasons for failure.

The dataset, primarily composed of categorical data and encompassing comprehensive log rows for each job, was transformed into a format more appropriate for analysis, a necessary step to enhance the interpretability of the analysis. The transformation was partially done by applying the one-hot encoding technique. This technique represents categorical variables as a group of binary variables, where each binary variable indicates the presence of a category with a value of 1 and its absence with a value of 0 (Galli, 2020, pp.92-100). Additionally, during the process of data alignment, the 'error_categories', 'failure_reasons', and 'secondary_causes' columns were retained in their categorical format. This ensured that the Python commands preserved only the unique error rows and that the relationship between these columns was accurately maintained.

4.2.3.2 Exploratory Data Analysis

Before continuing to the modeling phase, it is necessary to explore the merged dataset. This foundational step involves a detailed examination and summarization of the dataset, to effectively understand its characteristics (Chicco et al., 2022). Using the PyCaret setup function, which facilitates insights into the type of variables present, it was concluded that the dataset comprises 15 numeric features, 2 date features, and 11 categorical features, and exhibits 9.7% rows with missing data. Initial visualizations in PowerBI, analyzing the average execution time by process job, as illustrated in Figure 13, reveal outliers with the execution time exceeding expectations determined by the bot's trigger. This helps understand which values need to be filtered, since as a rule of thumb, it is suggested to filter the features that are at least ten times the median value (Chicco et al., 2022).

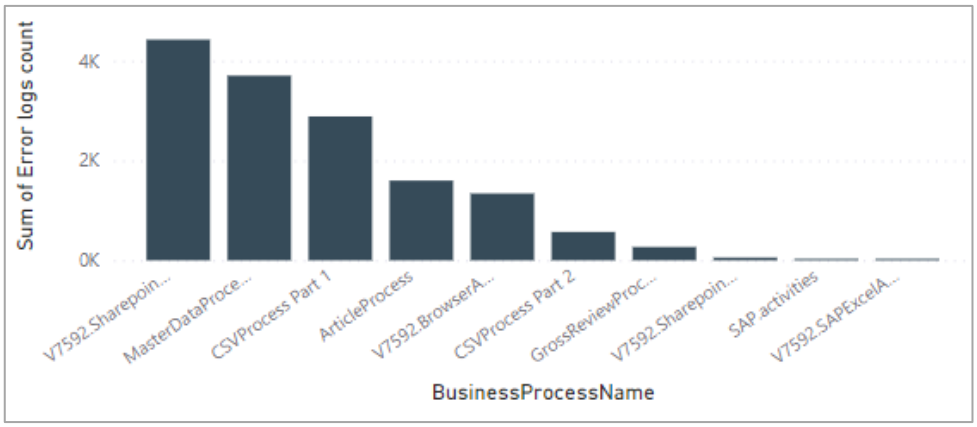
Figure 13: Average Execution Time Outliers



Source: Own work

Figure 14 showcases the error log count by the process name. This suggests that processes with higher frequencies of errors per job, such as the SharePoint Excel activities process, require closer investigation or optimization. They should be examined first to determine whether the errors encountered in these processes are common across others or unique to a specific process.

Figure 14: Error Logs Count by Business Process Name

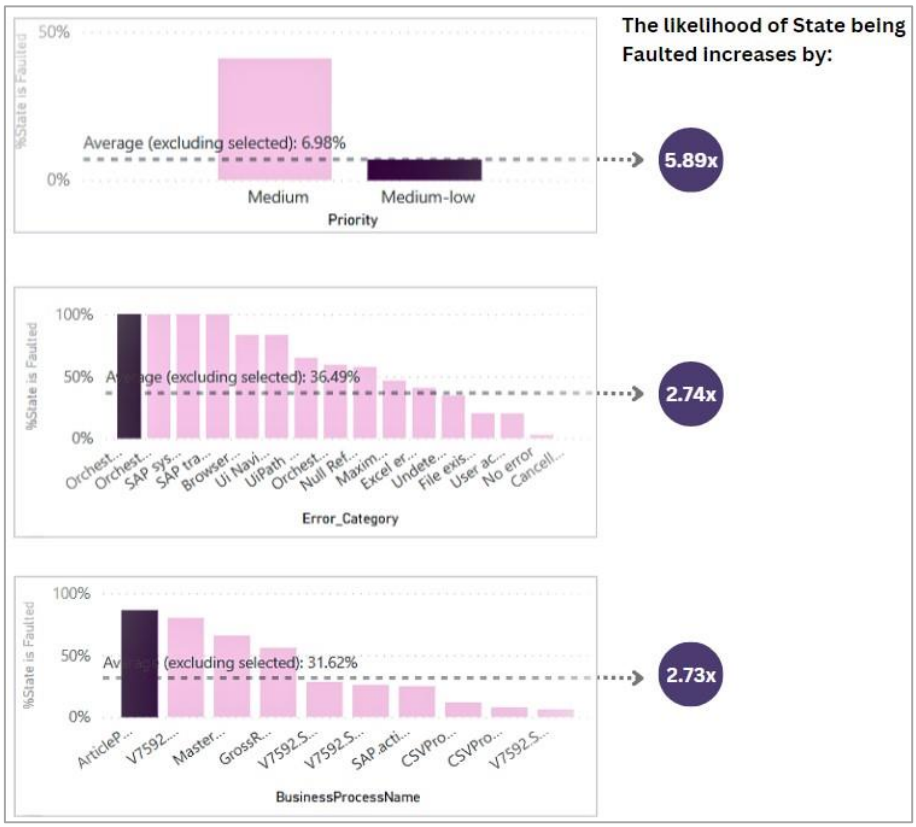


Source: Own work

Power BI's key influencer visual, suitable for categorical data analysis in this dataset, aims to answer what factors influence the likelihood of robot faults. This is done with a focus on identifying specific processes, error types, and the priority level impacting fault likelihood. It is important to note that the visual considers the number of data points when determining

whether a field is an influencer or not. After analyzing Figure 15, it is concluded that tasks of medium priority are more susceptible to failure, and Orchestrator asset setup is one of the top four error categories leading to such outcomes. Further, the 'ArticleProcess' is identified as being more prone to failure, with 86% of jobs within this process faulting compared to the average fault rate of 31.62% across other processes. The difference between the two numbers gives the presented likelihood.

Figure 15: Process Fault Likelihood



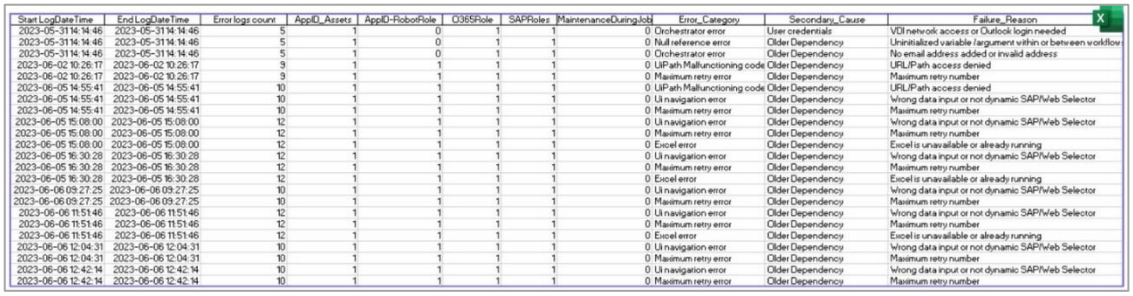
Source: Own work

Regarding the relationship between CPU usage and total execution time, a correlation coefficient of 0.68 indicates a positive correlation. This suggests that higher CPU usage may be associated with longer execution times, identifying opportunities for further investigation into process efficiency improvements. It also implies the need to consider the influence of other variables on this relationship.CPU usage data is presented only in test cases, not across all processes because the CPU usage information was extracted using monitor process information activities in UiPath Studio, that have not yet been integrated into existing processes. The situation suggests either the potential utility of collecting such data more broadly or the need to develop alternative methods for analyzing resource utilization.

Python commands were used to ensure data cleaning was correctly done in the final dataset, keeping only the necessary features and dropping the duplicates. The final dataset contains 2 testing machines on which the robots were run, 4 Windows identities, and 10 business

processes. Further, there are 36 failure reasons within approximately 780 rows, each presenting errors faced in one process job from which 476 represent a unique process job. This is spread across 28 variables encompassing a wide range of information, including 'business_processname', which represents the names of the business processes, and 'job_log_id', serving as the unique identifier for each job. Additionally, the 'state' variable indicates the job's outcome as either successful or failed among other variables such as 'machine_name', 'windows_identity', and 'total_execution_time'. A selection of the variables is represented in Figure 16.

Figure 16: Part of the Excel Dataset



Start LogDateTime	End LogDateTime	Error logs count	AppID_Assets	AppID-RobotRole	O365Role	SAPRoles	MaintenanceDuringJob	Error_Category	Secondary_Cause	Failure_Reason
2023-05-31 14:14:46	2023-05-31 14:14:46	5	1	0	1	1	0	Orchestrator error	User credentials	VDI network access or Outlook login needed
2023-05-31 14:14:46	2023-05-31 14:14:46	5	1	0	1	1	0	Null reference error	Older Dependency	Uninitialized variable /argument within or between workflow
2023-05-31 14:14:46	2023-05-31 14:14:46	5	1	0	1	1	0	Orchestrator error	Older Dependency	No email address added or invalid address
2023-06-02 10:26:17	2023-06-02 10:26:17	9	1	1	1	1	0	UiPath Malfunctioning code	Older Dependency	URL/Path access denied
2023-06-02 10:26:17	2023-06-02 10:26:17	9	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-05 14:55:41	2023-06-05 14:55:41	10	1	1	1	1	0	UiPath Malfunctioning code	Older Dependency	URL/Path access denied
2023-06-05 14:55:41	2023-06-05 14:55:41	10	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-05 14:55:41	2023-06-05 14:55:41	10	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-05 15:08:00	2023-06-05 15:08:00	12	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-05 15:08:00	2023-06-05 15:08:00	12	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-05 15:08:00	2023-06-05 15:08:00	12	1	1	1	1	0	Excel error	Older Dependency	Excel is unavailable or already running
2023-06-05 16:30:28	2023-06-05 16:30:28	12	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-05 16:30:28	2023-06-05 16:30:28	12	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-05 16:30:28	2023-06-05 16:30:28	12	1	1	1	1	0	Excel error	Older Dependency	Excel is unavailable or already running
2023-06-06 09:27:25	2023-06-06 09:27:25	10	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-06 09:27:25	2023-06-06 09:27:25	10	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-06 11:51:46	2023-06-06 11:51:46	12	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-06 11:51:46	2023-06-06 11:51:46	12	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-06 11:51:46	2023-06-06 11:51:46	12	1	1	1	1	0	Excel error	Older Dependency	Excel is unavailable or already running
2023-06-06 12:04:31	2023-06-06 12:04:31	10	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-06 12:04:31	2023-06-06 12:04:31	10	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number
2023-06-06 12:42:14	2023-06-06 12:42:14	10	1	1	1	1	0	Ui navigation error	Older Dependency	Wrong data input or not dynamic SAP/Web Selector
2023-06-06 12:42:14	2023-06-06 12:42:14	10	1	1	1	1	0	Maximum retry error	Older Dependency	Maximum retry number

Source: Own work

Given the data at hand, it is necessary to address why it might not be suitable for developing a classification model, specifically aligned with the goals of this project. Initially, the utilization of Python's library Pycaret was planned for its data preparation and model tuning capabilities, along with the possibility of exploring a variety of ML algorithms. The dataset uses error categories among other attributes listed in Table 2 as input variables, with the failure reason column as the target variable. While the number and quality of the input variable are essential to training a well-performing model (Sheikh, 2013), and in this scenario, the quality seems adequate, the dataset's size and type could present challenges. An introduction to the scenario of possible ML issues was also done in subsection 2.2.3. Lacity and Willcock's (2021) study research reports that adapting cognitive automation is more challenging than RPA due to the data availability factor, where the automation being deployed needs to train on large quantities of different types of structured data, as explained, "…needs thousands of accurately labeled training data examples to enable the machine-learning algorithms to reach an acceptable level of proficiency". This means that taking data availability, for instance, lack of quality and quantity of data available, can hinder accurate classification and anomaly detection. For automated analytics, this can mean the model cannot detect relevant patterns or potential noise in the data. Connected challenges include choosing the appropriate software integration that complies with standards and the risk of choosing the wrong ML model for a task.

To evaluate the overall impact of the model performance by taking a deeper look into the performance trade-offs in different classes. First is the challenge introduced by the failure reasons target variable, which includes over 20 distinct class labels. This creates additional

data availability, interpretability, and complexity problems. Data availability is limited as categorizing narrows the data pool for each class, undermining the model's reliability. The complexity due to a higher number of class labels increases training times and computational demands. At the same time, interpretability issues arise when more class labels complicate the understanding of the model's predictions. For example, a large confusion matrix, a detailed feature importance plot, or complex ROC curves can make it harder to draw actionable insights. Aggregating the class labels could simplify the model and improve its performance. However, from a business requirements standpoint, it results in more generalized failure reasons that offer the team little accountable benefit or productivity gains.

The second concern stems from a potential data imbalance. Imbalanced data occurs when a target class is underrepresented, skewing the model's understanding of its importance (Abbott, 2014). An examination of the error categories presented in Table 2 reveals a dominant distribution of specific class labels in the dataset, leading to bias toward these prevailing classes. This imbalance might additionally mislead evaluation metrics, such as higher accuracy levels than they are supposed to be, as the model correctly predicts the majority class, having been exposed to many examples of it and only a few examples from the minority class (Abbott, 2014). When presented as a percentage of the training dataset, the class distribution reveals that 29.59% of the data belongs to the first class and 27.15% to the second class. The following classes show a swift decline from 8% for the third class to just 0.19% for the last class. The expected results from an ML model from the skewed distribution hint at high-accuracy levels where the most successful models are expected to be the Extra Trees, Random Forest, and Gradient Boosting Classifier, with possible good results also being received from the Naive Bayes model.

There are different techniques to deal with imbalanced data, such as undersampling of the overrepresented class or class weighting that could help improve the model performance and make sure it accurately predicts the minority class as well. However, considering the current dataset's size, these methods could risk overfitting, where the model shows low precision with new data. That is why the third challenge is tied to the context of business requirements; in this case, with this dataset, deploying an ML model may not bring any additional advantages to the team beyond what can be achieved through semi-automated analytics. Utilizing descriptive analytics can unveil a similar understanding of the data, so if it is deemed in this case that the costs of balancing the data, training, and maintaining the model may not justify benefits over the insights already available from the more effective alternative. Semi-automated analytics already provides comprehensive and interactive analysis capabilities that meet business needs. After considering these factors during the data preparation phase, and taking into account the current expectations from the model and the level of preparedness, it was concluded that an ML approach would not be suitable for this situation.

4.2.4 Modeling

An in-depth understanding of the final features and their influencers was gained in the data preparation phase during the discussion of the quantitative metrics that are subject to analysis and the qualitative attributes that can be used to segment the data. Considering the analysis types defined in section 2.2.2, it is natural to proceed with the modeling phase. However, due to the described strategic decision to withdraw from an ML model, the focus is solely on semi-automated analysis. This analytical approach is dedicated to understanding the underlying phenomenon or process, rather than estimating values, focusing on extracting knowledge on what has happened (Provost & Fawcett, 2013). This entails gathering a detailed overview of the dataset, relating the purpose of each visualization, and describing the creation process.

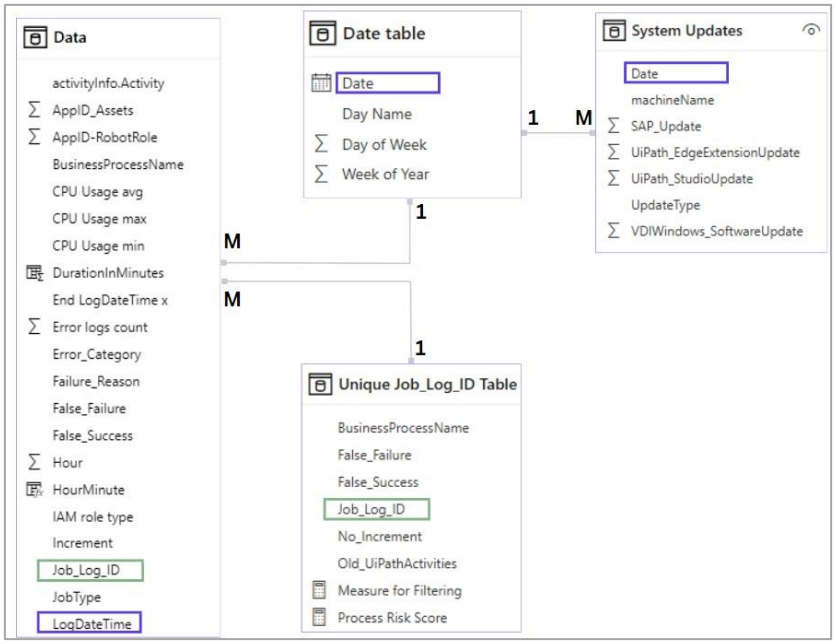
Defining the purpose behind each visualization type is important, as this aligns the data exploration outcomes with the planned strategic objectives. In the context of evaluating the testing environment setup, line charts are used for trend analysis, capturing the dynamics of error categories and failure reasons over time. Stacked column charts facilitate a comparative analysis by process name, illustrating how each process contributes to the overall error metrics. Pie charts portray the composition of error categories within the entire dataset, helping assess which category needs further exploration. One way this is enabled is by designing each visualization to support interactivity, using tools such as influential slicers or drill-down options to explore different facets of the data.

In the data model, a one-to-many relationship is established between the 'Data' table, depicted in Figure 11, and both the 'Unique job_log_id' and 'Date' tables. The 'Unique job_log_id' table consists of individual process jobs with additional features added. Meanwhile, the 'Date' table is structured to extract attributes from date columns, creating new features such as day of the week, month, and week of the year imperative to analyze seasonal trends throughout. A one-to-many relationship indicates that a single record in one table can relate to multiple records in another. For instance, a unique job identified by the 'job_log_id' feature in the 'Unique job_log_id' table may be associated with several error records in the 'Data' table, where initially, there could be more errors in a single job with the same id. Similarly, a single date entry in the 'Date' table can correspond to multiple records in the 'Data' table, reflecting all the activities or errors that occurred on that specific date. Both the 'Unique job_log_id' table and the 'Date' table also share a one-to-many relationship with the 'System updates' table, which is mentioned in subsection 4.2.2 and portrays the updated dates. A visual representation of these relationships is provided in Figure 17.

Calculations and aggregations performed to further customize the dataset are described next. First, the focus is on the date columns of the 'Data' table. The 'total_execution time' column, originally expressed as a fraction of an hour, this decimal representation of time was transformed into the 'hh:mm:ss' format for a more intuitive display of the time duration. Additionally, an 'Hour' column was extracted from the log tracking data, introducing a finer

level of granularity and enabling correlations with other variables as well as comparisons within planned trend analyses. In addition to these transformations, continuous data was segmented into bins to facilitate easier interpretation.

Figure 17: Data Model Relationships

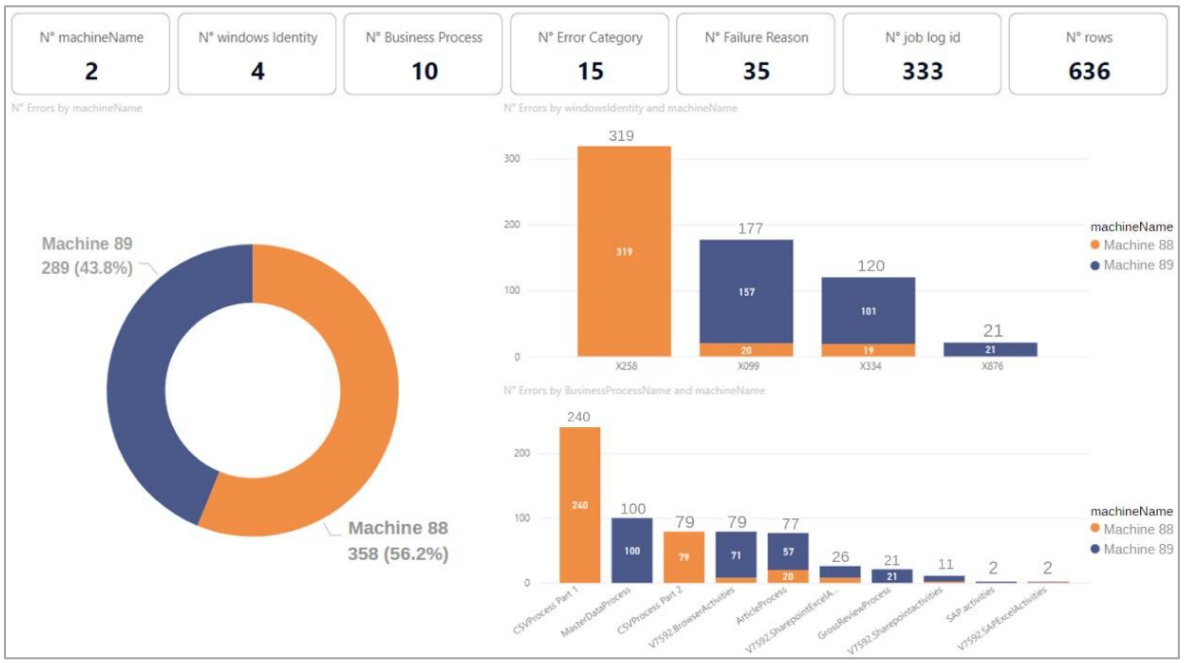


Source: Own work

This involved grouping data points for CPU usage into minimum, maximum, and average category values. New summary metrics were also created, comprising the average execution time, average error rate, and percentages of false failure, and success, which reflect discrepancies with the actual job execution outcomes (detailed in Appendix 5). Another metric was the process risk score, a weighted sum of the binary columns selected as risk indicators. The importance levels were based on business process logic, where the aggregate was the overall risk within the processes. The DAX formulas employed for these columns and measures are described in Appendix 7, providing an enhanced examination of the steps outlined.

The creation process of each visualization is documented with an emphasis on the technical aspect. The first visualization, portrayed in Figure 18, is designed to focus on the frequency of machine usage and the potential errors tied to each machine's setup. It aims to investigate how error distributions vary across different business processes and Windows identities, thereby conveying potential operational stability issues of each machine. This visualization, along with subsequent ones, features cross-filtering of the elements within, allowing users to filter based on selected criteria and examine associated outcomes.

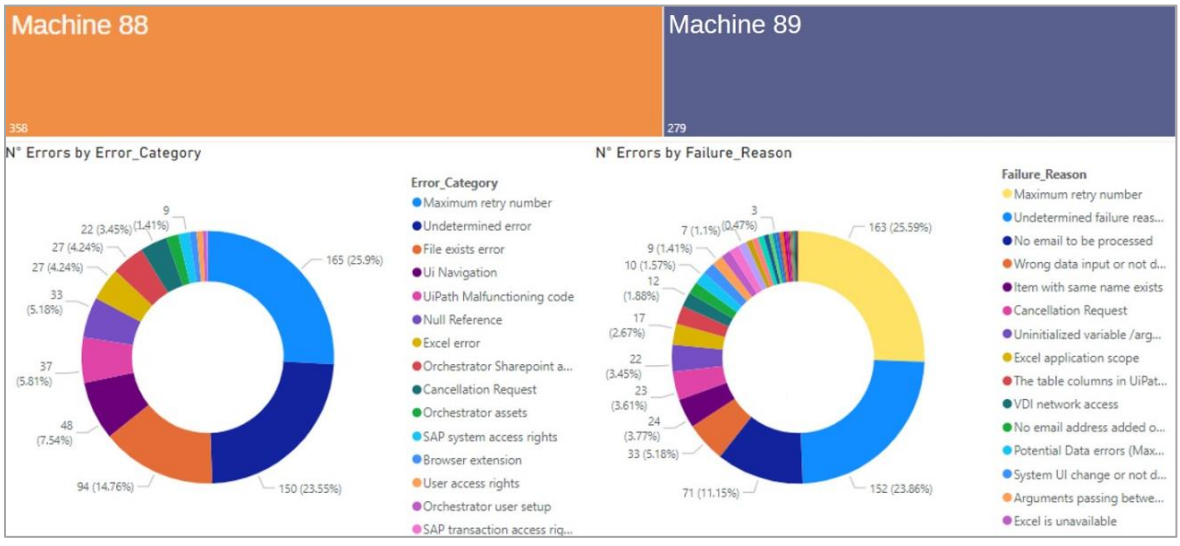
Figure 18: Error Distribution by Machine



Source: Own work

The second visualization, depicted in Figure 19, delves into the distribution of counted errors across categories, alongside their respective groups of failure reasons. Each pie chart is additionally labeled with the names of the variables to provide more details.

Figure 19: Error Categories, Failure Reasons, and Secondary Causes by Machine

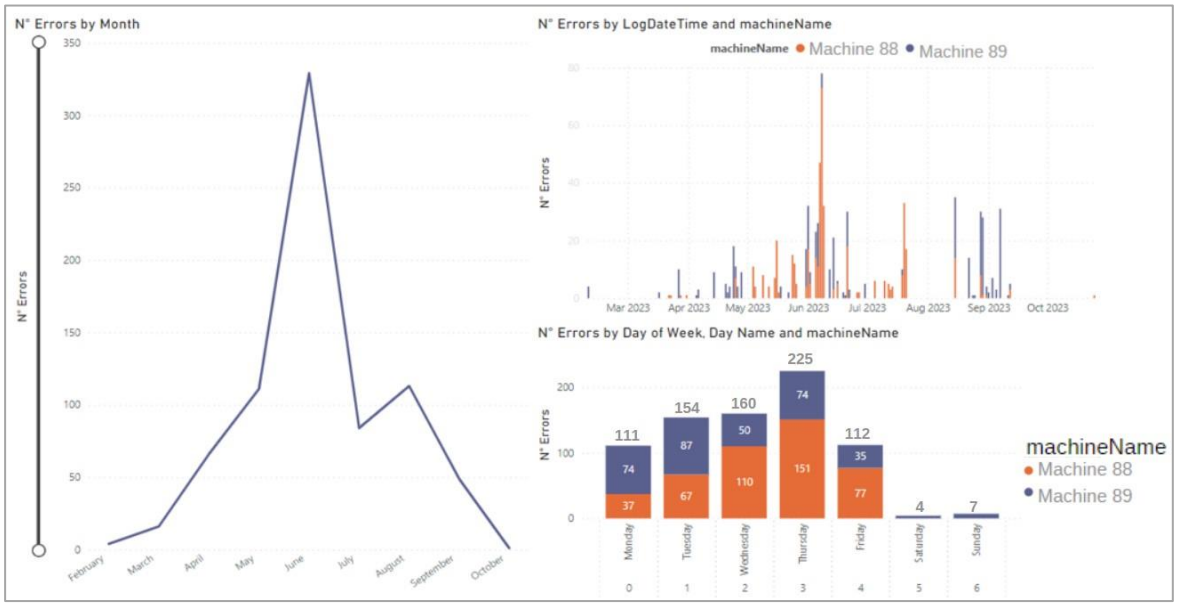


Source: Own work

In addition to associating the variable names, the second pie chart can drill down to uncover secondary causes behind the failures as well. Including secondary causes is informative; they are not necessarily the primary issue, but their presence indicates additional layers of

information regarding failures that merit further exploration. The third visualization focuses on the time aspect, incorporating the 'Date' table to explore the error distribution dynamics on a monthly and weekly basis, as illustrated in Figure 20, still highlighting differences between the two machines. Notably, the visualization offers drill-down capabilities, displaying the daily occurrence within each month. A pronounced spike in errors was observed in June, creating a lot of space to evaluate the causes behind this unexpected discrepancy.

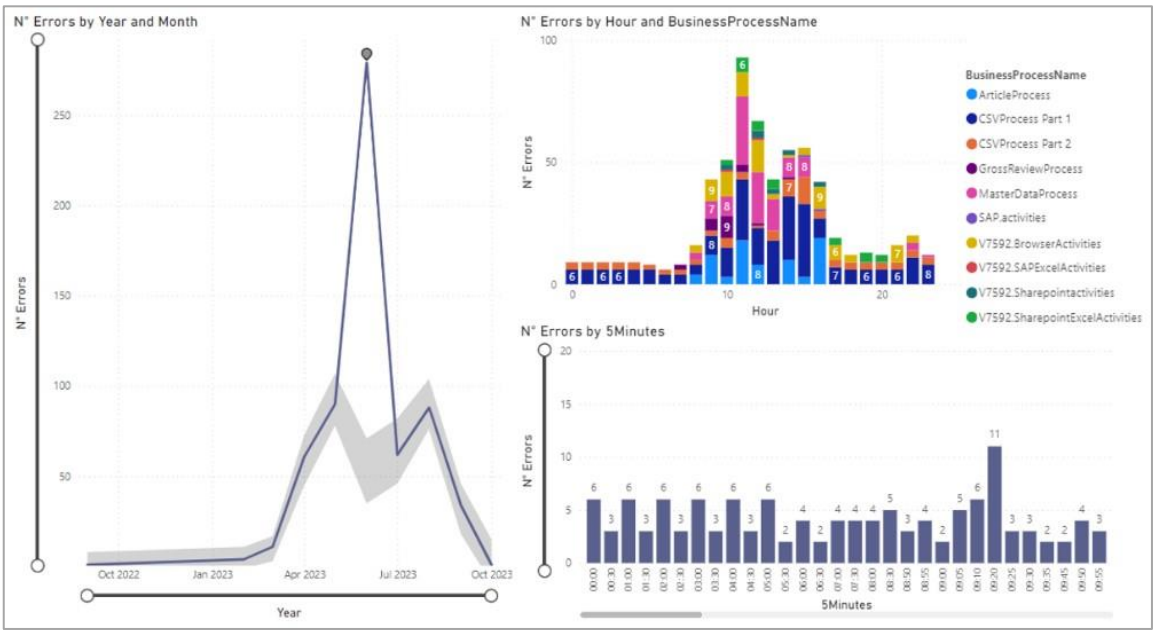
Figure 20: Error Distribution by Date



Source: Own work

The fourth visualization continues the focus on temporal analysis, concentrating on an hourly perspective. Figure 21 represents the distribution of errors throughout the day, offering insights into the time when business processes are more prone to errors. By utilizing a line chart with anomaly detection enabled, the visualization helps not only track the frequency of errors by time but also identifies the expected minimum and maximum error ranges for the period. This feature aids in identifying deviations from the norm.

Figure 21: Daily Error Frequency



Source: Own work

4.2.5 Evaluation

The purpose of the evaluation phase is to explore the practical implications of the visualizations, following the technical documentation in the modeling phase. The evaluation assesses the effectiveness of the visual representations, which can give significant additional meaning and drive improvements. Moreover, it prioritizes ensuring that the visualizations are comprehensible to stakeholders, as they are the ones who ultimately approve the project and whose satisfaction with the outcome is essential (Provost & Fawcett, 2013).

The first dashboard presented in subsection 4.2.4, depicted in Figure 18, is an analysis of errors in relation to Windows identities and machines, enabling the RPA and functional teams, serving as the primary stakeholders, to understand more the robustness of the features and what might require further attention. For that purpose, the 'No error' group is removed here. Machine 88 is predominately used for testing when we look at the usage per 'job_log_id'. In this instance, it also faces more errors as well, which is parallel, while filtering by this machine reveals that the robot user with Windows identity X258 has a vastly higher number of errors than its counterparts. Additionally, the business process name indicates that this robot user operates with the CSVProcess parts 1 and 2 processes, suggesting that these processes are tested more frequently and are more susceptible to errors. Meanwhile, Machine 89 displays a more evenly distributed error count across processes, yet the MasterDataProcess stands out with 100 errors. A detailed breakdown of the errors can be further explored in the following dashboard, which was introduced in Figure 19.

Understanding errors, their underlying reasons, and secondary causes can influence the decision-making process and drive operational improvements. By filtering again based on machine names, the dominant errors for Machine 88 are displayed; the most prevalent is the 'maximum retry number' at 29.33% and the 'file exists error' at 20.67%. While the 'undetermined error' and 'no error' categories are in between, they do not contribute in this context. The team can observe specific challenges within different processes, consequently easily targeting changes to reduce error rates. Pursuing the analysis of CSVProcess parts 1 and 2, with an added process name filter, shows that 'maximum retry number' again is the dominant issue at 31.66%. This confirms firstly CSVProcess as the leading contributor for the overall high percentage and secondly that the failures are mainly due to three reasons: reaching the maximum retry limit without encountering other errors in the process, cancellation, or issues with dynamic web elements. The latter, as explained in subsection 4.1 of this chapter, belongs to the UI navigation error and leads to failures due to web page loading time, nested elements, or pop-ups.

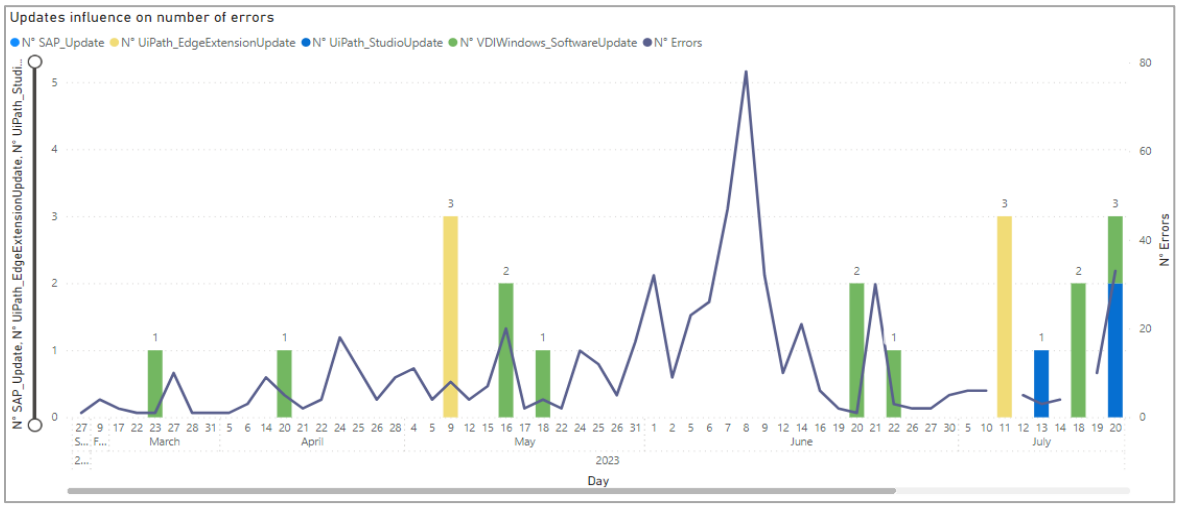
Moreover, applying the filter for the MasterDataProcess reveals additional errors, including 14% categorized as 'UiPath Malfunctioning code'. This is mostly caused by problems with table column handling, such as incorrect naming or length discrepancies, and with string manipulation, including misplaced functions like trim, substring, and split. To address these code-related issues, the RPA team can institute more rigorous code reviews and plan collaborative workstations where participants can share best design practices. Additionally, optimizing the UiPath framework, if issues stem from its setup, together with providing more precise fault documentation could constructively address them, leading to more detailed visualizations that spotlight recurring issues that require attention. The anticipated success of this initiative is to increase the confidence of the functional teams for RPA through a commitment to continuous improvement, which is quite an important step. In response to a question about ways to improve RPA process flows, the RPA consultant in the interview underlined key areas of focus: “It is essential to have stable robots so that failure becomes more of an exception instead of the rule. Further, IT consultants need to be informed about which processes are RPA ones and will therefore fail if there are any changes in the system. Lastly, when possible, avoid the use of outdated activities, such as those associated with Excel, since they face issues more regularly.”

Revisiting Figure 20, the considerable surge in error counts during June requires an analysis of specific events, changes in machine usage frequency, update influence, or other factors that could explain the anomaly. The already established higher error frequency in Machine 88, particularly with the CSVProcess accounting for the majority, 319 out of 358 errors, partially explains this trend. Drilling down further helps conclude that most of the errors occurred between June 1st and June 10th. A generalized examination of usage frequency suggests that the increase was likely attributed to a new process schedule being initiated in June, with triggers set every 30 minutes. Additionally, a manually thrown exception, used for testing purposes and contributing to 60 errors classified as undetermined, influenced the

error count. Adjustments in the following months, specifically the decision to no longer classify the mentioned manual exception as an error (as it was showing that there was no more data to process), resulted in an error count reduction for the process. Furthermore, the visualization shows that Tuesdays and Thursdays experience a higher error rate compared to other days of the week, which can be reasonably linked to the scheduling of certain processes on these days.

The visualization portrayed in Figure 22 was created using the 'System Updates' table to investigate the impact of system updates on failures. It categorizes the updates into three types observed during this period: Edge UiPath extension update, VDI Windows update, and UiPath Studio update. Visualization analysis reveals that Edge UiPath extension updates do not noticeably affect error rates; in contrast, VDI Windows updates are shown to have a substantial influence, as seen in the error increase following each monthly update. This is due to alterations in dependencies, resets of the permissions requiring adjustments, and changes in network settings that influence connectivity. However, to pinpoint a specific cause, a more intricate exploration needs to be done, focusing on the dates, processes, and types of activities most influenced. For instance, analyzing May 16th**,** which affected Machine 88 and CSVProcess part 1, can provide the time the errors happened and the nature of the errors. Moreover, the increase on August 15th happened across both machines and impacted three different processes, so this gives a reason to look into it further. Identifying the RPA processes with a history of being negatively impacted by updates could also guide targeted adjustments to minimize the issues in a timely manner, keeping in mind the external factors.

Figure 22: Impact of Updates on Error Frequency

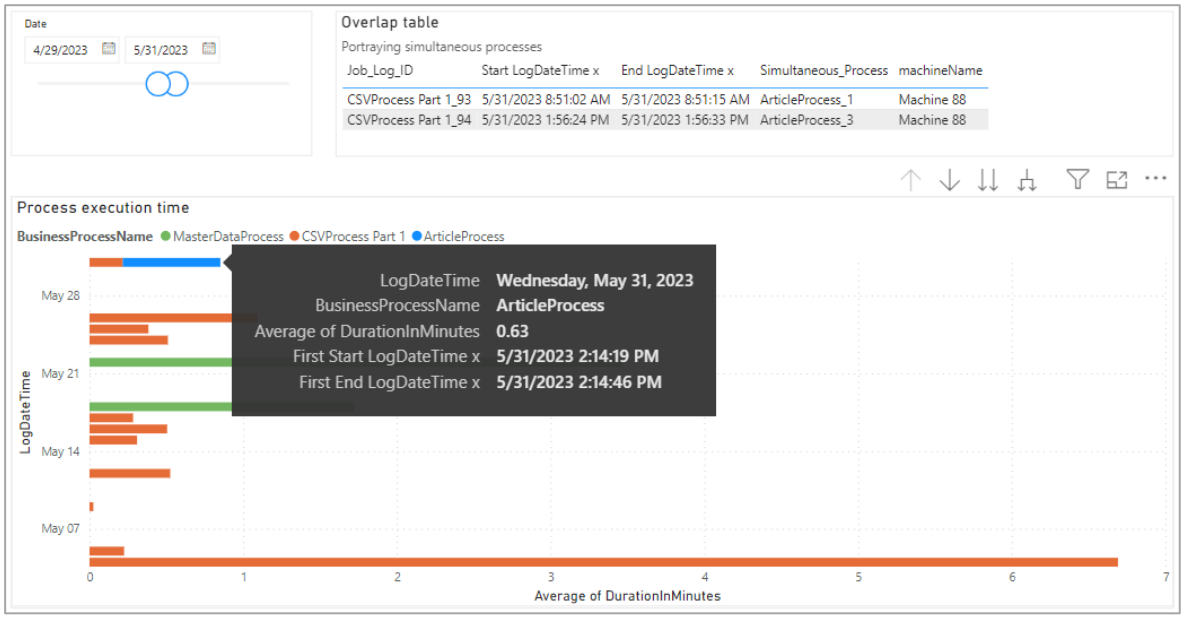


Source: Own work

Supplementing the overview of the time variable in Figure 21, an important point to consider is the correlation between job execution times and the timing of job triggers, particularly concerning issues of trying to start processes simultaneously or producing errors when

starting a process on a machine that has already one running. Given that only one process can run at a time in the same environment, this correlation emphasizes the importance of correct scheduling and resource allocation, which sometimes can be challenging. Using the 'simultaneous_process' column in this visualization, illustrated in Figure 23, shows the two processes found to run concurrently.

Figure 23: Simultaneous Process Execution



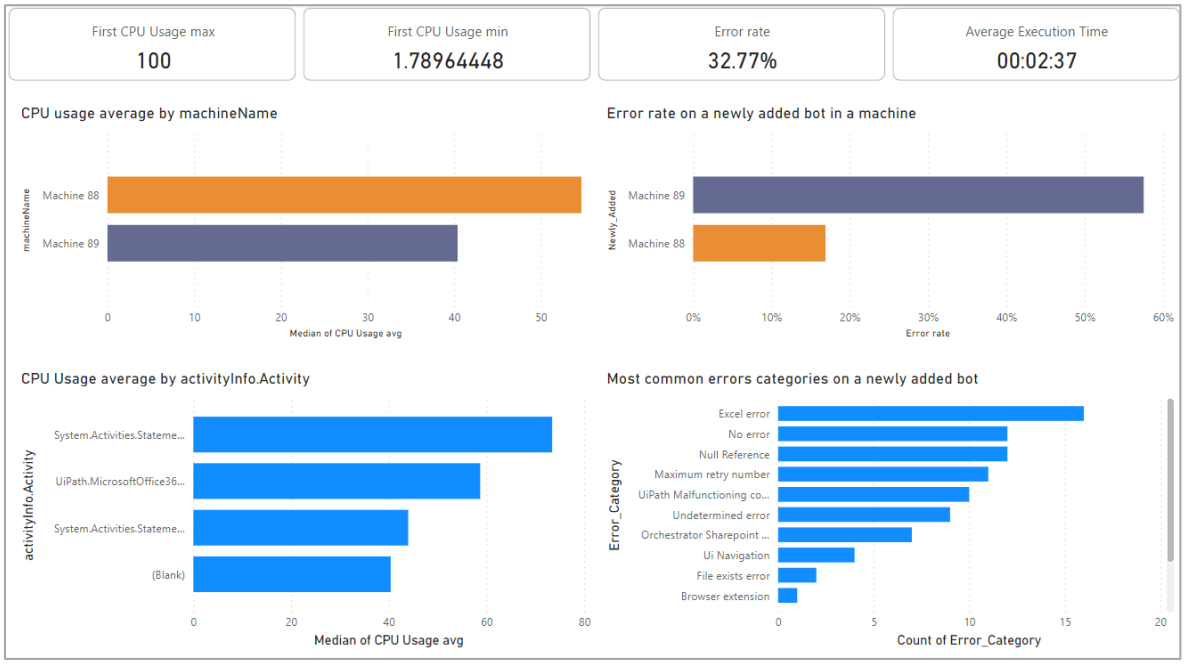
Source: Own work

By providing an overview of the process execution time, complete with tooltips, the visualization enables a detailed examination of how many processes were executed on a specific day and the timeframes of their operation. Being able to see this information can help the team find overlaps more easily in the process of adjusting an effective schedule, prepare appropriate triggers on Orchestrator by also being aware of how much of the day is available in a specific machine, and correlate the execution dates to the error distribution in Figure 15.

The integration of data aggregates, such as average execution time, error rate, and process risk presented in Figure 24 and Figure 25, assists in answering additional questions regarding the data. Starting with Figure 24, the visualization indicates that Machine 88 exhibits a higher CPU usage compared to Machine 89, suggesting that processes on the first machine may demand more computational resources or involve more resource-intensive activities. Filtering further detects the two SharePoint activity processes as the leading cause of elevated CPU usage. The analysis aligns with the discovery that in these two processes, 53.84% of errors are associated with SharePoint activities or the installation of Microsoft 365 process packages for Orchestrator, pinpointing specific areas for potential optimization. Exploring the CPU usage by UiPath Studio activities reveals that sequences account for the

highest usage, which is expected because most of the sequences have complex calculations and incorporate various other activity types, consequently requiring more processing power. Microsoft Office 365 activities, utilized in SharePoint processes, have the second-highest CPU usage. This adds to the initial observation of Machine 88 and offers further context on why certain processes require higher computational effort.

Figure 24: CPU Usage, Error Rates, and New Process Errors



Source: Own work

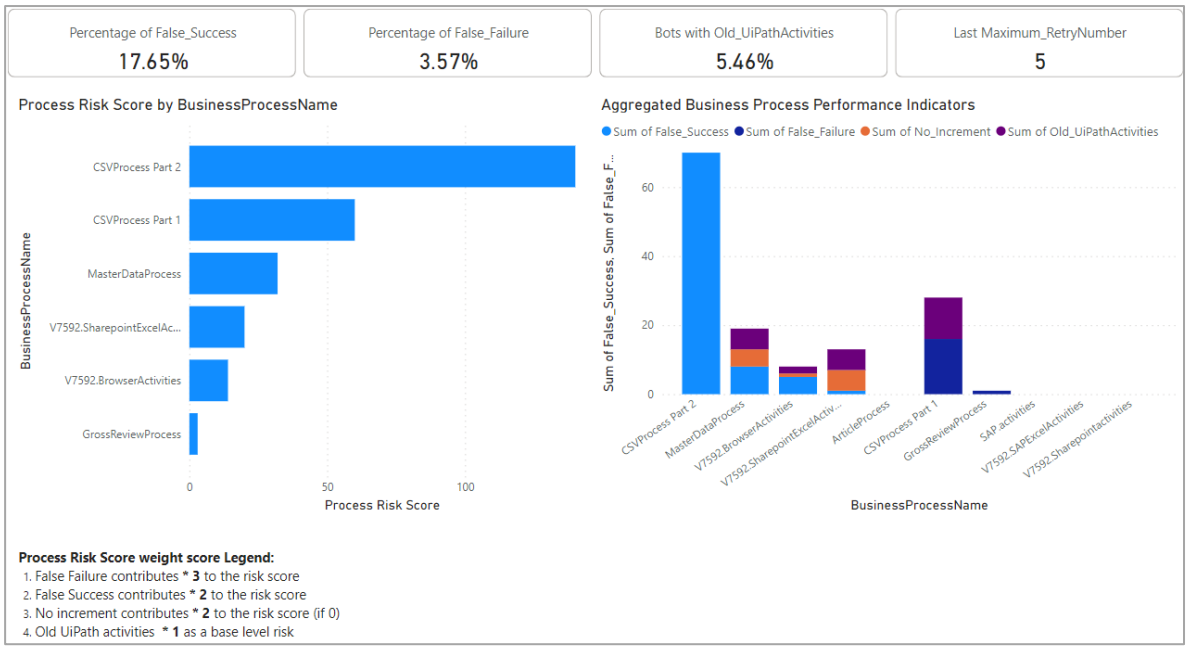
Excel errors are recognized as significant when answering the question regarding the most common failures after introducing a new process, meaning the 'newly_added' column is marked as true (detailed in Appendix 5). This may explain the RPA team's preference for minimizing Excel activities and utilizing SharePoint more, despite its own acknowledged set of challenges.

Figure 25 depicts the process risk, incorporating binary variables and their weighted score into the overall risk score (detailed in Appendix 7). CSVProcess parts 1 and 2 are the highest risk contributors, explained by their highest risk of false success.

Subsequently, the MasterDataProcess, BrowserActivities, and SharePoint Excel Activities processes have a combination of challenges including false success statuses at the end of the job execution, usage of outdated activities, and lack of transaction increments. This combination suggests that the process dependencies and, more importantly, logic issues should be investigated further. Such issues can result in longer process running times, excessive retry attempts, or problems with setting up the transaction status as an essential part of the framework. The secondary causes for the three selected processes unveil that 24.88% of the causes suggest checking the processes using an older dependency version. In

contrast, 17.07% relate to verifying correct UiPath ReFramework error handling and ensuring page loading accuracy. Some of the steps to minimize these risk indicators are implementing helpful measures like enhanced log details, appropriately named activities, and updated activity versions.

Figure 25: Process Risk Aggregation



Source: Own work

4.2.6 Deployment

The Power BI dashboards are primarily designed for the RPA team. However, if necessary, operational management and other decision-makers can gather information on the success of the RPA processes and make strategic decisions based on that. Access provided to the users enables them to interact with the dashboards. The analysis results assist the team in addressing real and recurring issues that impact the processes. Identifying the most common sources of failure reasons can provide a clear view into the primary factors that impact the processes most and the trends over time, helping allocate resources for those areas.

To do this successfully, thorough documentation of the data preparation process and training for all involved users should be provided. The semi-automated process can be configured to gather data over three months, aligning with quarterly team retrospectives' cadence. Further, it is also possible to automatically email the dashboard insights to stakeholders at a similar timeframe, thus informing them of the ongoing improvements of the RPA processes.

Regardless of whether deployment is successful, the process should be flexible to circle back to the Business Understanding phase since the next iteration can generate an improved

solution. Deliberating on business, data, and performance goals sparks new ideas for enhanced business outcomes (Provost & Fawcett, 2013).

5 DISCUSSION OF FINDINGS

5.1 Comparative Analysis of the Two Approaches

Three main questions were investigated during the research process of improved RPA performance through data analytics based on theoretical knowledge and practical analysis with an influence of the CRISP-DM methodology.

The **first question** concentrated on the benefits and limitations of implementing data analytics in the company's RPA processes. The answer to the research question addresses the objective of initially investigating the benefits of integrating data analytics into the RPA department, which is significant and includes informed decision-making that leads to improved operational workflows. This integration ensures a more comprehensive overview, making decisions that align more closely with the performance data. The deployment of dashboards is expected to influence the end-to-end process portrayed in Figure 6 positively. Explicitly during maintenance, it facilitates a better utilized monitoring step, within a predefined timeframe, that adds to the decision-making.

Furthermore, the benefits include addressing the roots of frequent errors by identifying patterns in the data, leading to expected RPA process optimization and enhanced system reliability and efficiency. Improvements were evident during the evaluation phase, which detailed the challenges faced connected to their potential causes, with enhanced system reliability being a primary outcome of reduced failures. The reliability allows for greater confidence among functional teams in entrusting the automation of their processes to RPA. This is further confirmed by the theoretical contribution on the topic that establishes stakeholder trust as one of the leading factors in successful implementation. The limitations encountered, both expected and unexpected, are elaborated in Chapter 6.

The **second question** investigated which of the approaches, automated or semi-automated, would detect RPA process anomalies more effectively. Through data preparation, it became evident that using algorithms and an ML model for anomaly detection, given the purpose and dataset, is expected to lack accuracy, particularly when dealing with an array of issue associations. Suppose the team agrees to go forward with it. In that case, the dataset can benefit from further refinement while the model from additional training and testing, as well as a more meticulous examination of the data inputs. Other things to consider, as examined during data preparation, involve potentially aggregating distinct labels further. If having larger groups still steers toward reaching the business requirements after careful evaluation of balancing simplification with the loss of information. On the other hand, the semi-automated detection, involving the automated RPA data extraction and human overview,

demonstrated many advantages in terms of effectiveness and nuanced understanding of the data, since anomalies are more accurately interpreted.

The **third question** focused on the long-term critical factors necessary for leveraging RPA process optimization. Some of these factors include an agreement with the importance of change management, also highlighted during theoretical research, that would support stakeholders in adapting to the new processes, thereby leveraging new knowledge in the most advantageous way possible. Embracing change involves improved communication and feedback mechanisms during the transition period, as adaptations are expected, and the necessity for regular updates in response to evolving business needs. The essential components of this question are explored in the subsequent subsection 5.2.

During the **case analysis**, accessible company data from the RPA department was preprocessed and analyzed before examining the findings. These findings include an emphasis on how data-driven analysis aids in a better understanding of system usage, error patterns, and inefficiencies. The outcome was practical PowerBI dashboards, providing descriptive analytics to foster RPA process improvement. The dashboards provide a centralized platform, ensuring RPA team members are presently aware of errors faced, thus reducing potential knowledge silos and reaching a proactive approach to problem-solving. This shift leads to increased productivity for the team, allowing them to directly address issues, focus on productive tasks, and reduce the time spent bug-fixing. The insights obtained from the analysis revealed specific error patterns, considering CSVProcess, MasterDataProcess, and Machine 88 as notable starting points for addressing issues. A crucial connection was also identified between scheduling patterns and the occurrence of errors, by evaluating trends throughout the months and uncovering underlying influences such as simultaneous process overlaps, update days, and maintenance overlaps during that period. Furthermore, the evaluation shed light on the causes of increased CPU usage, particularly in the utilization of SharePoint activities and Microsoft 365 process packages, guiding potential areas for optimization.

Some of the important considerations to achieve meaningful results from data analytics involvement include:

- Make sure the process for data access is straightforward and regular reviews for permissions access prevent exceeding the expiration period.
- Identify the correct metrics that match directly with the precise error reduction goal.
- Establish clear rules for the rule-based RCA that reflect best the common errors encountered, making certain to produce a dataset of sufficient quality and size.
- After the analysis outcomes, warrant learning from failure groups and use the results to precisely improve the processes.
- Implement productive communication and documentation requirements to increase stakeholder trust. Providing transparent access to dashboards and outlining actionable steps for improvement.

- Measure performance over time, making it easier to implement changes and adopt a flexible approach to the CRISP-DM methodology flow, where it is encouraged to track and improve the processes continuously.

During the **theoretical contribution**, comparable research indicates the importance of automating a suitable process, where otherwise it would lead to inefficiencies and additional costs and resources to adapt a new helpful technique to improve the process. The comparative analysis of RPA and BPM elucidated the role of aligning stakeholders' needs and robot capabilities to improve the end-to-end process, increasing accuracy in repetitive tasks. The theoretical research on social/implementation confirmed the final considerations of the importance of establishing governance, communication, and following company standards. These elements were further corroborated by interview responses, during which interviewees described challenges in the absence of such an organization and the need for a centralized platform.

Furthermore, the importance of data analytics in the RPA lifecycle was also a focal point in the case analysis and theoretical discussion. In this instance, notes can be taken from the research that confirms the practical results, specifies the change management factors, and offers data analytics benefits for RPA, like operational visibility and anomaly detection.

5.2 Recommendations for Successful RPA Monitoring and Error Handling

Additional columns and data sources can enhance the failure reason detection ability. However, before integrating them two factors should be assessed: the capability of adding new information, considering access rights and security concerns, and the feasibility of correctly processing and maintaining it. This ensures that the new data is easily accessed, does not complicate the maintenance process, and contributes valuable insights. There are several potential increments to consider once it is confirmed that the integration of the new data is fitting. Including information from JIRA tickets related to bug-fixing can offer comprehension into the severity levels, solution strategies, parties involved, resolution timelines, and satisfaction levels with the outcomes from the functional team. Furthermore, incorporating PDDs and other useful process documentation can explain the expected process execution and provide a better understanding of the process complexity. It can be beneficial to incorporate features like performance data from external systems that RPA interacts with that cause failure in the processes and data on changes within the RPA environment, like the already existing dependency versions. This can include process workflow changes, process selector configurations, proxy network connections, and others to understand the modifications' impact.

The monitoring and error-handling strategy should prioritize adaptability to UI changes in the applications from which the robot extracts data. While most of the steps are straightforward, manual adjustments to the RPA process to make it stable again can be necessary, alongside manual execution of the data as a temporary solution. To reduce this

risk, at least associated with the logs data, direct log sources from the machine environment can be appraised. This requires further examination because it eliminates the need to extract data from Orchestrator and Elastic but, at the same requires a longer filtering time to collect the data, especially for a sizable dataset, and adjustment of the data preprocessing steps to fit the format. Maintenance reviews of the existing Python commands include updates to the error categorization dictionary, either in keywords or in error category groups, and adjustments to the rule-based RCA criteria for creating failure reasons. Moreover, infrastructure changes, such as a new machine being used, require command function modification. Finally, as seen in the list of considerations, it is vital to have an outlined plan for updating the dashboards to include the additional data, ensuring strategic improvement with the evolving business needs.

The examination reveals that the semi-automated process can be leveraged better long term. This advantage was observed through the data extraction steps and aggregation required for dataset preparation, alongside the management of changes and ensuring the initial project goals were met.

6 CONCLUSION

In assessing whether the goals of the thesis research were met, an overview is outlined to determine the extent of the alignment.

The **first goal** of this thesis was to investigate the most common causes of robot failures within the company's context and to identify the benefits of process monitoring. The goal was realized by delving into the literature that concentrated on the challenges of robot failures. This investigation specifically went into the social/implementation and technological categories. The insights gained proved useful in combination with the expert interviews conducted internally and outside the company. The interviewees shared valuable information on data categorization, relying on their extensive experiences and expertise in the field, relating it directly to the company's needs. The discussions, therefore, determined the most common causes for failure. For the monitoring part of the first goal, the benefits were presented as offering a more effective means to conduct root cause analysis, which is helpful for this thesis. Further, they were proposed to support automation and identify metrics that share necessities for process improvements. The approach is supported by the CRISP-DM methodology, as seen from its iterative and cyclical nature, complying with potential standards. Sharing these benefits is important from a company perspective, especially considering that the team still needs to fully leverage its advantages in analyzing the reasons for the failure and success of process execution and tracking changes directly.

The **second goal** was to implement data analytics and examine the potential of using it for RPA's performance improvement through error identification. This requirement was done

during the analysis process composed of six phases. The focused examination of the tool's potential occurred predominantly during the evaluation phase.

The **third goal** was to use the study results to successfully mitigate the analyzed failures in the future. This goal represents a significant part of the problem the thesis is trying to solve, and reaching it depends on the error categories and secondary causes, along with other critical features of the dataset that encapsulate the context of log messages and reference files, thereby explicitly recognizing the failure reasons. By providing visibility into the frequency and nature of anticipated errors, the team can make informed decisions, prioritizing their efforts constructively to the most high-impact failures and allocating resources correctly.

Focusing on user adaptation is also an important facet of this goal; it includes steps like training, feedback, change management, adaptation support tools, and monitoring. Its emphasis is covered during the interview discussions, which concluded that actively communicating encountered issues and their resolutions keeps the whole team up to date about what is happening with the processes. Thus, leveraging the study results needs to be done beyond analyzing historic error and failure data that can identify and address systemic issues within their RPA processes, but knowing that information, to strive for creating resolution techniques and best practices that lead the team forward in anticipation of the failure numbers to keep reducing. Moreover, continuously adding keywords from log messages significantly enriches the dataset, expanding the range of identifiable error categories. This process reduces the number of unknown errors and simplifies the process of understanding the implications of each error. Consequently, as more error category groups are integrated into the analysis, the team gains a more profound understanding of the RPA process landscape. This enhanced insight simplifies error tracking and resolution, fostering a more informed and responsive team environment.

There were **limitations** of the research and unexpected challenges identified. A primary issue faced was related to data availability, it was one of the reasons necessitating the creation of test cases due to the inability to directly use production data, in adherence to data confidentiality policies. Several subfactors were created from this issue.

Firstly, the creation of test cases required time investment, both in terms of development and preparing the machine applications and robot user to have all the necessary access rights.

Secondly, misrepresentation of data between the two applications, Orchestrator and Elastic, was observed during the data extraction process. There were instances where jobs were recorded in Orchestrator, but Elastic had no corresponding logs for that job, and vice versa. This misalignment was particularly notable in cases where jobs stopped or failed before producing any logs. Despite this, such instances proved to be informative, as they could be accurately classified under the stopped end-state status. This categorization aids in identifying errors related to dependency installation or asset access issues within the testing

or production environments. Additionally, it provides valuable insights into processes initiated while another was running, seen under ‘simultaneous_process’ (detailed in Figure 23), highlighting scenarios where jobs run for only a few seconds without reaching the initialization stage. The second issue had a bigger impact due to Orchestrator’s configurable log retention policy, which varied across different processes. This inconsistency led to the deletion of executed job information for certain processes, making data from the logs associated with them unusable. However, this challenge is deemed manageable, with the understanding that, during production data extraction there is a uniform log retention time frame across all processes. Given the plan to extract data over three months, removing jobs and logs is expected to be feasible.

Another area for improvement between the expected and actual outcomes stems from the feasibility of applying the classification ML model, which is connected to the size and structure of the available data. The unexpected finding during this research underscores the importance of data readiness and suitability for advanced analytical techniques. Addressing these challenges and adapting the research approach in response to such unforeseen obstacles highlights the dynamic nature of the research process and the necessity for flexibility and innovation in problem-solving.

Future research in the domain of RPA process analytics, particularly in minimizing errors and failures through Power BI visualizations, and incorporating additional features and data sources as discussed in subsection 5.2, marks a step towards a more detailed analysis of failure reasons. These categories can potentially be organized into tags within JIRA tickets if extracting this information from the application proves beneficial.

Going beyond the inclusion of new features, exploring how other environmental changes affect RPA performance and stability - such as Edge guidelines, organization settings, login types, security policies, and compliance metrics - can lead to the development of RPA systems that are more resilient to failures caused by external changes, thereby aiding risk management. Furthermore, investigating the influence of compliance requirements on RPA design and operation, as well as utilizing RPA to enhance compliance through automated controls and reporting, can provide valuable guidance for organizations striving to balance operational efficiency with regulatory adherence.

Finally, focusing on the unexpected outcomes and further investigating the integration of predictive analytics into PowerBI visuals to forecast errors, alongside prescriptive analytics for recommending corrective actions, is an optimal approach. This can extend to analyzing images and videos of the process during testing to compare successful output against potential errors, facilitating error identification directly. Researching the creation of real-time dashboards in PowerBI that provide immediate insights into RPA performance can also enhance monitoring and management practices.

REFERENCE LIST

1. Abu Sulayman, I. I. M., & Ouda, A. (2018). Data analytics methods for anomaly detection: Evolution and recommendations. In *2018 International Conference on Signal Processing and Information Security (ICSPIS)* (pp. 1-4). Dubai, United Arab Emirates. https://doi.org/10.1109/CSPIS.2018.8642713
2. Abbott, D. (2014). *Applied Predictive Analytics : Principles and Techniques for the professional data Analyst.* (1st ed.). Wiley.
3. Accenture. (2016). *Getting Robots Right: How to avoid the SIX most damaging mistakes in scaling up Robotic Process Automation*. https://dokumen.tips/documents/getting-robots-right-accenture-getting-robots-right-how-to-avoid-the-six-most.html?page=12
4. Agaton, B., & Swedberg, G. (2016). *Evaluating and Developing Methods to Assess Business Process Suitability for Robotic Process Automation – A Design Research Approach* (master's thesis). University of Gothenburg.
5. Alberth, M., & Mattern, M. (2017). Understanding robotic process automation (RPA). *The Capco Institute Journal of Financial Transformation*, *46*, 54–61.
6. Alpaydin, E. (2014). *Introduction to machine learning*. (4th ed.). MIT Press.
7. Asatiani, A., Copeland, O., & Penttinen, E. (2023). Deciding on the robotic process automation operating model: A checklist for RPA managers. *Business Horizons, 66*(1), 109-121. https://doi.org/10.1016/j.bushor.2022.03.004
8. Axmann, B., & Harmoko, H. (2020). Robotic Process Automation: An overview and comparison to other technology in Industry 4.0. In *10th International Conference on Advanced Computer Information Technologies (ACIT*) (pp. 559-562). Deggendorf, Germany. https://doi.org/10.1109/acit49673.2020.9208907
9. Borkowski, M., Fdhila, W., Nardelli, M., Rinderle-Ma, S., & Schulte, S. (2019). Event-based failure prediction in Distributed Business Processes. *Information Systems, 81*, 220–235. https://doi.org/10.1016/j.is.2017.12.005
10. CapGemini Consulting. (2016). *Robotic Process Automation-Robots conquer business processes in back offices.* https://www.capgemini.com/consulting-de/wp-content/uploads/sites/32/2017/08/robotic-process-automation-study.pdf
11. Chakraborti, T., Isahagian, V., Khalaf, R., Khazaeni, Y., Muthusamy, V., Rizk, Y., & Unuvar, M. (2020). From robotic process automation to intelligent process automation. In *Lecture Notes in Business Information Processing*, (pp. 215–228). https://doi.org/10.1007/978-3-030-58779-6_15
12. Chicco, D., Oneto, L., & Tavazzi, E. (2022). Eleven quick tips for data cleaning and feature engineering. *PLOS Computational Biology*, *18*(12), e1010718. https://doi.org/10.1371/journal.pcbi.1010718
13. Cuesta, H., & Kumar, S. (2016). *Practical Data Analysis.* (2nd ed.). Packt Publishing.
14. Davenport, T. H. (2009, November 1). Make better decisions. *Harvard Business Review*. https://hbr.org/2009/11/make-better-decisions-2

15. DeDavis, J. (2022). Dewey goes corporate: Examining the suitability of Kotter's change management model for use in libraries. *Journal of Library Administration, 62*(3), 275–290. https://doi.org/10.1080/01930826.2022.2043687
16. Deloitte & Blue Prism. (2023, September 11). *Calculating real ROI on intelligent automation (IA).*Blue Prism. https://www.blueprism.com/resources/white-papers/calculating-real-roi-on-intelligent-automation-ia/
17. De Moraes, C. H.V, Scolimoski, J., Lambert-Torres, G., Santini, M., Dias, A. L., Guerra, F. A., Pedretti, A., & Ramos, M. P. (2022). Robotic Process Automation and Machine Learning: A systematic review. *Brazilian Archives of Biology and Technology*, *65*. https://doi.org/10.1590/1678-4324-2022220096
18. dmTECH. (2023). *Description of the end-to-end process documentation* (internal material). Unpublished.
19. Dumas, M., Rosa, M. L., Mendling, J., & Reijers, H. A. (2013). Introduction to Business Process Management. In *Fundamentals of Business Process Management* (pp.1-31). Springer. https://doi.org/10.1007/978-3-642-33143-5_1
20. El-Gharib, N. M., & Amyot, D. (2022, April 2). *A review of data-driven robotic process automation exploiting process mining*. School of Electrical Engineering and Computer Science, University of Ottawa. https://www.researchgate.net/publication/359729662_A_Review_of_Data-driven_Robotic_Process_Automation_Exploiting_Process_Mining
21. E.Y.F.S. Insights (2016, May 15). *Get ready for robots*. EY. https://www.ey.com/en_iq/financial-services--emeia-insights/get-ready-for-robots
22. Feio, I. C. L., & Santos, V. (2022). A strategic model and framework for intelligent process automation. In *17th Iberian Conference on Information Systems and Technologies (CISTI*) (pp. 1-6). Madrid, Spain. https://doi.org/10.23919/cisti54924.2022.9820099
23. Flechsig, C., Lohmer, J., & Lasch, R. (2019). Realizing the full potential of robotic process automation through a combination with BPM. *Logistics Management*, 104–119. https://doi.org/10.1007/978-3-030-29821-0_8
24. Galli, S. (2020). *Python Feature Engineering Cookbook: Over 70 recipes for creating, engineering, and transforming features to build machine learning models* (2nd ed.) Packt Publishing
25. Galletta, A., & Cross, W. E. (2013). *Mastering the Semi-Structured Interview and Beyond: From Research Design to Analysis and Publication.* NYU Press. http://www.jstor.org/stable/j.ctt9qgh5x
26. Gartner. (2022). *Magic QuadrantTM Report*. Microsoft. https://info.microsoft.com/ww-landing-2022-gartner-magic-quadrant-for-robotic-process-automation.html
27. George, A. L., & Bennett, A. (2005). Case studies and theory development in the social sciences. *The Journal of Politics*, *67*(4), 725-736.https://doi.org/10.1017/S0022381607080231

28. Grand View Research (2022). *Robotic Process Automation Market Size & Share Report 2030.* https://www.grandviewresearch.com/industry-analysis/robotic-process-automation-rpa-market
29. Gurumdimma, N., & Bisandu, D. B. (2018). Understanding error log event sequence for failure analysis. *Science World Journal*, *13*(4), 8 – 15. https://www.ajol.info/index.php/swj/article/view/183592
30. Hamill, M., & Goseva-Popstojanova, K. (2017). Analyzing and predicting effort associated with finding and fixing software faults. *Information and Software Technology, 87*, 1–18. https://doi.org/10.1016/j.infsof.2017.01.002
31. Hanemann, A. (2006). A hybrid rule-based/case-based reasoning approach for service fault diagnosis. In *20th International Conference on Advanced Information Networking and Applications - Volume 1* (AINA'06). Vienna, Austria. https://doi.org/10.1109/aina.2006.29
32. Hazen, B. T., Overstreet, R. E., & Cegielski, C. G. (2012). Supply Chain Innovation Diffusion: Going Beyond Adoption. *The International Journal of Logistics Management,* 23(1), 119–134. https://doi.org/10.1108/09574091211226957
33. Herm, L.-V., Janiesch, C., Helm, A., Imgrund, F., Hofmann, A., & Winkelmann, A. (2022). A framework for implementing robotic process automation projects. *Information Systems and E-Business Management*, *21*(1), 1–35. https://doi.org/10.1007/s10257-022-00553-8
34. Hindel, J., Cabrera, L. M., & Stierle, M. (2020). Robotic Process Automation: Hype or Hope? In *15th International Conference on Wirtschaftsinformatik*. https://doi.org/10.30844/wi_2020_r6-hindel
35. Huff, C. (2021, December 16*). Worker satisfaction improves with Intelligent Automation and RPA.* Spiceworks. https://www.spiceworks.com/hr/hr-strategy/guest-article/worker-satisfaction-improves-with-intelligent-automation-and-rpa/
36. Ivančić, L., Vugec, D. S., & Vukšić, V. B. (2019). Robotic Process Automation: Systematic Literature Review. In C. Di Ciccio et al. (Eds.), *Business process management: Blockchain and Central and Eastern Europe Forum* (pp. 280-295). Springer, Cham. https://doi.org/10.1007/978-3-030-30429-4_19
37. Koch, C., & Fedtke, S. (2020). *Robotic Process Automation: Ein Leitfaden für Führungskräfte zur erfolgreichen Einführung und Betrieb von Software-Robots im Unternehmen*.(1st ed.). Springer Vieweg. https://doi.org/10.1007/978-3-662-61178-4
38. Krakau, J., Kaupe, V., & Feldmann, C. (2021). Robotic Process Automation in Logistics: Implementation Model and Factors of Success. In *Proceedings of the Hamburg International Conference of Logistics (HICL) (219-256).* https://doi.org/10.15480/882.4005
39. Kunz, M., Puchta, A., Groll, S., Fuchs, L., & Pernul, G. (2019). Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications*, *44*, 64–79. https://doi.org/10.1016/j.jisa.2018.11.004
40. Lacity, M. C., & Willcocks, L. P. (2016, September 13). A New Approach to Automating Services. *MIT Sloan Management Review*, *58*(1), 41.

41. Lacity, M.C., & Willcocks, L.P. (2018). Client Service Automation Deployments - What Do They Mean for Your Job and Organization?. *Pulse Magazine.* https://iaoppulse.net/research-corner-ows18-findings-revealed-on-client-service-automation-deployments-what-do-they-mean-for-your-job-and-organization/
42. Lacity, M.C., & Willcocks, L.P. (2021). Becoming Strategic with Intelligent Automation. *MIS Quarterly Executive*, *20*(2), 1–14. https://doi.org/10.17705/2msqe.00047
43. Lensen, A., Al-Sahaf, H., Zhang, M., & Xue, B. (2016). Genetic Programming for Region Detection, Feature Extraction, Feature Construction and Classification in Image Data. In M. Heywood, J. McDermott, M. Castelli, E. Costa, & K. Sim (Eds.), *Genetic Programming: EuroGP 2016* (pp 51–67). Lecture Notes in Computer Science (Vol. 9594). Springer, Cham. https://doi.org/10.1007/978-3-319-30668-1_4
44. Leshob, A., Bédard, M., & Mili, H. (2020). Robotic Process Automation and Business Rules: A perfect match. In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications*, *2*(9),119-126 https://doi.org/10.5220/0009886701190126
45. Leshob, A., Bourgouin, A., & Renard, L. (2018). Towards a process analysis approach to adopt robotic process automation. In *2018 IEEE 15th International Conference on E-Business Engineering (ICEBE)* (pp. 46-53). Xi'an, China: IEEE. https://doi.org/10.1109/icebe.2018.00018
46. Lievano-Martínez, F. A., Fernández-Ledesma, J. D., Burgos, D., Bedoya, J. W. B., & Builes, J. a. J. (2022). Intelligent Process Automation: an application in manufacturing industry. *Sustainability*, *14*(14), 8804. https://doi.org/10.3390/su14148804
47. Lokrantz, A., Gustavsson, E., & Jirstrand, M. (2018). Root cause analysis of failures and quality deviations in manufacturing using machine learning. *Procedia CIRP*, *72*, 1057–1062. https://doi.org/10.1016/j.procir.2018.03.229
48. Moreira, S., Mamede, H. S., & Santos, A. R. (2023). Process automation using RPA – A literature review. *Procedia Computer Science*, *219*, 244–254. https://doi.org/10.1016/j.procs.2023.01.287
49. Ng, K. M., Chen, C., Lee, C. K., Jiao, J., & Yang, Z. (2021). A systematic literature review on Intelligent Automation: Aligning concepts from theory, practice, and future perspectives. *Advanced Engineering Informatics*, *47*, 101246. https://doi.org/10.1016/j.aei.2021.101246
50. Norman, D. (2017). Design, business models, and human-technology teamwork. *Research-Technology Management*, *60*(1), 26–30. https://doi.org/10.1080/08956308.2017.1255051
51. Oracle. (2018). *JD Edwards EnterpriseOne Applications Financial Management Fundamentals Implementation Guide*. *https://docs.oracle.com/cd/E16582_01/doc.91/e15109.pdf*
52. Potturu, S. M. (2023). UiPath Bot Framework: Accelerating RPA Development and Innovation. *IJRDO-Journal of Computer Science Engineering*, *9*(4), 1–15. https://doi.org/10.53555/cse.v9i4.5853

53. Pohlmeyer, F., Kins, R., Cloppenburg, F., & Gries, T. (2022). Interpretable failure risk assessment for Continuous Production Processes based on Association Rule Mining. *Advances in Industrial and Manufacturing Engineering*, *5*, 100095. https://doi.org/10.1016/j.aime.2022.100095
54. Provost, F., & Fawcett, T. (2013). Data Science and its Relationship to Big Data and Data-Driven Decision Making. *Big Data*, *1*(1), 51–59. https://doi.org/10.1089/big.2013.1508
55. Quille, R.V.E.; Almeida, F.V.d.; Borycz, J.; Corrêa, P.L.P.; Filgueiras, L.V.L.; Machicao, J.; Almeida, G.M.d.; Midorikawa, E.T.; Demuner, V.R.d.S.; Bedoya, J.A.R.; et al. (2023). Performance analysis method for robotic process automation. *Sustainability*, *15*, 3702. https://doi.org/10.3390/su15043702
56. Ribeiro, J., Lima, R., Eckhardt, T., & Paiva, S. (2021). Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Computer Science*, *181*, 51–58. https://doi.org/10.1016/j.procs.2021.01.104
57. Santos, F.A., Pereira, R., & Vasconcelos, J. B. (2019). Toward Robotic Process Automation Implementation: An end-to-end perspective. *Business Process Management Journal*, *26*(2), 405–420. https://doi.org/10.1108/bpmj-12-2018-0380
58. Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science, 181*, 526-534. https://doi.org/10.1016/j.procs.2021.01.199
59. Sharda, R., Delen, D., & Turban, E. (2018). Predictive Analytics I: Data Mining Process, Methods, and Algorithms. *In Business Intelligence, analytics, and Data Science: A Managerial Perspective* (4th ed., p. 157). Pearson.
60. Sheikh, N.M. (2013). *Implementing analytics: A blueprint for design, development, and adoption*. (1st ed.). Morgan Kaufmann Publishers Inc. https://dl.acm.org/citation.cfm?id=2500962
61. Sheikholeslami, A., & Graffi, K. (2015). A systematic quality analysis of virtual desktop infrastructure technologies. In *Lecture Notes in Computer Science* (pp. 311–323). https://doi.org/10.1007/978-3-319-27308-2_26
62. Siderska, J., Aunimo, L., Süße, T., Von Stamm, J., Kedziora, D., & Aini, S. N.B.M. (2023). Towards intelligent automation (IA): Literature review on the evolution of Robotic Process Automation (RPA), its challenges, and future trends. *Engineering Management in Production and Services*, *15*(4), 90–103. https://doi.org/10.2478/emj-2023-0030
63. Stirrup, J., & Ramos, R. O. (2017). *Advanced Analytics with R and Tableau* (1st ed.). Packt Publishing.
64. Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., Ter Hofstede, A. H. M., Van De Weerd, I., Wynn, M. T., & Reijers, H. A (2020). Robotic Process Automation: Contemporary themes and challenges. *Computers in Industry*, *115*, 103162. https://doi.org/10.1016/j.compind.2019.103162

65. Tracy, M. C., Jansen, W., Scarfone, K., & Winograd, T. (2007). *Guidelines on securing public web servers*. National Institute of Standards and Technology U.S. Department of Commerce. https://doi.org/10.6028/nist.sp.800-44ver2
66. UiPath Inc. (2020, January). *Measure and Optimize: How RPA Analytics Drive Better Business Outcomes.* https://www.uipath.com/resources/automation-whitepapers/how-rpa-analytics-drive-better-business-outcomes
67. UiPath Inc. (2022, January 19). *Survey reveals businesses are doubling down on automation to balance against the Global LA.* https://www.uipath.com/newsroom/survey-reveals-businesses-are-doubling-down-on-automation
68. UiPath. (2023). *Attended vs Unattended robots*. https://docs.uipath.com/robot/standalone/2023.4/user-guide/attended-vs-unattended-robots
69. UiPath Inc. (n.d.). *Orchestrator User Guide Introduction.* https://docs.uipath.com/orchestrator/standalone/2023.4/user-guide/introduction
70. Wanner, J., Hofmann, A., Fischer, M., Imgrund, F., Janiesch, C., & Geyer-Klingeberg, J. (2019). Process Selection in RPA Projects – Towards a Quantifiable Method of Decision Making. In *Proceedings of the 40th International Conference on Information Systems(ICIS),*Munich,Germany. https://aisel.aisnet.org/icis2019/business_models/business_models/6/
71. Wirth, R., Hipp J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Practical application of knowledge discovery and data mining* (pp. 29-40). Practical Application Company Ltd.

APPENDICES

Appendix 1: Povzetek (Summary in Slovene language)

V vseh podjetjih, ne glede na njihovo velikost in panogo, je skupna točka vzpostavitev učinkovitih poslovnih procesov ključnega pomena. Taki procesi prispevajo k boljšemu doseganju strateških ciljev in dodajajo vrednost samemu podjetju. Ena od faz pri številnih procesih je zajem, pridobivanje in obdelava podatkov, ki morajo biti standardizirani, konsistentni in zanesljivi. Eden od načinov avtomatizacije takšnih strukturiranih opravil za podjetja, ki želijo prihraniti čas in ostati konkurenčna, je uvedba robotske avtomatizacije procesov (RPA). RPA je programska tehnologija, ki lahko lokalno ali prek virtualnega stroja avtomatizira predvidljiva, ponavljajoča se opravila in upravlja z aplikacijami tako, kot bi to storila oseba skozi računalniški zaslon (Alberth & Mattern, 2017). Čeprav so njene prednosti jasne in ima RPA veliko potenciala, njena uporaba ni brez težav, prevladujoči problem je verjetnost pogostih okvar. To magistrsko delo obravnava izziv, s katerim se soočajo podjetja pri zagotavljanju, da so njihovi procesi, ki se izvajajo z RPA, učinkoviti in brez pogostih napak, ki lahko ovirajo strateško usklajevanje in ustvarjanje vrednosti prek nalog vnosa, pridobivanja in obdelave podatkov.

Namen naloge je podjetjem zagotoviti smernice za uporabo podatkovne analitike za indetifikacijo in zmanjšanje vzrokov za napake v procesih RPA, s čimer se izboljša učinkovitost RPA. Kot prvo je cilj naloge preučiti najpogostejše vzroke za napake robotov v kontekstu podjetja s pomočjo analize dokumentacije in identificirati koristi spremljanja procesov. Kod drugo dokumentirati implementacijo podatkovne analitike in oceniti njen potencial za izboljšanje učinkovitosti RPA z uporabo dveh različnih pristopov za analizo podatkov: polavtomatiziran pristop in pristop avtomatiziranega strojnega učenja (ML). S primerjalno študijo se potem oceni, kateri pristop je učinkovitejši pri odkrivanju potencialnih anomalij in napak v procesih RPA. Kot tretje je cilj uporabiti rezultate študije za zagotovitev vpogleda v izboljšave robota, zlasti za preprečevanje ali ublažitev istih napak v prihodnosti.

V magisterskem delu si zastavljamo naslednja raziskovalna vprašanja:

1. Kako lahko podatkovna analitika napove napake RPA in kakšne so koristi ter omejitve njegove implementacije v procese RPA podjetja?
2. Kateri pristop, polavtomatiziran ali ML model, je učinkovitejši za zaznavanje anomalij v procesu RPA?
3. Kako lahko rezultati pristopa izkoristimo za dolgoročno optimizacijo procesa RPA in splošno izboljšano učinkovitost?

Metodologija vključuje kombinacijo teoretičnih analiz, pogovorov s strokovnjaki za RPA glede pomembnosti stalnega spremljanja parametrov za zagotavljanje izboljšav procesa in razvoja robota za pridobivanje podatkov za zbiranje preteklih podatkov o delovanju. Orodja, kot sta Python in PowerBI, se uporabljata za analizo podatkov in ustvarjanje interaktivnih nadzornih plošč za spremljanje metrik in lažje raziskovanje podatkov.

Ključne ugotovitve kažejo na pomembnost enostavnega postopka dostopa do podatkov, izbire ustreznih metrik za zmanjšanje napak, vzpostavitve jasnih pravil za analizo temeljnih vzrokov (RCA), ki temelji na pravilih, učenja iz vzorcev napak, izboljšanja komunikacije z deležniki in merjenja uspešnosti skozi čas s prilagodljivim pristopom k izboljšanju procesa.

Prispevek te magisterske naloge je v njenih praktičnih in teoretičnih spoznanjih o avtomatizaciji in izboljšanju procesov RPA. Ponuja primerjalno analizo različnih pristopov podatkovne analitike za optimizacijo RPA, skupaj s praktičnimi orodji, kot so nadzorne plošče PowerBI, ki podpirajo prizadevanja za izboljšanje procesov. Raziskava poudarja potrebo po uskladitvi potreb delešnikov z zmožnostmi robotov, za izboljšanje natančnosti in učinkovitosti ponavljajočih se opravil, kar na koncu prispeva k boljšemu odločanju in učinkovitosti poslovanja.

Appendix 2: Interview Questions

RPA developers

1. Based on your experience, what are the primary factors contributing to RPA project failure?
2. What techniques do you use to identify bottlenecks or inefficiencies within RPA processes?
3. How frequently have you utilized Elastic for monitoring robots?
4. How regularly do you communicate with your team about encountered failures, and do you have suggestions for streamlining the communication?
5. How can we effectively manage unforeseen issues in robots that occur sporadically and without predictability?

RPA consultants

1. What are the typical steps taken to address a process when it fails?
2. What is the typical duration required to resolve a failure in a process?
3. What methods are currently in place to ensure the team is informed on time about issues and their needed involvement?
4. What areas do you believe should be our primary focus for process improvement, are there specific metrics or KPIs you consider crucial in this regard?
5. What strategies do you think are most effective in motivating clients to adopt new processes and in demonstrating the success of RPA?

External interview

1. Could you share some insights about your role and your journey within the company so far?
2. Given your extensive experience in both technical and sales dimensions, could you elaborate on how RPA has enhanced business processes, particularly in the retail sector?
3. Which UiPath tools and analytics features have you utilized in your RPA projects, and could you share any notable experiences from working with them?
4. What best practices would you recommend for organizations looking to integrate analytics into their RPA processes or to enhance their process monitoring?
5. How do you perceive the role of machine learning in UiPath's ecosystem today, and what are your predictions for the future intersection of RPA and data analytics?
6. What are some specific challenges you've encountered with RPA, and in your experience, what are the most frequent causes of RPA projects failing?
7. Could you share any personal tips or advice from past experiences that might be relevant for improving RPA processes?

Appendix 3: Python Commands – Reference File Preprocessing

Table 3: Python Commands - Reference File Preprocessing

Command	Description
df.dropna(inplace=True)	Removing empty cells
final_df.replace(to_replace=[r"\"", r"\[", r"\]"], value='', regex=True, inplace=True)	Replacing unwanted characters
datetime.strptime(str(date), '%d/%m/%Y %H:%M:%S'	Fixing formatting inconsistency
dt_obj = datetime.strptime(ts, '%b %d, %Y @ %H:%M:%S.%f') datetime_combined= dt_obj.strftime('%m/%d/%Y %H:%M:%S.%f')[:-3]	Fixing formatting inconsistency - split timestamp
df.drop(['Key', 'Robot', 'Environment'], axis=1, inplace=True)	Retaining only relevant features
logsmerged_data.rename(columns={'message': 'LogMessage'}, inplace=True)	Standardizing column names
if len(merged_data.columns) == 1: merged_data = merged_data[merged_data.columns[0]].str.split(delimiter, expand=True)	Split columns example

Source: Own work

Appendix 4: Python Commands – Merging Jobs and Logs Datasets

Table 4: Python Commands – Merging Jobs and Logs Datasets

Command	Description
group_dict[job_log_id] = f"{business_process_name}_{group_counter}"	Key identifier 'job_log_id' creation

datetime_to_log_id = {row['log_datetime']: row['job_log_id'] for index, row in logs_data.iterrows()}
 	
def find_closest_log_id(start_time):
min_diff = pd.Timedelta(minutes=2)
closest_log_id = None
 	
for log_datetime, log_id in datetime_to_log_id.items():
diff = abs(log_datetime - start_time)
if diff <= min_diff:
closest_log_id = log_id
min_diff = diff
 	
return closest_log_id
 	
jobs_data['job_log_id']=jobs_data['Start(absolute)'].apply(find_closest_log_id)	

merged_df = pd.merge(logs_data, jobs_data, on='job_log_id', how='left')	Merging the jobs and logs data

Source: Own work

Appendix 5: Constructed Column Descriptions

Table 5: Constructed Column Descriptions

Column	Description
newly_added	True if the project version is added within hours and does not exceed 2 days.
log_exception_type	System or business exception type indicated.
retry_number	Count of the consecutive transaction retries.
maximum_retry_number	The maximum retry number reached.
secondary_cause	The defined statement of secondary failure reasons, such as updates that happened in the last two days or that newer dependency versions existing that the project's one,
false_success	True if there is a success mail being sent, while there is a log level error, warn or fault message.
false_failure	True if maximum retry is the only error in the process or there are no log level error, warn or fault messages, but the end status is still faulted.
increment	False if the same transaction keeps repeating, suggesting that the retry_number column repeatedly shows 0.
CPU_usage	CPU usage number taken from the robot's monitor process information activities is indicated in a specified column.
old_UiPath_activities	True if there is a legacy language used and the UiPath framework error handling is not properly setup.
simultaneous_process	The name of the process that was already running when the current process was initiated, meaning that there is a timeline overlap.

Source: Own work

Appendix 6: Python Commands – Resulting Failure Reasons

Table 6: Orchestrator User Setup Failure Reasons

Command	Failure reasons
if pd.isnull(group['windowsIdentity']).all()	User is not entered in folder Users
matching_versions = df_gitlab_filtered['projectVersion'] == process_version	GitLab project version is different
(pd.isna(valid_from) or valid_from <= log_datetime) and \ (pd.isna(valid_until) or valid_until >= log_datetime) and \ (pd.isna(cancelled_date) or cancelled_date >= log_datetime)	No IAM Role access rights
if pd.isnull(row['Failure_Reason']):group.loc[index, 'Failure_Reason']	Undetermined failure reason

Source: Own work

Table 7: SharePoint Activities Failure Reasons

Command	Failure reasons
for v1, v2 in zip(v1_parts, v2_parts):if v1 < v2:return -1	Check the Dependency version
if not df_rpa.empty and not check_windows_identity_presence(df_rpa, extracted_windowsidentity)	No App ID and Tenant ID Asset Value added
valid_roles_appid = df_roles_filtered_appid.apply(lambda role: is_role_valid(role, log_datetime), axis=1)valid_roles_mailbox = df_roles_filtered_mailbox.apply(lambda role: is_role_valid(role, log_datetime), axis=1)	No IAM Role access rights (SharePoint related 'Mailbox' and 'AppID' roles checked)
if 'error occurred sending the request' in log_message	User credentials or URL link issue

Source: Own work

Appendix 7: DAX Formulas – Creating Columns and Measures

Table 8: DAX Formulas – Creating Columns and Measures

DAX formula	Column/Measure
VAR TotalHours = [total_executiontime] * 24 VAR Hours = FLOOR(TotalHours, 1) VAR TotalMinutes = (TotalHours - Hours) * 60 VAR Minutes = FLOOR(TotalMinutes, 1) VAR Seconds = ROUND((TotalMinutes - Minutes) * 60, 0) RETURN FORMAT(Hours, "00") & ":" & FORMAT(Minutes, "00") & ":" & FORMAT(Seconds, "00")	Total execution time in minutes
Table.AddColumn(#"Removed Columns", "Time", each DateTime.Time([log_datetime]), type time) Table.AddColumn(#"Inserted Time", "Hour", each Time.Hour([Time]), Int64.Type)	Extracting the time portion from log_datetime column
DIVIDE([Total Failed Jobs], [Total jobs], 0)	Error rate as a division between failed jobs and total number of jobs executed
VAR FailureIDs = CALCULATE(DISTINCTCOUNT('Data'job_log_id]), 'Data'[false_failure] = 1) RETURN DIVIDE(FailureIDs, [Total jobs], 0)	Percentage of false failure
SUMX('Unique Job_Log_ID Table', 'Unique Job_Log_ID Table'[false_failure] * 3 + 'Unique Job_Log_ID Table'[false_success] * 2 + 'Unique Job_Log_ID Table'[old_UiPath_activities] * 1 + IF('Unique Job_Log_ID Table'[No_Increment] <> 0, 1, 0) * 2)	Process Risk Score: • False robot failure in a job significantly increases the risk score • False robot success and no increment in the job are also a risk factor, but it's given a lower weight • Old_UiPathActivities' is less critical

Source: Own work