RABAN - A SOFTWARE IMPLEMENTATION PROCESS FOR ROBOTIC PROCESS AUTOMATION (RPA) PROJECTS

K.V. Jeeva Padmini

(168013L)

Degree of Doctor of Philosophy

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2022

RABAN - A SOFTWARE IMPLEMENTATION PROCESS FOR ROBOTIC PROCESS AUTOMATION (RPA) PROJECTS

Kumara Vidanalage Jeeva Padmini

(168013L)

The thesis was submitted in partial fulfilment of the requirement for the Degree of Doctor of Philosophy in the Department of Computer Science and Engineering.

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2022

DECLARATION

I declare that this is my work, and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

UOM Verified Signature

..... 25-5-2022

K.V. Jeeva Padmini Date:

(Signature of the candidate)

The above candidate has carried out research for the Ph.D. thesis under my supervision.

UOM Verified Signature

25-5-2022

(Dr. G.I.U.S Perera) Date

Signature of the Supervisor

UOM Verified Signature

25/05/2022

(Dr. H.M.N. Dilum Bandara) Date

Signature of the Supervisor

COPYRIGHT STATEMENT

I hereby grant the University of Moratuwa the right to archive and to make available my thesis or dissertation in whole or part in the University Libraries in all forms of media, subject to the provisions of the current copyright act of Sri Lanka. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

UOM Verified Signature

K.V. Jeeva Padmini (168013L)

ABSTRACT

Robotic Process Automation (RPA), the next level of business process automation, provides adaptive and transformative solutions to replace time-consuming, non-value-adding, and repetitive human tasks in a Business Process (BP). RPA based BP transformation projects differ from typical software development projects because RPA bots are developed on stable code. It is counterproductive to use existing software processes in RPA projects. A process template (i.e., software implementation process and metrics to track the project) is yet to be derived for RPA projects. The estimated initial RPA project failure rates are 30-50%, and the lack of a fitting implementation process is attributed as one of the key contributors to failure. We addressed this gap and derived a novel process for RPA projects named Raban and metrics to track RPA projects.

Scrum was used to formulate the Raban. Focus group discussions were conducted with scrum teams and identified 80 challenges. Those analyzed in Straussian grounded theory are grouped into six categories (i.e., *lack of agile mindset*, *inconsistency in story estimation*, *client management issues*, *lack of adherence to agile practices*, *scope change in requirement freeze*, and *lack of quantitative measurement*). Prioritized 15 burning challenges were classified based on significance, and taxonomy was developed. Derived steps to estimate RPA use-cases and a framework to achieve customer satisfaction adopting design thinking practices in agile projects. Moreover, 17 software metrics and three artifacts were derived and validated in five scrum projects. Raban was derived based on the solutions identified and further fine-tuned based on the feedback from follow-up interviews with the stakeholders and two workshops conducted with the other RPA project teams. After that, 14 metrics and two artifacts were derived for Raban and validated in a RPA project. Moreover, to select the right candidate BP for RPA transformation, predictive machine learning model was developed, where the decision made as yes/no on RPA suitability. We used 16 factors and a two-class decision forest classification model to develop the model.

Keywords: agile framework, agile metrics, design thinking, decision support tool, robotic process automation

ACKNOWLEDGEMENT

Completing this research work would not have been possible without the support of many people who have always wished for my success. I would like to use this space to express my deepest gratitude to those with me in my dark times and for their encouragement and continuous support.

I am eternally grateful to my supervisors Prof. G. I. U. S. Perera, and Dr H. M. N. Dilum Bandara of the Department of Computer Science and Engineering department at the University of Moratuwa. First of all, I would like to thank Dr. Dilum Bandara for his encouraging words to start a Ph.D. journey. His insight was helpful for me to view the research from a different angle. There would not be an appearance of this thesis without whose support. It is a pleasure to study under his guidance, immense patience, and encouragement during the tough times in my Ph.D. journey. I am grateful to Prof. G. I. U. S. Perera, who opened my eyes to see the opportunities to liaise behind in the world as a researcher. His scholarly advice colored and shaped my Ph.D. journey. I owe Dr. Chandana Gamage, who offered relentless cheer and support to survive in the research environment. He never gives a second thought to give his hand when in need. Further, I would like to thank the academic and non-academic staff in the Department of Computer Science and Engineering at the University of Moratuwa. Thanks also to Mr. Dinesh Bandara and Ms. Nisansala Samarasuriya, for the administrative support given within the University.

Moreover, a huge thank goes to Mr. Madu Rathnayaka and Mr. Sirinival Alluri for the enormous support by sharing their insight into research work and using Virtusa Pvt Ltd as the base for my research. I take this moment to thank especially my colleagues in the process team, project managers and development, testing teams, and all the staff at Virtusa Pvt Ltd for their tremendous support and patient show during data gathering and analysis. I'm grateful to the professionals in the industry and friends who supported me by participating in the online surveys, face-to-face interviews, and focus group discussions conducted within the research. Moreover, hats off to the academic staff, colleagues, and friends in the Department of Computer Science and Engineering at the University for their assistance and advice contributed to this

research. Thank you very much, Prof. Uditha Rathnayaka, who always encouraged and took time to check on me and made me shine inside throughout my Ph.D. journey.

I am forever grateful to my dearest parents, who have encouraged me every time in my academic achievements. My beloved husband, daughter, and son are always there to relieve the stress and improve the self-belief to succeed in challenges faced.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASD	Agile Software Development
BPC	Business Process Complexity
CBPS	Candidate Business Process Status
CCM	Cyclomatic Complexity Metric
CF	Cognitive Features
CMMI	Capability Maturity Model Integration
CSM	Certified Scrum Masters
DD	Data-Driven
DMI	Delivery Maturity Index
DST	Decision Support Tool
EOL	End Of Life
HCM	Halstead Complexity Metric
HE	Human Error
IC	Integrated Circuit
IOF	Impact Of Failure
IT	Information Technology
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
KLOC	Thousands of Lines Of Code
LOC	Lines of Codes
MMI	Multi-Model Input
NOSA	No of Systems to Access
OC	Operational Cost
OTD	Ontime Delivery
PO	Product Owner
PQ	Product Quality
PP	Project Predictability

PT	Process Templates
PVT	Production Verification Testing
QA	Quality Assurance
ROC	Rate of Change
RBBP	Rule-Based Business Process
RC	Regulatory Compliance
RPA	Robotic Process Automation
SIT	System Integration Testing
SLA	Service Level Agreement
SLOC	Source Line Of Code
SME	Subject Matter Expert
SOE	Stability Of Environment
SPSS	Statistical Package for the Social Sciences
TDD	Test Driven Development
VOT	Volume of Transaction
SQuaRE	Software PQ Requirements and Evaluation
TP	Team Productivity
TVI+	Target Value Increase
UAT	User Accepting Testing
VOT	Volume of transaction
WIP	Working In Progress
WLV	WorkLoad Variance
WV	Workload Variance
XP, XP2	Extreme Programming

LIST OF APPENDICES

1 INTRODUCTION

1.1 Motivation

Business process automation is the computing-technology-enabled automation of key business processes [1]. Robotic Process Automation (RPA) is the frontier of business process automation [2], representing computer software (aka. virtual bots) programmed to execute repetitively, non-value-adding, and stable codebase. It replaces labor-intensive tasks that do not require high skills, as it automates actions at the application front end [2], [3]. RPA could gain benefits such as cost reduction, improved process efficiency, and reduced rework tasks within the process [4]. RPA is not like software development, maintenance, testing, or service projects [2], because RPA automates business processes on existing systems with a stable codebase [5]. Therefore, RPA projects cannot be executed efficiently using waterfall, spiral, agile, scrum, and kanban [6]. Further, as RPA is a new process automation technology, a process template is yet to be defined. Consequently, projects and teams struggle to meet the client's expectations and improve client satisfaction. Hence, it is imperative to formulate a new process template for RPA projects.

1.2 Problem statement

RPA is not without its challenges. According to Lumberton et al. [7], at the first attempt of 30%-50% of RPA projects are failing. The authors identified 10 common issues for such failures. Four of them can be attributed to business reasons, and the rest can be attributed to project challenges. One of the critical issues among them is the application of traditional software implementation methodologies. Watson and Wright's [8] survey highlighted the need to carefully select a project governance process and correct implementation at the onset of the RPA journey.

Further, the author's findings emphasized the importance of using agile practices for RPA project governance. Moreover, Lamberton et al. [7] proposed the agile delivery approach for the RPA implementation process. However, they did not

verify RPA project execution in the agile approach empirically, where we derived the process for RPA projects based on agile principles and values. While Asatiani and Penttinen [2] discussed the RPA business process selection procedure in the Opus Capita case study, RPA implementation methodologies were not covered. As RPA projects automate end-to-end business processes on top of stable systems, existing process templates cannot be used for RPA projects. For example, even though well-defined process templates for development, maintenance, testing only, and service delivery projects are available in the industry, they cannot be used in RPA projects. Scratch product development was categorized into the development projects, making changes due to end-user requirements to the existing stable product was categorized as maintenance projects, and executing only testing activities in a project was categorized into a testing project. The team that provides only customer support was categorized as service delivery projects. However, RPA projects cannot be categorized into either of the categories mentioned above because RPA projects automate the end-to-end business process on top of stable systems. Therefore, the company could not use existing Process Templates (PT) for RPA projects.

Further, during a preliminary discussion with project managers in RPA projects, the need for a process template for RPA projects was highly emphasized. Therefore, identify an appropriate process template for the successful implementation of RPA projects is crucial. While success has different dimensions, in this work, we consider reducing the client-reported defects at the User Acceptance Testing (UAT) as the critical measure of success. We formulated our research question to be addressed as follows:

How to derive a process template for RPA projects that minimizes client-reported defects at the UAT?

As a process template has many aspects, this broader research question could be addressed by exploring answers to the following sub-questions:

1. To what extent do existing software implementation processes apply to RPA projects?

2. How to reduce additional costs due to change requests identified during the UAT phase of a bot implementation project?
3. How to select the Candidate Business Process (CBP) for RPA transformation?
4. How to identify customer needs and improve customer satisfaction in RPA projects?
5. How to track and manage RPA projects to improve product quality, team productivity, and project predictability?

1.3 Objectives

The above sub-questions are addressed by achieving the following objectives:

- To investigate the applicability of existing software implementation processes for RPA projects.
- To formulate a software implementation process to execute RPA projects to reduce additional costs due to change requests identified during the UAT.
- To derive a tool to identify CBP for RPA transformation.
- To identify methods to capture customer needs and best practices to improve customer satisfaction.
- To derive software metrics to track RPA projects to improve product quality, team productivity, and project predictability.

1.4 Research contribution

The literature review did not reveal any process templates designed for RPA projects. We focused on formulating a software implementation process for RPA projects and deriving metrics to track and manage RPA projects. We conducted the research at global IT services and delivery company to identify the problem domain, test the proposed solutions, and solicit expert feedback.

In phase I, the global IT services and delivery company executed an RPA project in a plan-driven process. It is the most used software implementation process

in the industry and is known to most practitioners. However, 29 change requests, 33 clients reported defects, and 120 other defects were captured at the UAT proving that the plan-driven process was unsuitable for RPA projects. Then in phase II, we executed RPA projects using an in-house process developed at the global IT services and delivery company based on test-driven development methods. However, after phase II, the project had twenty change requests, 10 client-reported defects and twenty-two other defects.

Consequently, we decided on formulating a novel software implementation process for RPA projects. Most of the change requests and defects were due to the changes made to the UAT environment by the client maintenance team without informing the project team (i.e., the RPA development team) and requirements missed (e.g., rarely used features) during the requirement gathering phase. We identified these issues were due to the communication gap between the client maintenance team and the project team. Therefore, in developing a novel process model, we set the following objectives:

- Identify the best way to capture requirement changes early in the bot implementation process.
- Identify the best way to capture missing requirements early in the bot implementation process.
- Reduce additional costs due to change requests identified during the UAT phase.

Given the agile process's potential suitability for RPA, we focused on identifying a suitable agile practice to adopt for RPA projects. We first assessed the agile suitability using the checklist developed by the global IT services and delivery company process team. The agile suitability tool proposed by Jalali et al. [9] was used to validate the agile suitability further. As per the outcome of both the tools, we confirmed the agile process's suitability for RPA projects. According to the literature, it is known that the scrum process absorbs requirement changes during product implementation and improves customer collaboration [10], [11]. However, the scrum framework cannot be used for RPA projects due to the fundamental differences

between product development and RPA projects. Based on the results gathered in phases I and II, we formulated the Raban framework in phase III. Salient features of the proposed implementation process are as follows:

- Covers identification, bot implementation, and deployment stages of the process.
- The use-case backlog is used in the identification stage to capture and maintain requirements, requirement changes, and missing requirements as early as possible.
- The use-cases of the backlog are estimated during the estimation meeting.
- Use recurring collaborative sessions with process executers, business Subject-Matter Experts (SMEs), and Business Process owners (e.g., use-case backlog building, robotize-sprint review) to eliminate requirement issues.

We empirically validated the proposed Raban framework in a project at global IT services and delivery company. Phase III resulted in only three change requests and zero defects. Based on the stakeholders' feedback, it was determined that the proposed Raban framework significantly improved the visibility, adaptability, business value, and on-time implementation of the bot. Moreover, the client could start user and cognitive bot training once the client received the first working bot in the project, enhancing the accuracy of the overall bot functionality. Collaborative sessions proposed in the Raban framework develop a strong client relationship, which was essential to familiar with the existing system and promptly acknowledge new feature developments integrated by the client maintenance team.

At first, we studied the malpractices and challenges faced by agile practitioners and identified best practices and solutions for agile projects. The same we used in formulating the software implementation process for RPA projects as it implements adhering to agile principles and values. Due to that, we conducted a set of focused group discussions within the company agile practitioners. Eighty challenges were identified during the focus group discussion. Out of the 80 challenges, 15 challenges were prioritized as burning challenges by the agile practitioners. We

researched best practices and solutions for those 15 challenges as an initial step. Then, to get better understanding on the solutions for 15 challenges, online survey developed and shared across the industry. We developed a taxonomy considering the identified 15 challenges and respective solutions. We then conducted semi-structured interviews with five agile SMEs to validate the proposed recommendations with the taxonomy report. However, no change was made to the taxonomy. Finally, we proposed solutions and best practices for scrum teams to overcome the identified 15 burning challenges.

In addition, in order to improve the choice of CBP in Raban, we have introduced a Decision Support Tool (DST). Using DST, the prediction model helps select the candidate process for RPA projects. Of the factors identified through interviews with RPA SMEs, 16 were prioritized and developed the DST model. The variability of the workload, the number of access plans, and the service level agreement are positively correlated. Negative relationships have elements of artificial volume, compliance rules, and perceptual elements. The other 10 factors had a non-trivial relationship so, we used all (16) factors to improve the model. Because, for RPA SMEs these factors are important. Moreover, our database is small. To predict whether to conduct RPA transformation or not, we used a two-class decision forest classification. A four-step verification process was used to define the accuracy of each model, which achieved 90% accuracy. Five RPA SMEs' have successfully used DST on three completed RPA projects to validate.

The 80 challenges from focused group discussions were categorized into six groups using Straussian grounded theory: *lack adherence to agile practices*, *scope change in requirement freeze*, *lack of quantitative measurement*, *lack of agile mindset*, *inconsistency in story estimation*, and *client management issues*. As a solution to Lack of adherence to agile practices, scope change in requirement freeze, and lack of quantitative measurement challenges, we identified 17 metrics and three artifacts to track and measure scrum projects in product quality, team productivity, and project predictability. We conducted a four-day agile workshop and one-hour assessment to enhance the agile mindset of team members. We developed a set of steps to follow during the use-cases estimation of an RPA project to overcome the challenges identified under inconsistency in story estimation. We derived a framework to achieve

customer satisfaction in agile-based projects by adopting design thinking practices for the challenges listed in the client management issues category. A detailed description of the solutions for the six categories of challenges is explained in the upcoming paragraphs. Further, data analysis proves that everyone is familiar and comfortable with the company's existing scrum tool usage.

To solve the challenges of *lack of quantitative measurement*, *scope change in requirement freeze*, and *lack of adherence to agile practices* categories, we derive software metrics to track and manage the RPA projects to improve product quality, team productivity, and project predictability. Based on nine semi-structured interviews with scrum masters, we identified 32 metrics. Further, 49 metrics were captured from the literature review. We used the constant comparative method to analyze the 81 metrics captured from the literature review and semi-structured interviews. 19 duplicate, one irrelevant, 19 good metrics, and 26 metrics that violate scrum principles were removed from further analysis. Consequently, we finalized 13 metrics and three artifacts to manage scrum projects to improve product quality, team productivity, and project predictability. Then, we conducted another set of semi-structured interviews with agile SMEs to validate the findings. Agile SMEs also proposed four new metrics. After incorporating the comments from the SMEs, we finalized must have 17 metrics and three artifacts. The proposed metrics cover the challenges listed in the *scope change in requirement freeze* and *lack adherence to agile practices* categories. Considering the proposal made by the agile SMEs, we derived two metrics to assess the adherence to agile practice and scope change in requirement freeze. We validated the metrics and artifacts in five scrum projects of the global IT services and delivery company. Next, a brainstorming session was conducted with the project team, RPA SMEs, and process consultants to finalize the metrics for RPA projects run on the Raban framework. We identified five product quality metrics, three-team productivity metrics, one artifact, and one project predictability metric. One metric and an artifact were designed to measure in the speculate stage. Two metrics and one artifact were proposed to measure in the explore stage. Five metrics target the inspect and adapt stage. One of the metrics needs to be measured monthly, while two other metrics

should be measured quarterly. To validate the identified metrics, we then executed the identified metrics in five RPA projects.

Under the *lack of agile mindset* category, we identified a lack of awareness of agile principles and values within the RPA team. We developed a four-day scrum master workshop and a one-hour assessment test to address that. Only those who passed the assessment were chosen for the RPA project team. Moreover, we introduced an estimation process to estimate use-cases in RPA projects to solve the challenges listed in *inconsistency in story estimation*. Size and effort estimation cannot estimate the use-cases in RPA projects. RPA expert knowledge is required to understand the use of case complexity.

Further, we identified best practices to achieve customer satisfaction by adopting design thinking practices to solve the challenges listed under *client management issues*. We analyzed the data collected from interviews with SMEs using the Straussian grounded theory approach to give meaning to develop a theory. We identified 10 challenges while satisfying customer expectations and 15 techniques to understand customer expectations. Moreover, we derive a framework from achieving customer satisfaction in agile-based projects adopting design thinking practices. The framework consists of five sections where we identified best practices for each section to improve customer satisfaction. We use these results in formulating the software implementation process for RPA projects.

Moreover, we can list the research contribution as follows.

1. Derived a process template (i.e., software implementation process and metrics to track and manage the project) for RPA projects.
    a. Derived a novel process for RPA projects named Raban.
    b. Derived 14 metrics and two artifacts to track and manage the RPA projects.
2. Developed a taxonomy to find solutions for the 15 prioritized challenges.
3. Derived steps to estimate RPA use-cases in RPA projects.

4. Derived a framework to achieve customer satisfaction by adopting design thinking practices in agile projects.
5. Seventeen software metrics and three artifacts were identified for scrum projects to track and manage product quality, project predictability, and team productivity.
6. Developed a predictive model to select the right candidate BP for RPA transformation.

1.5 Outline of the research

The thesis structure is as follows. Chapter 2 provides an overview of the related work on robotic process automation in Section 2.1. Section 2.2 discusses the iterative and incremental process against the plan-driven process, and Section 2.3 describes the high-level view of the agile framework. The five phases of the agile project management model are described in Section 2.4. Section 2.5 discusses agile practices available in the industry, and section 2.6 describes software metrics. Finally, it explores related work on design thinking and process templates in Sections 2.7 and 2.8, respectively.

Chapter 3 presents the research methodology used to achieve research objectives. Section 3.1 elaborate on the research approach overview followed in this thesis. Section 3.2 to 3.6 describes the five steps to achieve the research objectives. Those are background study, execute RPA process in plan-driven process and in-house developed model, formulate a process for RPA project, execute RPA projects in novel process, derive metrics from managing RPA projects, respectively.

The detailed design process followed in formulating the software implementation process is described in Chapter 4. Section 4.1 presents a background study to understand the existing software implementation processes. RPA project execution in a plan-driven process is explained in Section 4.2, whereas Section 4.3 explains the RPA project execution using an in-house development process. Next, in Section 4.4, we describe the process used to identify the suitable agile practice to adopt

the software implementation process. Then in Section 4.5, we describe the formulation of the proposed software implementation process. RPA project execution in the Raban framework and its results illustrates in Section 4.6. Section 4.7 clearly describes the end-to-end steps of the Raban framework. Finally, Section 4.8 showcases the output of the RPA project execution in the Raban framework, along with a justification for the Raban framework for RPA projects.

Chapter 5 describes the implementation of the Decision Support Tool (DST) to select CBP. Section 5.1 presents the steps to understand the factors' impact on CBP selection. Section 5.2 describes the steps followed in the development of the prediction model. Section 5.3 discusses the prediction model outcome after executing it for the existing RPA transformation project and the model's success in the CBP selection for RPA transformation.

Chapter 6 describes the detailed steps to identify must-track metrics in scrum projects. Section 6.1 describes the steps to identify must-track software metrics for Scrum projects to improve product quality, team productivity, and project predictability. Then Section 6.2 describes the steps to identify software metrics to be used in the Raban framework.

Concluding remarks are presented in Chapter 7. First, research implications are presented in Section 7.1. Next, research limitations are described in Section 7.2, and in Section 7.3, we describe future work.

2 LITERATURE REVIEW

The competitive pressure of today's business and market demand have enhanced the shortest possible time to deliver software solutions. Therefore, the software development industry exercised to use the methodologies under the agile umbrella, and the number of large enterprises that embrace agile continues to increase each year [1]. This chapter discusses relevant work on robotic process automation and process implementation for RPA projects.

Section 2.1 discussed robotic process automation. Section 2.2 presents the agile framework and Agile project management model discussed in Section 2.3. Section 2.4 discusses the practices available under the agile umbrella, such as scrum, kanban, scrumban, and extreme programming. Related work on agile-specific metrics is evaluated in Section 2.5. Section 2.6 focuses on design thinking, and Section 2.7 presents process templates.

2.1 Robotic Process Automation (RPA)

RPA is a groundbreaking technology to perform repetitive, rule-based knowledge work substituting human workers [12]. A virtual worker, aka., bot, is used to carry out the repetitive and non-value, adding structured operational tasks and procedures typically performed by humans [5]. The bot is designed to behave like a proxy human worker[13]. Virtual workers let the human workers focus more on knowledge-based creative work. The person involved in customer service could spend more time to interact with the customer instead of uploading data forms which filled by the customer.

To successfully deploy, it is essential to spend a reasonable amount of time evaluating, analyzing, and planning the implementation [2], [6]. The author proposed a four-step process for bot development: conduct an RPA workshop, conduct the process assessment, submit the business case proposal, and then work on the RPA implementation. However, the authors provided minimal details on how to execute each step. [7] discovered after analyzing 20 companies experience in RPA transformation, that 30-50 percent of the first projects failed. The use of traditional software implementation methodologies to bot implementation was also identified as

a critical issue for the high failure rate. While [7] suggested adhering to agile principles as a potential solution, it was not verified empirically. Therefore, we focused on deriving a framework for RPA projects adhering to agile principles.

2.2 Iterative and incremental process vs. Plan driven Process

Plan-driven process such as the waterfall model was relatively simple to understand. Within the process, phases will not be overlapped [14]. At the end of one phase only, the next phase will start (see Figure 2.1). However, as seen in Figure 2.2, all the phases in the agile framework are based at the same time, iteratively. It absorbs feedback and encourages frequent inspection and adaptation [15]. In a plan-driven process, completed software will be delivered mainly at the end. However, in the ASD process, the working software and features will be shipped as deliverables within a minor period [16]. The Plan-driven process has a high risk and high probability of failing the project since after requirements gathering, the customer will not get involved with the product until the product is handed over, other than if there is no specific requirement gathering [14]. If the customer wants any changes towards the final stage of the project, the cost will be high since it needs more effort and time to consume. Further, code change can be huge, and effort will be high in cost. As illustrated in Table 2.1, user requirements elicitation and analysis are completed in the first phase of the product. Requirement revisit happens after that, whereas agile requirements gather iteratively absorbing changes [16]. Therefore, rework cost is low, and customer involvement is high compared to the waterfall model. At the same time, hyper productive teams are expected in an agile environment with cross-functional capabilities and to adapt to frequent change in a limited time [17]. However, no such expectation with the teams in waterfall model projects. The software development life cycle contains frameworks and models to plan and maintain the process. The iterative and incremental process helps enhance the product features to meet rapidly changing user requirements. The plan-driven process was insufficient to satisfy the frequently changing requirements in the fast-moving business environment. An iterative and incremental process introduced, and best practices documented in 2001 as the “Agile

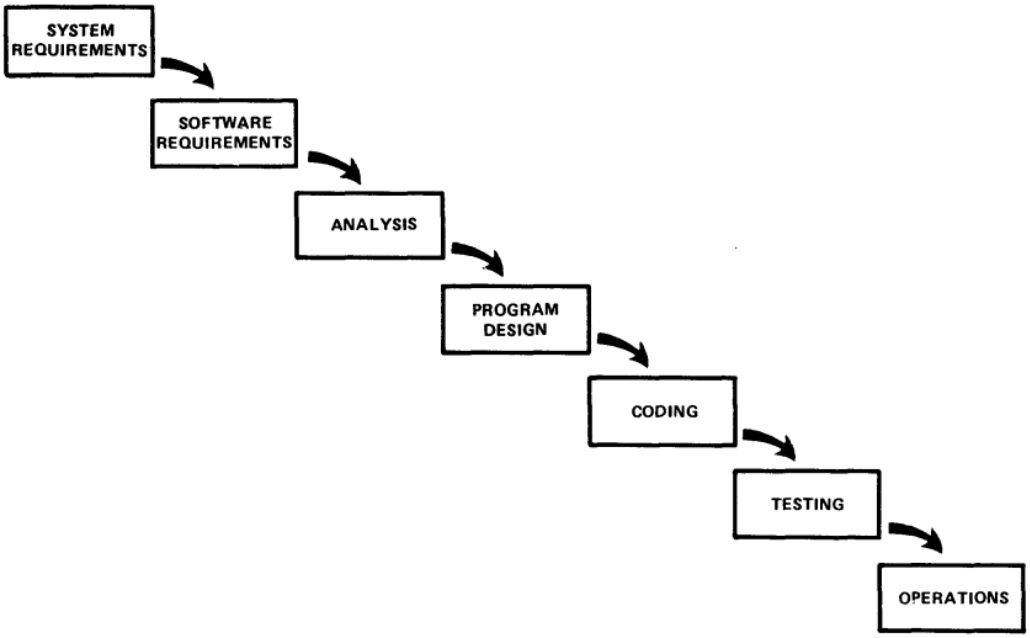

Figure 2.1 Phases flow in waterfall methodology: source [14].

Manifesto" with four values and 12 principles [15] to deliver a minimal viable product. Because both iterative and incremental and waterfall model processes have their pros and cons, we decided to test both the processes for RPA projects because they do not have a properly defined framework for companies. Chapter 4 describes the work carried out to derive a process template for RPA projects.

2.3 Agile framework

Agile principles and values focused on delivering a software product as early as possible to the customer with high quality, high business value, with lower cost. The agile framework helps overcome the need to respond in this rapidly changing environment, which delivers working software to the client in every iteration with hyper-productive and self-organizing teams [17]. Four values proposed in "Agile Manifesto" are as follows [15];

> "*Individuals and interactions* over process and tools
> *Working software* over comprehensive documentation
> *Customer collaboration* over contract negotiation
> *Responding to change* over following a plan."

There are 12 principles [15] that follow those four values. The first value emphasizes improving interaction within the agile project team. Therefore, the interaction between stakeholders and the agile project team was improved by conducting face-to-face communication instead of sending e-mails. No project manager for agile teams. Scrum master facilitates the agile team to achieve the sprint goals with the self-organized and hyper-productive agile team members. Each team member is responsible for completing assigned tasks on time, which reduces conflict between the development team and stakeholders. "Working software over

Table 2.1: Deviation from Traditional to Agile method.

#	Feature	Agile Method	Traditional Method
1	User Requirement	Iterative acquisition	Detailed user requirements are well-defined before coding/ Implementation
2	Rework cost	Low	High
3	Development direction	Readily changeable	Fixed
4	Testing	On every iteration	After coding phase completed
5	Customer involvement	High	low
6	Extra quality required for developers	Interpersonal skills and basic business knowledge	Nothing in particular
7	Suitable Project scale	Low to medium-scaled	Large-scaled

Source: [16] .

comprehensive documentation" value focuses on iterative and incremental product development. Completing comprehensive documents does not bring any value to the customer. Therefore, the agile project team completes an end-to-end product feature at the end of each iteration. At the end of each iteration, the customer can play with the completed features delivered and decide on the changes required for future implementations. The third value focuses on developing what the client wants and satisfying the customer. Work was considered a close interaction and collaboration with the customer to enrich customer satisfaction. Finally, the fourth value helps the

customer compete with the market trends by absorbing changes at any given time. The customer has the best fit for the current industry to compete with others.

Agile is a lightweight software development process. Agile practices and customized frameworks available under the agile umbrella help remove the heaviness associated with the plan-driven process. The agile framework can be applied in many situations where other methodologies failed to be used. Simultaneously, the agile framework is less filled with heavy documentation than a plan-driven process [18]. It still allows having required documentation as per the project requirements. Active customer involvement and frequent feedback are indispensable features in the agile framework compared to other software life cycles [18]. However, as seen in Figure 2.2, the agile framework is based on iterations, encouraging frequent inspection, adaptation, and absorbing feedback. Stakeholders review (Sprint review) the work carried out during the iteration (Sprint). Finally, working software or completed features will be delivered, enabling swift feedback from customers rather than waiting till the user acceptance phase. Consequently, the cost of rework due to changes can be minimized**.** Therefore, the agile framework is more reliable compared to other software implementation processes, as it welcomes changes to the existing requirements at any time during the project implementation [9]. In short iterations, the iterative and incremental approach develops innovative solutions, where design thinking, a catalyst for innovative development or transformation, leads to transformation [10].

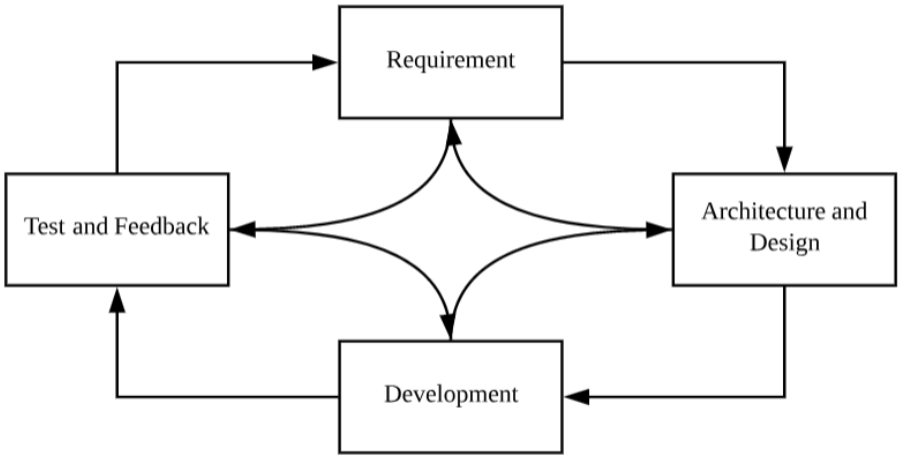

Figure 2.2: Iterative and incremental process in agile practice. Source: [10].

According to the agile report's 11th annual state, 94% of the companies are practicing agile and agile projects have a success rate of 98% [12]. The projects' success rate based on the agile framework was three times higher than non-agile projects [12]. Therefore, in the next sections, we studied the agile project management model and agile practices more.

2.4 Agile project management model

As illustrated in Figure 2.3, the Agile project management model is structured into five steps: envision, speculate, explore, adapt, and close [14]. Each step is presented in detail in the next session.

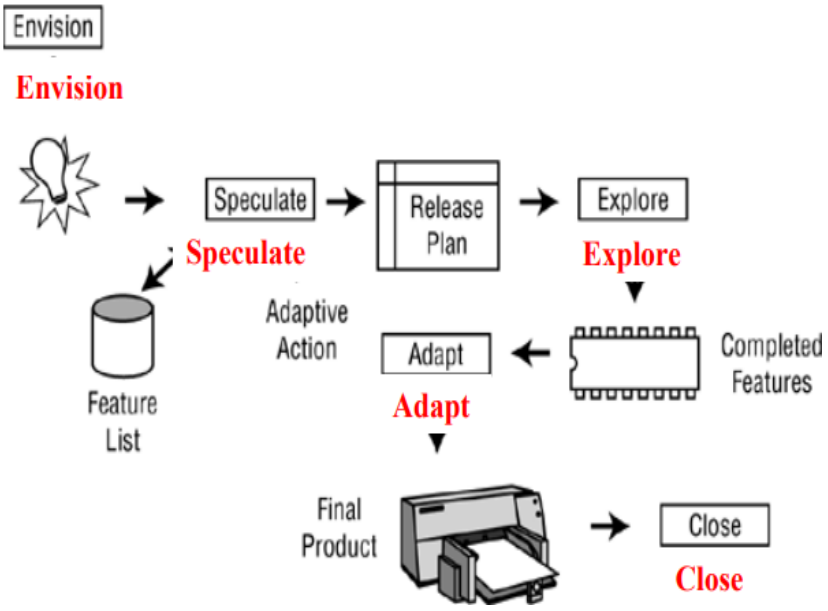

Figure 2.3: Agile project management model. Source: [14].

2.4.1 Envision

Envision is the initial stage of the project. Customers, stakeholders, or product owners engage with the project team to identify the project's vision and scope, the community of the project engaged with, and project team working strategy. The project community understands what to deliver, who will be involved, and how everyone intends to work together.

2.4.2 Speculate

During the speculate phase, the project vision transformed into a set of requirements and an understanding of the final product delivery plan. The team will start elaborating on the requirements by breaking them into user stories until they can estimate. The team also needs to prioritize to understand the work order on product implementation. Once the team has groomed the requirements, they need to work on release level planning, wave level planning, and iteration level planning. At the same time, risk mitigation strategies were also incorporated into the plan.

2.4.3 Explore

Explore is the execution phase of product implementation. In this phase, the team delivers completed features. Three critical activities happened within this phase. Those activities deliver planned features, maintain a collaborative, self-organizing project community, and manage the proper team interaction with the customer and stakeholders.

2.4.4 Adapt

The team received review feedback from the customer for their completed work. The team improved through iterative development by identifying their own mistakes. Change can happen on the project approach, project process, project objective, and project environment during project execution.

2.4.5 Close

This phase aims to understand the completed project's learning outcome and apply the collective knowledge to the new projects.

We used these five steps in the project management tool to derive the Raban framework and metrics to track and manage the project.

2.5 Agile practices

Agile development is an umbrella term. Under that, there are different methodologies derived based on the agile framework. The agile umbrella has many offerings depending on the industry requirements. As of 2001, Scrum, XP, Crystal, FDD, and DSDM were recognized under the agile umbrella. Today many more options are available, and some of those are listed in Table 2.2.

2.5.1 Scrum

Among all iterative and incremental methodologies listed in Table 2.2, scrum is the most commonly used and most adopted development framework in more than 50% of projects [12]. Takeuchi and Nonaka (1986) developed the scrum framework based on the best practices of successful companies such as Toyota, Canon, Honda, and Fujixerox [17]. As illustrated in Figure 2.4, by applying simple constraints, average teams in the scrum framework can self-organize into a hyper-productive state. The scrum team works in 2-4 weeks iterations to deliver a potentially shippable product. Daily scrum, backlog grooming sessions, planning of the sprint (sprint planning), Review of the sprint deliverables (sprint review), and Retrospect with the team on sprint (sprint retrospective) are the scrum framework ceremonies.

Table 2.2: Agile development methods.

#	Agile Method	Description
1	Crystal methodologies	This is a popular, flexible, and lightweight approach which can be used in the software development process. Clear, Yellow, Orange, Red, and Blue are the categorization of the team size. Focused on developing small software product which is not life critical.
2	Dynamic software development method	Dynamic software development method consists of eight key principles focused on the quality, business need, deliver on time, development iteratively, collaborate, demonstrate control, build incrementally from firm foundations, communicate clearly and continuously.
3	Feature-driven development	This process is focused on design-oriented developments. A software development project is divided into small pieces of the product, called features. This follows a five-step development process like; build by feature, plan by feature, develop an overall model, design by feature, build a feature list.
4	Lean software development	Adopting principles used in the manufacturing industry. Focused on minimum waste to clean products at low cost. Lean principles cover seven principles; namely, eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole.
5	Scrum	The self-organizing hyper-productive team executes in increments (sprint), Team starts with sprint planning and end the sprint at sprint review. List of product features listed in the backlog. The product owner manages the product backlog and scrum master is there to facilitate the scrum team to work towards the project goal.
6	Extreme programming (XP; XP2)	Focuses on five values to achieve high-quality product development with quality life for software developers. Communication, simplicity, courage, respect, and feedback are those five values. A lightweight process to develop software in a fun way.

Rugby sports use the word scrum to denote the self-managed team member and to showcase the spirit of teamwork in a cross-functional team [15], [16]. The word “Scrum” is derived from that. The scrum framework increases productivity by 5-10 times against the industry average [17]. In distributed scrum, three distributed scrum modules can be found, such as isolate scrums, fully distributed scrum, and distributed scrums of scrums [19]. Scrum behavior goes hand in hand with RPA project characteristics. Therefore, we used scrum as a base to derive the Raban framework. However, we cannot use scrum as it is for the RPA project process due to the fundamental difference between product development and RPA projects. We described this in detail in Section 4.4.

2.5.2 Kanban

Kanban was one method used to achieve the “Just in Time” production developed by Taiichi Ohno at Toyota. This method enhanced production and maintained high-level production [20]. Kanban methodology mainly focuses “to accurately state what work needs to be done and when it needs to be done, prioritize tasks and define workflow, as well as lead-time to delivery” [21]. For a defined time, developers focus only on a few items to maintain releases to the customer in a constant flow. It communicates priorities and highlights bottlenecks, which helps to improve visibility in the software implementation process. Kanban focuses on maintaining the continuous flow of work over fixed iterations [22].

Kanban can improve scrum and other agile methods, focusing on fast production, rapid and continual user feedback, and interaction [23]. Kanban principles address the human tendency to resist change. While scrum defines three roles, kanban does not define any and can be defined based on user needs or continue using existing user roles. Kanban is a more flexible framework than scrum and can be modified based on users' needs.

2.5.3 Scrumban

Scrumban is a combination of scrum and kanban. Iteration planning is conducted at regular intervals, along with reviews and retrospectives. The goal of planning is to fill the slots available, not fill all the slots, and not define the number of slots. Iteration is not defined as in the scrum framework instead defined on a need base as in kanban. Roles are not defined and can be used on a need basis, as in kanban. New tasks can be accepted in the live iteration. The limitation is only for Work In Progress (WIP) items. Work prioritization is managed in separate columns. Scrumban does not have planned meetings as in the scrum framework, conducted on-demand. This method is suitable for maintenance projects, projects prone to programming errors, or projects with incredible user stories. This methodology is not recommended for massive projects. However, projects with five teams can be suggested since

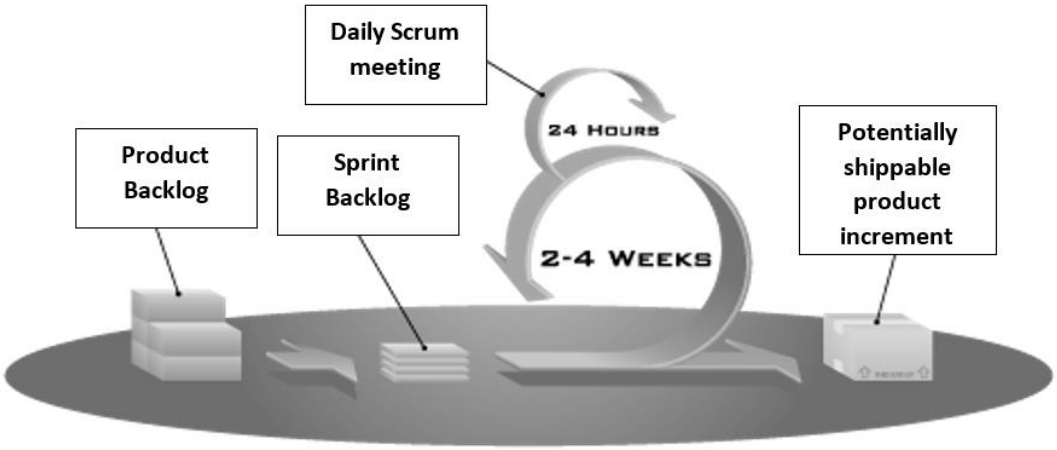

Figure 2.4: Scrum framework. Source: [12].

coordination issues may become a problem. The number of kanban and scrumban users nearly doubled in 2012 and increased more in 2013 [24].

2.5.4 Extreme Programming (XP)

Kent Beck and Martin Flower are the co-founders of Extreme programming (XP). XP helps to manage vague or rapidly changing requirements. Moreover, XP is a lightweight, flexible, and low risk disciplined approach to software implementation. XP methodology is better to be used in small and medium-sized teams. XP enables frequent releases, rapid feedback, and defects management [25]. XP starts with planning and then moves into designing, coding, and testing. Different practices are available in different phases. Planning porker used in XP estimation process. XP works on pair programming through collective code ownership. In coding, the customer also gets involved in helping the team and is considered a team member. The Test-Driven Development (TDD) process is followed by XP, where the test is written before coding starts. Other than testing in TDD, programmers are involved in writing automated test cases for the individual development of a testable product. The programmer or the tester can execute manual or automated testing. However, TDD expects to have defect-free code. TDD has been used to develop the in-house developed process model for RPA project execution at the global IT services and delivery company. Section 4.3 illustrates the steps followed in the RPA project execution using an in-house developed process model.

2.6 Software Metrics

Measurements help take proactive actions to assist project management [18], and metric helps estimate how well we did it [19]. Further, it would be helpful to analyze the project quantitatively. Moreover, metrics assist in developing models [20]. “Software metrics measure all the phases in software process, such as requirement gathering, design, development, and test. Therefore, software metrics are valuable entity in the process” [20]. Separate metrics are required for agile framework as metrics used in waterfall model process cannot be directly used in the agile framework [21]. In contrast to the plan-driven process, agile framework only simple

metrics may help maintain safe and consistent growth in a hyper-productive agile team [18].

In the plan-driven process, software metrics are divided into three sections: Project, process, and product metrics [22]. The product characteristics like performance, quality, design aspects, size, and complexity were measured using 'Product Metrics'. The 'Process Metrics' measured the software development, maintenance, and testing. Various project characteristics, like productivity and time [22], were measured using 'Project Metrics'. In the plan-driven process, the complexity of the software was measured by the Cyclomatic Complexity Metric (CCM), Halstead Complexity Metric (HCM), and Lines of Codes (LOC) metrics. McCabe's Cyclomatic complexity metric measured several independent paths in a software module. Defect Removal Efficiency metric measures how many bugs are captured in production against the metric identified during implementation [23]. Defects per KLOC (1,000 lines of code) or defects per function points to measure the software reliability using the Defect Density metric [24]. Source line of code or SLOC metric, Object-oriented metrics [20], and Function point metric was measured in the plan-driven process. Duplicated code metric and dead code metric to measure duplicate code and to measure never used code and duplicate code, dead code metric, and duplicate code metric are used respectively [32].

However, there is no specific method of metric classification in the agile framework. As described in [22], most metrics focus on ceremonies, fixing software process problems quality measurement, motivating people in the scrum framework, and progress tracking. In [25], metrics are divided into business value, predictability, quality, cost, and lean. Under quality, Technical debt, Defect count, Faults-slip-though, Lead-time, and Work in progress queue metrics are measured. The cost attribute was measured using the Average cost per function metric, Value measured using velocity, predictability, Running automated test cases, and business value delivered by Customer satisfaction survey, respectively. Based on the study conducted by [17] metrics are classified as Economic metrics (e.g., earned business value and break-even point), Productivity/effort level (e.g., burn-down charts and project size

units), and Code level (e.g., Running tested features, Leffingwell's iteration, and release perspectives, and code quality and design metrics). Further in [26], the authors have categorized the metrics into seven: Engineer, Test, Automation, Project management, Strategic, Iteration, and Code, as illustrated in Figure 2.5. In each category, the author has identified metrics. Useful metrics in the agile framework helps to improve team performance. Hence, when metrics select the agile framework, they should be lightweight and easily measurable. Further, it should be a simple metric that can be maintained easily [17], as the agile framework is more lightweight than a plan-driven process. Metrics used in the agile framework should provide visibility into code rather than setting up code level metrics [27]. The author in [18] has identified 10 essential and meaningful metrics that management can use for decision-making in the scrum. These are Velocity, Work Capacity, Focus Factor, Percentage of Adopted Work, Percentage of Found Work, Accuracy of Forecast, Targeted Value Increase (TVI+), Success at Scale, and Win/Loss Record without any categorization. In [10], the author has listed a set of customized metrics resulting from the survey conducted within a selected company. These metrics are Delivered on time, Technical debt, Unit test coverage for the developed code, Fault correction time to "closed" state, and in testing, Regression test cycle time, Smoke test cycle time, future measurements, such as the Definition of the done checklist. However, no classification was highlighted. In [28], the author has conducted an industrial survey and has identified industry-used metrics. Out of the 22 metrics, the most used 10 metrics focus on Product Quality, Team Productivity, and Project Predictability categorization. Once the results were finalized author has interviewed the SMEs and come up with the metric categorization. Those metrics are Open defect severity index, Unit test coverage for the developed code, Bug correction time from "new" to "closed" state, in product quality. Work capacity, Velocity, Percentage of adopted work, Sprint level effort burn down, Percentage of found work in team productivity. Delivery on time and Focus Factor on project predictability. Moreover, without looking into metric categorization, authors have studied focusing on different areas of a software project, such as quality [20], [19], [29], [30], predictability [21], [31], and team performance [32].

2.6.1 Software development product quality

Quality in a software product is essential. In [20], the author has stated that "Given the penetration of computer code into everyday objects like washing machines, automobiles, refrigerators, toys and even things like the mars rover, any system be in a large one or a small system running embedded Integrated Circuit (IC) technology, ensuring the highest levels of software quality is paramount." To achieve customer expectations, we must understand the quality of the software product [19]. ISO/IEC 9126:1991 quality standards are for software quality metrics. A feature can define the value taken for a specific software product [29]. In [20], the author stated that "Software product metrics play a central role in software engineering, and their proper

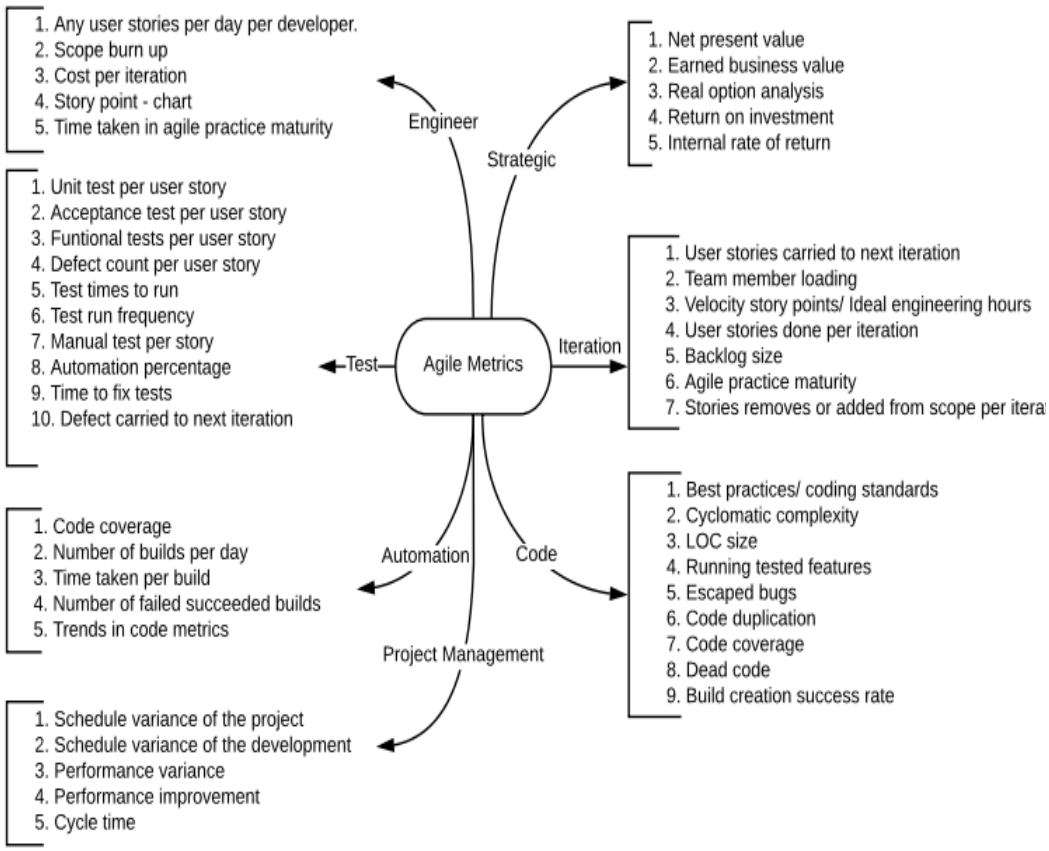

Figure 2.5: Agile metric mind map. Source: [26].

validation will ensure that there is a compelling case for their use in practice." Software product metrics can be validated as follows [29]:

- What product metric is supposed to measure. (e.g., coupling metric should measure coupling)
- Associated with some essential external metrics (e.g., maintainability or reliability)
- An improvement over existing product metrics. (e.g., it is easier to collect the metric or a better predictor of faults.)

Quality characteristics for the first international consensus were published in ISO; "software product evaluation quality characteristics guidelines for their use" in 1991. The standard continues to expand with some changes introduced between 2001 and 2004 (ISO 9126: 1991). New metrics and artifacts suitable for the projects run on the agile framework [30] are required to identify. Next-generation software product quality standards are the Software product Quality Requirements and Evaluation (SQuaRE - ISO 25000), which replaced ISO 14598 and ISO 9126 series of standards.

2.6.2 Software development team productivity

Agile is a human-centric and adoptive framework, which allows small and collaborative teams to work together. Related work focuses on product, personal, project, and process factors. Table 2.3 lists the team productivity factors identified in empirical studies. However, very little can be identified, especially concerning team productivity. Two industrial case studies [32] mentioned that team composition and allocation, staff turnover, and external dependencies are the main factors influencing agile team productivity.

Table 2.3: Factors impact on Team productivity.

Productivity Factors		Description
PROD	Reuse	Reuse the software processes, artifacts, software products, frameworks, and product lines.

Productivity Factors		Description
	Software characteristics	System characteristics like; domain, stability requirement, architecture, complexity, non-functional requirements, user interface, and software size.
PERSONNEL	Team experience and capabilities	Consider the factors like customer experience, domain knowledge, and generational experience, i.e., staff capabilities and experience, programming language experience, and capability of tool handling.
	Motivation	Motivation to work in the company and perform on a project.
	Project Management	Considered conflict management, task assignment, aspects of quality of management, and administrative and formal coordination.
PROJECT	Resource constraints	Constraints include project duration storage, reliability, team size, and timing.
	Schedule	Considered expansion and schedule compression.
	Team composition	Considered team collocation, team size, and staff turnover.
	Communication	Considered informal face-to-face communication.
PROCESS	Customer participation	Considered direct customer involvement with the project.
	Daily builds	Frequent integration of system components.
	Documentation	Project registration and product knowledge.
	Early prototyping	Resolve potential high-risk issues in prototyping.
	Incremental and iterative development	Different ways of producing system parts, testing, and receiving feedback on the final product or partially completed working product.
PROCESS	Modern programming practices	Structured code, top-down requirements analysis, design structure, design notation, etc.
	Programming language abstraction	Level of abstraction of the language (e.g., Java is a high-level language)
	Software Methods	Different software practices and methodologies.
	Tools usage	Use of IDEs, CASE tools, etc.

Source: [32]

2.6.3 Software development project predictability

Improving predictability in the software implementation industry leads to smooth work and reduces variability by identifying and removing bottlenecks and eliminating delays. In the plan-driven process, project tasks are pre-defined. In completing one phase, the only team can move into the next phase. Requirements are defined in the requirement phase only. Once the client signifies the requirement,

changes are not accepted with effect to the budget. Defining requirements clearly at the requirement phase is challenging for clients due to complex, interconnected, interdependent, and interrelated business processes. However, the agile framework is the alternative approach when the requirements are elusive, volatile, and subject to change [21]. Predictability is the foremost important thing to most customers. Predictability in a project helps to prevent forming a problem. The use of predictability metrics helps the team make accurate estimates in pre-sales and helps with work prioritization and stakeholder management. Following are measurements for predictability [31]:

1. Work in Progress – How many items we are working on in a given time.
2. Cycle Time – Duration of work takes to complete the process.
3. Throughput – In project predictability, number of work items complete per unit of time.

2.7 Design Thinking

Understanding customer needs and wants precisely is the most critical part of software implementation [33]. Many software implementation projects have failed due to clarity in user expectations. Therefore, we used design thinking practice to understand customer needs and wants early as possible in a project. Design thinking is a systematic process and a new concept for the software development industry [34]. The study conducted by [35] has proposed a framework focused on design thinking practices, cognitive approach, and mindset. Human-centered design, thinking by doing, visualizing, synthesis of diverging and converging, and collaboration are listed under design thinking practice. The cognitive approach has four elements: holistic viewpoint, integrative thinking, abductive thinking, and reflective reframing. Future-oriented, experimental and explorative, ambiguity tolerant, and optimistic elements are described under mindset. Out of these three dimensions in the framework, we focused on five design thinking practices and their applicability to improve customer satisfaction in software development projects. Design thinking provides a creative strategy for complex, unknown, or ill-defined problems [36], where the design

thinking process evolves through empathizing, defining, ideate, prototype, and test stages [37]. Design thinking focused on understanding customers' expectations before finalizing the preferred solution. Design thinking helps project managers to gain new ideas from available resources [36]. Design thinking is novel for the software development industry, especially in agile. However, it is used effectively in entertainment, education, space exploration, and medicine [36]. Design thinking practices could significantly enhance product needs and customer expectations. Both the processes can be practiced iteratively. However, using design thinking in agile practice to improve customer satisfaction is unknown among professionals. Agile methodology and design thinking go hand in hand. Both principles focus on poorly understood problems, close interaction, going for feedback, and solution providing an iteration [11]. Design thinking chose to take up biotechnology research and development, entrepreneurs in problem-solving or looking for opportunities and facing challenges or improving decision-making. Likewise, design thinking can be used in user interface designing in agile methodology [13].

2.8 Process Templates

Different templates are used to ensure the process executes with the expected quality. System requirement templates [38], system architecture templates, software detailed design templates, unit test templates, software integration templates, and Quality Assurance (QA) test templates [39] are a couple of those templates. The use of these templates helps to get everyone in the software development life cycle on one page. *Process templates* are one of the template types used in software development that describes the process behavior [40]. It should be clear to understand the process by reading the template itself. Process templates include process guidance, tools, work items, permission information, a project structure, project road map, and check-in policies to be used in the software creation. Process templates are fully customizable to deliver the project aligning to the selected process. Process templates include all the steps in the project life cycle. It clearly explains contract type with the customer to project close procedures. It describes the tools used throughout the process. Process templates describe the metrics used within the selected process life cycle. The process

templates are clearly defined, selecting the proper process template according to the project behavior in selecting the proper process template.

2.9 Discussion

We identified the research gap as no defined process template for RPA projects. Existing process templates are for development, maintenance, testing, and services projects run on the plan-driven and agile processes. Literature has not discussed the software development process for RPA projects. The industry is new to RPA technology, and limited organizations run on RPA projects that are not mature enough to concentrate on the software development process for RPA projects. Hence, in this research, we conducted a case-based study on global information technology services and delivery companies to formulate a process template named Raban for RPA projects. Moreover, we used Straussian grounded theory [41] for data analysis because it supports literature review to derive new concepts and theoretical sampling. We immediately completed the open coding process at the end of each interview and analyzed the contents with text. After that, we broke the data into parts then compared it to identify the key terms to add them into the categories. Key terms are compared continuously with the collected data and identify new inclinations till they reach theoretical saturation. Relationships and disaggregation of the open coding were identified using axial coding and finally identified the factors using selective coding. Further, we derived a software development process to develop Raban and identified metrics to measure the formulated software development process.

3 RESEARCH METHODOLOGY

This chapter discussed the steps followed and procedures used during the design, data gathering, and study analysis. We used four research tools: literature review, focus group discussions, online survey, and face-to-face semi-structured interviews for the data collection. Literature review findings mainly contributed to developing the research methodology.

The research approach followed in this thesis elaborate on Section 3.1. Section 3.2 to 3.6 describes the five steps to achieve the research objectives. Step 1 in Section 3.2 describes the literature review carried out in the background study. Step 2, described in Section 3.3, presents the steps followed while executing RPA projects in existing process models. Step 3 in Section 3.4 describes the steps followed while formulating the software implementation process for RPA projects. Steps 4 in Section 3.5 explain the steps followed while the RPA project was executed using the proposed process model. Step 5 in Section 3.6 describes the steps followed in deriving metrics for the novel process. Finally, Section 3.7 presents the chapter summary.

3.1 Overview of the research approach

This research focused on developing a process template for RPA projects. For the research study, we used the empirical research method out of four research paradigms: scientific, engineering, empirical, and analytical [42]. As described in [42], the scientific method is more suitable for physics, chemistry, and biology experiments. The engineering method is more suitable for the work carried out to improve the solution for a model they already have. The analytical method is more formal and follows a mathematical method that the researchers are using this to propose formal theories. Only the empirical method uses statistical and qualitative methods to measure and analyze the new model's impact [42]. Therefore, we selected an empirical method to conduct our research study. Borges *et el.*, [43] have identified 375 mechanisms in an empirical method, especially for software engineering research. The researcher has the freedom to select the best suitable method for the study based on the research objectives, environment, and available materials [44]. We referred to the decision-

making model proposed by [41] to select research mechanisms. Which explains eight steps in three phases to conduct the research. *Research Outcome*, *Research Logic*, *Research Purpose*, *Research Approach* in Strategy Phase. *Research Process* and *Research Methodology* in Tactical phase. *Data Collection Methods* and *Data Analysis methods* in the Operational phase. Further, the model describes different mechanisms to be used in eight fields. Based on the mechanisms described by [45], in the strategy phase, "Applied Research" was selected as *Research Outcome* because we focused on finding the solution for an existing problem, i.e., deriving a process to execute RPA projects. "Inductive Research" is used for *Research Logic*, as we focus on deriving the process for RPA projects based on the observed data. Moreover, *Research Purpose* falls into "Exploratory Research" as RPA is a booming technology with little research work. We have used "Interpretivist" research aims under *Research Approach* to understand the human behavior in executing RPA projects. In the tactical phase, we have used the "Mixed Approach" for *Research Process* and the "Case Study" method for *Research Methodology*. In the operational phase, we have used "Interviews", "Observations", and "Surveys" for *Data Collection Methods*, and "Grounded theory" as *Data Analysis Methods.*

We conducted the study at global IT services and delivery company, which does not have a defined process template for RPA projects. However, the company had well-defined process templates for development, maintenance, testing only, and service delivery projects. RPA was new to both the project team and the client. As discussed in Chapter 2, minimal literature was available on the RPA technology, and no literature was found on the software implementation process for RPA projects.

Moreover, the researcher could not identify the process template used in the industry for RPA projects during face-to-face interviews with industry experts. Hence, we formulated a process template for RPA projects consisting of the software implementation process and metrics to track and manage the tasks executed based on the formulated software implementation process. Figure 3.1 illustrates the detailed steps we carried out to formulate the process template for RPA projects. Our research methodology can be explained using the five steps that are described next.

3.2 Step 1 – Background study

As illustrated in Figure 3.2, we studied the RPA project execution methodologies used within the industry in the first step. We understood that the RPA technology is new to the industry, and minimal academic and industry publications were available. Even in that, none of the researchers or the technologists discussed the implementation process used in RPA projects. Hence, we could not identify a suitable implementation process for RPA projects from the existing literature.

3.3 Step 2 – Execute RPA projects in existing process models

The project team and the client decided to execute phase I of the project to execute the RPA project based on the plan-driven process. This research refers to the waterfall model as the plan-driven process. Phase I took eight months to complete and was delayed and eventually completed with 29 change requests, 33 defects, and 120 other defects. The project team consisted of 23 team members: a project manager, an architect, two business analysts, 13 developers, and six QA engineers. In the requirement gathering phase, a business analyst and an architect were on-site with the client, and the rest of the team was offshore. The business analyst and architect conducted requirement gathering workshops at the client site with operational system users (aka. process executors) who demonstrated their existing process. The client conducted system demonstrations while the business analyst captured screenshots of the business process workflows. At the end of the requirement gathering phase, the requirement document (i.e., process definition document) was developed, and approval was obtained from the client's leadership. The client had a pipeline of business processes to automate. Hence, once a team completed a bot implementation, they moved on to the pipeline's selected CBP. Among popular RPA bot implementation tools such as Workfusion, Automation Anywhere, and Blue Prism, the client chose to use WorkFusion. Workfusion is a cloud computing platform to automate business process operations [46]. Three bots had to be developed for the business process selected in this phase. Two bots were implemented parallels as two sub-projects. The business process use-cases were estimated during the design stage.

Moreover, the architecture design and QA test plan were finalized and signed off by the client. After completing the development, the deliverables were shared with the QA team for System Integration Testing (SIT). The first two bots took four weeks to develop and another three and six weeks to SIT. The deliverables then moved on to the UAT phase. While the first bot took three weeks for UAT, the second bot took four weeks. Once the UAT was completed, the client conducted Production Verification Testing (PVT) in the pre-production environment. Due to many change requests, the project team had to put extra effort into completing the work. Next, we interviewed the project team, SMEs, RPA practice heads, and process consultants to identify what went wrong in phase I.

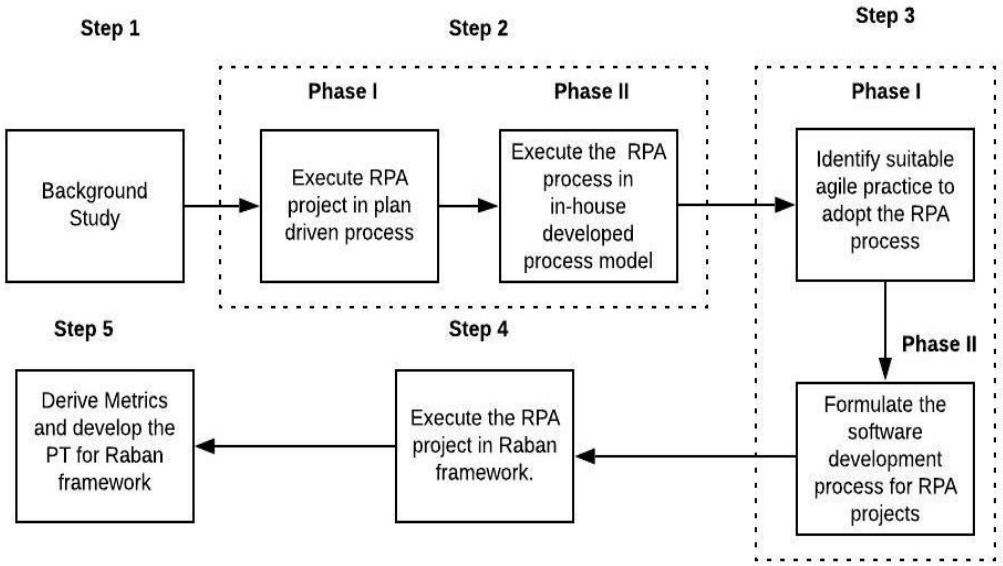

Figure 3.1: High-level view of the research methodology.

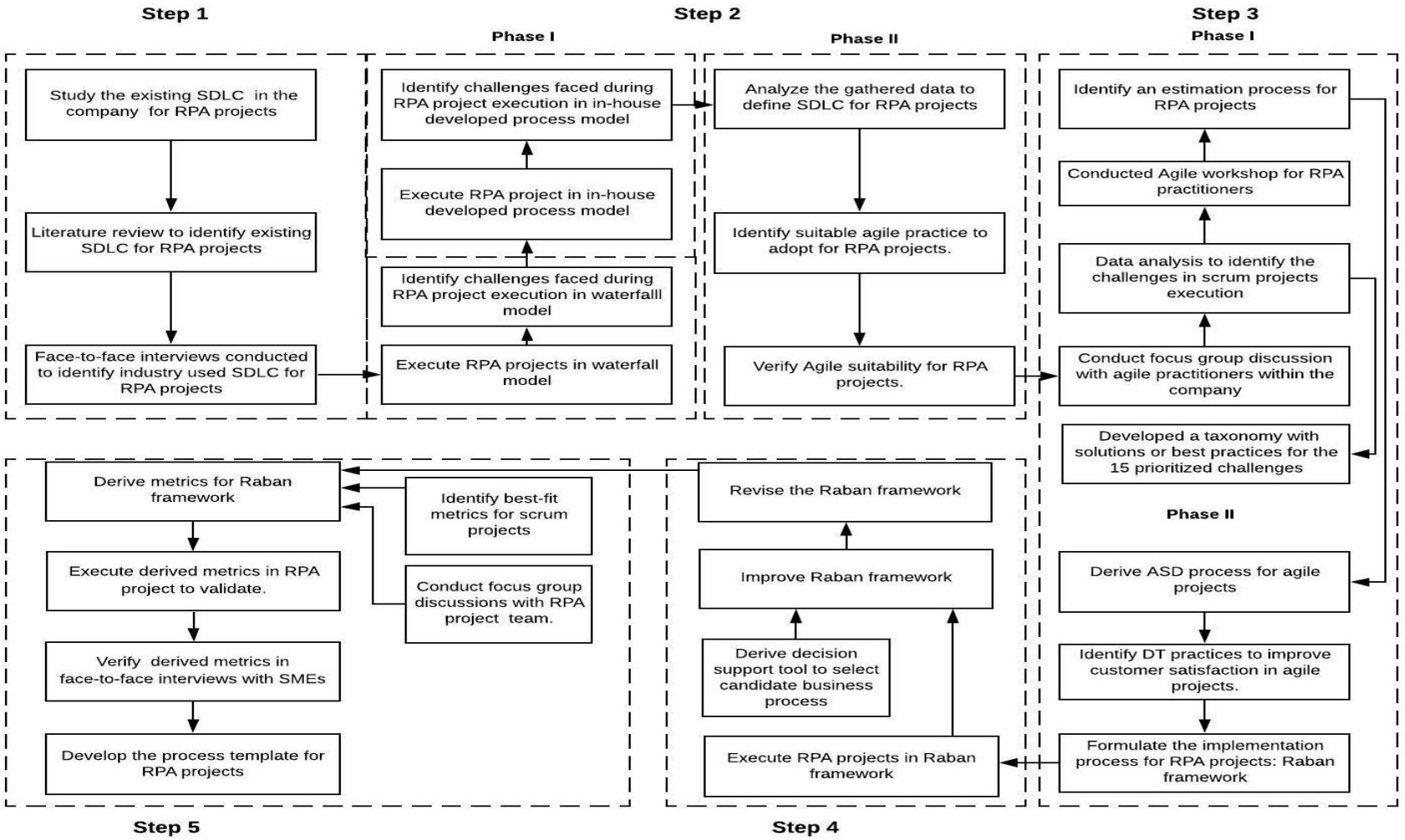

Figure 3.2: Detailed view of the research methodology.

Phase II in step 2 spanned 12 months and implemented three bots for the selected business process. The project team remained the same. To overcome the issues experienced in Step 1, the global IT services and delivery company developed a process based on test-driven development and named the ‘in-house process model’ to execute phase II.

3.4 Step 3–Formulate a process for RPA projects

The plan-driven and in-house developed processes do not eliminate the change requests and defects identified during UAT. Hence, we decided on formulating the software implementation process for RPA projects and started exploring the existing software implementation processes. We studied the implementation processors, frameworks, and industry best practices, to manage the required changes during the implementation of a bot. Moreover, we selected the scrum framework as a suitable agile practice to execute RPA projects as the scrum process absorbs frequent requirement changes in product development and improves customer collaboration.

First, we assessed the agile suitability using the checklist developed by the global IT services and delivery company’s process team. Then we generated the radar chart for the chosen RPA project. We used the agile suitability tool proposed by [9] to validate further the agile suitability tool developed by global IT services and delivery company. However, we understood that the scrum framework could not be used for the RPA projects due to the fundamental differences in product development projects and RPA projects. Hence, we started formulating the implementation process for RPA projects in Step 3. As the initial step, we try to identify the challenges and malpractices in the scrum and eliminate those when formulating the Raban. Therefore, we collected eighty challenges at the focus group discussions conducted with the agile practitioners of the company and identify solutions for those challenges. Out of 80 challenges, 15 burning challenges were selected based on the priority assigned by the agile practitioners during the focus group discussions to identify solutions and best practices. We collected the data from an online survey. After that, we developed a

taxonomy adhering to the findings. Details on survey questions are given in Appendix I. We then conducted semi-structured interviews with five SMEs to check the applicability of the proposed recommendations. We finalized the solutions and best practices for the 15 burning challenges based on feedback.

Eighty challenges were analyzed using the Straussian grounded theory [41]. Using the constant comparative method, we identified six categories of challenges: *lack of agile mindset*, *inconsistency in story estimation*, *client management issues*, *lack of adherence to agile practices*, *scope change in requirement freeze*, and *lack of quantitative measurement*. As a solution for the challenges categorized as a *lack of agile mindset,* a four-day scrum master workshop was conducted to enhance the awareness of agile principles, values, and the scrum framework. It was followed up by a one-hour assessment, where the company issued a certificate for the successful completion of the assessment. Only the participants who passed the assessment were included in the RPA team for phase 3. To provide solutions for the challenges identified under *inconsistency in story estimation*, we identified a set of steps to follow in use-case estimation with the consultation of RPA SME, QA lead, and technical lead. For each use case, RPA SMEs estimated the number of automation steps, tool complexity, and logical complexity with the assistance of the project team.

Six challenges were listed under the *client management issues* category. Hence, to improve customer expectations, we studied five design thinking practices. There were 77 organizations listed on the SLASSCOM website [47]. Investigate through LinkedIn profiles of respective companies that approve that only 50 organizations were into IT services. Out of the 50, 26 organizations had more than 50 employees. We understood that it is unlikely for an organization with limited resources to practice design thinking techniques at the start-up stage; hence, we focused on companies with more than 50 employees. From those 26 companies, we identified 10 who engage with design thinking practices using the snowballing method. We conducted interviews with business analysts, project managers, and leads. The semi-structured interview

questionnaire was developed based on the literature survey. Details on questions are given in Appendix II. After selecting three project stakeholders (business analysis, project manager, and Tech lead) from the industry, a pilot interview was conducted to validate the questionnaire. The word 'Tech lead' in this thesis represents the technical lead for the project. We analyzed the data using the Straussian grounded theory approach [41] to give meaning to develop a theory. We started working on the open coding after each interview without waiting long. After that, the textual contents will be analyzed and categorized into sections. Then identified the key terms. Those key terms were used to identify the categories. We identified 10 pain points while satisfying customer expectations and 15 techniques to understand customer expectations. Moreover, we identified best practices to satisfy customer expectations, which we used in formulating the software implementation process for RPA projects.

3.5 Step 4 - Execute RPA projects in the novel process

In Step 4, we execute the RPA projects in the formulated software implementation project for RPA projects named the Raban framework. Step four was spanned three months. Several team members that worked in phases I and II resigned, citing the heavy workload and frustration from the high number of change requests and bug fixes identified during the UAT. However, we still had 23 resources for phase III as new members were recruited to the team, including the project manager who played the role of the scrum master. We formed two teams to develop two bots in parallel. The selected business processor was similar in complexity to phases I and II in steps two and three.

After that, we derived a DST to identify CBP to transform the RPA projects. Literature identified 25 factors impact on CBP. Then interviews with RPA SMEs helps to limit the list to 16 factors. To predict the outcome of the RPA transformation, an online survey was conducted. The potential respondents were identified using snowball sampling. The findings were used to identify the significance of the 16

identified factors to select the CBP transformation. Appendix III contains information on survey questions. To developed the two-class decision forest-based classification model was used. The proposed DST was executed in three RPA projects identified within the company.

3.6 Step 5 - Derive metrics to manage RPA projects run on the novel process

In Step 5, the focus was to identify the metrics to track and manage the RPA projects run using the Raban framework for successful project delivery. Raban adopted it based on the scrum framework. Hence, we first looked at the metrics tracked and measured in scrum projects.

Eight challenges were listed under the *lack of quantitative measurement* captured in the focus group discussions conducted with agile practitioners at the company. Mixed-method was used to conduct the research, combining qualitative and quantitative research methods. We used literature review, focused group discussions, and face-to-face interviews. Moreover, the constant comparative method in Straussian grounded theory was used for data analysis. Semi-structured interviews were conducted with nine scrum masters who handled nine scrum projects. Thirty-two metrics were captured during the semi-structured interviews with the scrum masters. The literature review identified 49 metrics. Data analysis using the constant comparative method was conducted for the identified 81 metrics and 58 challenges.

With the best-fit metrics identified for scrum projects, we conducted face-to-face semi-structured interviews with 10 SMEs in the agile discipline from 10 different companies to analyze and identify the must-track metrics to measure and manage scrum projects. We identified the companies with scrum practice using the snowball sampling method. Agile SMEs highlighted the necessity of the scope change metric. It is complicated to complete a story when user stories need to be frequently updated after sprint requirements freeze. Adding lack of clarity stories to the active sprint violating the sprint scope by the client is also a problem. With the review feedback, we again conducted the data analysis. Finally, we identified 17 metrics (i.e., 15 existing

metrics and two new metrics) and three existing artifacts. One metric and one artifact should be measured in the speculate phase. Another three metrics and two artifacts should be tracked in the explore phase. Further, seven metrics and two artifacts should be measured in the inspect and adapt phase. Moreover, two metrics should be tracked each month, while two other metrics should be tracked each quarter. We validated the metrics and artifacts by tracking and measuring them in five scrum projects of the global IT services and delivery company.

To identify how those metrics could be adopted in RPA projects, we first collected the project team's preference on which metrics could be used in RPA projects executed using the Raban framework. For this, we conducted two brainstorming sessions with the project team and RPA SMEs in the company. After that, we conducted focus group discussions with the project team, RPA SMEs, and process consultants and finalized metrics for RPA projects run on the Raban framework. Here we considered customizing the 17 metrics and three artifacts identified for scrum projects. Finally, based on the feedback, we identified 14 metrics and two artifacts to track and manage the RPA project run on the Raban framework. Eight metrics measure the product quality, three metrics and one artifact to measure the team productivity, three metrics, and one artifact measure project predictability.

Moreover, we figured out at what stage we needed to measure those. Consequently, we identified one metric and artifact to be measured in the speculate stage. Two metrics and one artifact should be measured in the explore stage. Five metrics could be measured in the inspecting and adapt stage. One should be measured monthly, while two metrics should be measured quarterly. We validated the metrics by tracking and measuring those metrics in two RPA projects.

3.7 Summary

This research focused on identifying the process template for RPA projects. We followed a five-step process to achieve it. At first, we conducted an RPA project

in a plan-driven process. Next, we used an in-house development process. We formulated a software implementation process when we realized the existing software implementation processes were unsuitable for RPA projects. Then, we derived metrics to track and manage the RPA projects run on the Raban framework. Moreover, we identified the minimum number of metrics to track and manage scrum projects for product quality, project predictability, and team productivity. Further, we developed a DST to select a CBP for RPA transformation. We derived a taxonomy from selecting the solution or the best practice for the challenges faced by agile teams. Further, we derived a tool to estimate RPA use-cases and identified methods to improve customer satisfaction.

4 FORMULATING SOFTWARE IMPLEMENTATION PROCESS FOR RPA

The waterfall model, V-model, spiral model, iterative, and incremental models are different software implementation processes. To understand the most suitable software implementation process for RPA projects, we conducted a comprehensive literature review, industrial survey, and semi-structured interviews with the IT services company's project teams. The literature and nine scrum masters and RPA SMEs we interviewed confirmed the unavailability of a suitable software implementation process for RPA projects. Hence, at first, we executed the RPA projects in the well-known waterfall model process, and the results showed the waterfall model is not a good fit for RPA projects due to the large count of change requests and defects identified during the UAT. After that, RPA projects were executed in an in-house developed model where there was no reduction to the count of change requests, and the client reported defects captured during UAT. Hence, to overcome this issue, we decided to formulate the software implementation process for RPA projects. This chapter discussed the steps to formulate the software implementation process for RPA projects.

Section 4.1 presents a background study of the existing software implementation processes we analyzed. RPA project execution in a plan-driven process and an in-house developed development process are explained in Section 4.3 and 4.4, respectively. Section 4.5 describes the way forward to identify a suitable agile practice to adopt in RPA projects. Section 4.6 described the formulation of the software implementation process for RPA projects. Section 4.7 has listed the results and analysis. Finally, section 4.6 summary section concludes the chapter.

4.1 Background study

As per [7], first attempt of RPA projects fails in 30%-50%. Further, Lamberton [7] has identified 10 common issues for such failures. Four of them can be attributed to business reasons, and the rest can be attributed to project challenges. One of the critical issues is applying traditional software implementation methodologies to execute RPA projects. Moreover, face-to-face interviews conducted with industry experts identified that the industry is new to RPA technology to discuss the software implementation process for RPA projects. The company has defined process templates for development, maintenance, and testing only in the waterfall and agile discipline and service delivery projects. Agile or waterfall model process templates for development projects are used for scratch developments, where the requirements frequently change [33]. RPA transformation is not possible for such projects. RPA projects neither accept modifications to the existing codebase nor fit the maintenance template. Also, not only a testing project because development happens in RPA projects on top of a stable code base where rules were defined correctly [2]. Therefore, existing process templates were unable to use in RPA projects. Moreover, face-to-face interviews conducted with RPA SMEs approved the unavailability of the process template or execution process for RPA projects. Hence, in Step 2, we execute the RPA project in the plan-driven process and, after that, in an in-house developed process. This thesis uses the terms ‘plan-driven process’ and ‘waterfall model’ to denote the same purpose.

4.2 Executing the RPA process using the plan-driven process

The first phase of bot implementation was based on the plan-driven process and took eight months to complete. The project team consisted of 23 team members: one PM, two business analysts, one architect, 13 developers, and six Quality Assurance (QA) engineers. In the requirement gathering phase, one business analysts and an architect were onsite with the client, and the rest of the team was offshore. The business

analysts and architect conducted requirement gathering workshops at the client site with operational system users (aka. process executors) who demonstrated their existing process. The client also conducted a system demonstration once, and the business analysts captured screenshots of business process workflows. When the requirement gathering was completed, the requirement document was developed, and approval was obtained from the client leadership. In Phase-I, three bots were developed for the selected business process. At the design stage, the estimation was completed, as well as the architecture design and QA test plan were finalized and signed off by the client. Two bots were implemented parallelly as two projects. Once the implementation was completed, deliverables were shared with the QA team for System Integration Testing (SIT). The deliverables were then shared for the UAT. Once the UAT was completed, the client conducted Production Verification Testing (PVT) in the pre-production environment. Table 4.1 shows the project execution schedule for implementing the two bots. Nine to 10 weeks were spent completing end-to-end functionalities. The word 'Dev' in this thesis represents the development team.

Table 4.1 : Project execution schedule.

w/c	**December**			**January**					**February**		
	12	**19**	**26**	**2**	**9**	**16**	**23**	**30**	**6**	**13**	**20**
w/e	**December**			**January**				**February**			
	16	**23**	**30**	**6**	**13**	**20**	**27**	**3**	**10**	**17**	**24**
Week	**4 th Week**	**5th Week**	**6th Week**	**7th Week**	**8th Week**	**9th Week**	**10th Week**	**11th Week**	**12th Week**	**13th Week**	**14th Week**
Construction	Dev	Dev	Dev	Dev							
SIT				SIT	SIT	SIT					
UAT						UAT	UAT	UAT			
Code Freeze									PVT	PVT	
Construction	Dev	Dev	Dev	Dev							
SIT			SIT	SIT	SIT	SIT	SIT	SIT			
UAT					UAT	UAT	UAT	UAT	UAT	UAT	
PVT											PVT

4.3 Execute the RPA process using the in-house development process model

Phase II in Step 2 spanned for 12 months and implemented another three bots for the selected business process. The project team remained the same. The in-house process model (see Figure 4.1) proposed by the company was used to execute the RPA projects. The in-house process model had three stages: identification, bot implementation, and deployment. We identified the business process candidate for RPA implementation during the identification stage, verified its technical suitability and RPA suitability for the identified business process, and conducted an environment readiness check to verify the bot development and testing environments' stability. Both technical suitability and RPA suitability were verified through a checklist prepared by the company's process consultants. The environment readiness check was executed with the help of the client's maintenance team.

Then, the project team moved on to the bot implementation stage. We derived the business process based on the finalized solution architecture. Then, we identified the main flows and sub-flows in the selected business process and derived the use case diagrams. Once the QA team completed the test cases, developers started the bot implementation. The test team executed the SIT once they received the QA deliverables. If any bug was identified during the SIT, they were fixed simultaneously. Hence, the process is more like test-driven development. However, the developers did not wait for the test cases to start development. They used high-severity test cases as unit test cases. Once SIT was completed, deliverables were moved on to UAT. Finally, when UAT is completed, deliverables are moved into the production environment in the deployment stage. The client and his technical team managed the deployment stage.

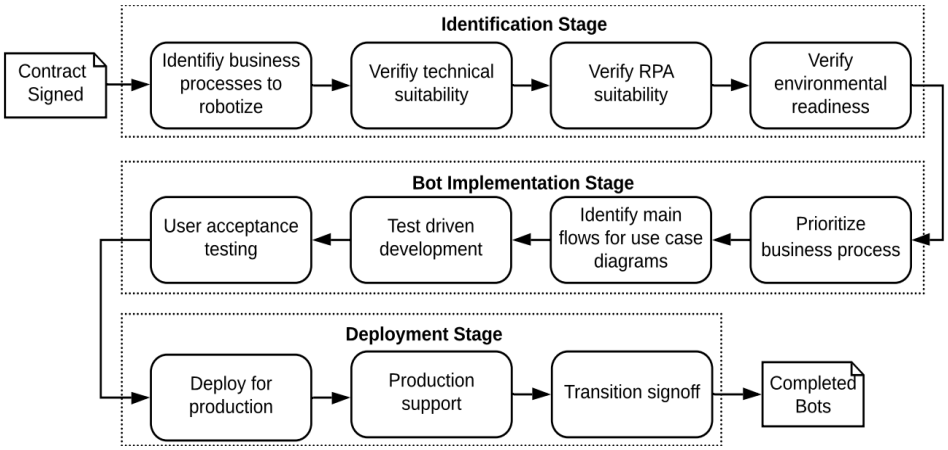

Figure 4.1 : In-house process model proposed by the global IT services and delivery company.

4.4 Identify suitable agile practices to adopt the RPA process

29 and 20 change requests, and 32 and 10 defects were identified at UAT in phases I and II in Step 2, respectively. Therefore, the project team could not deliver the bots within the stipulated time and budget in both phases. Most of the change requests were due to the changes made to the UAT environment by the client's maintenance team without informing the project team. We identified it as a communication gap between the client's maintenance team and the project team. Moreover, several rarely used business flows were identified during the UAT, which the client forgot to mention during the requirement elaboration stage in Steps 1 and 2. The plan-driven and in-house developed models were ineffective in addressing such communication issues and later revealed requirements. We could not identify an alternative software implementation process from the literature, so we decided to develop a new model for RPA projects while building on best practices. In developing such a model, we set the following objectives:

1. Identify the best way to capture requirement changes early in the bot implementation process.
2. Identify the best way to capture missing requirements early in the bot implementation process.
3. Reduce additional costs due to change requests identified during the UAT phase.

The fourth agile value, "responding to change over following a plan," welcomes requirement changes [15], which was the central issue during the bot implementation. Therefore, we choose to derive the new model by extending agile practices. However, first, we had to determine whether agile practices suit RPA projects. Therefore, we first assessed the agile suitability using the checklist developed by the global IT services and delivery company's process team. The checklist was designed to assess the agile suitability of a new project using 41 questions. These questions assessed the status of the client environment (e.g., development, testing, and production environment), client partnership (i.e., the strength of partnership with the client), client engagement (i.e., client's continuous engagement with the team), requirement/solution (i.e., clarity of the solution), and team engagement (i.e., engagement of the project team). We generate the radar chart in Figure 4.2 for the chosen RPA project based on the project manager's answers. The solid line illustrates the suitability of agile practices, while the scattered line indicates the risk. Higher values indicate high suitability or high risk. Team engagement, client engagement, and client partnership favor applying agile practices and having low risks. However, the requirement/solution is less favorable and higher risk. While the client environment is highly favorable, the risk is relatively high. However, while the agile suitability tool was designed to assess new product development projects, RPA projects run on top of a stable product with limited to no changes made to the product's requirements. Furthermore, relatively more minor client engagement is required for RPA project execution. Therefore, we could conclude that agile practices are suitable for

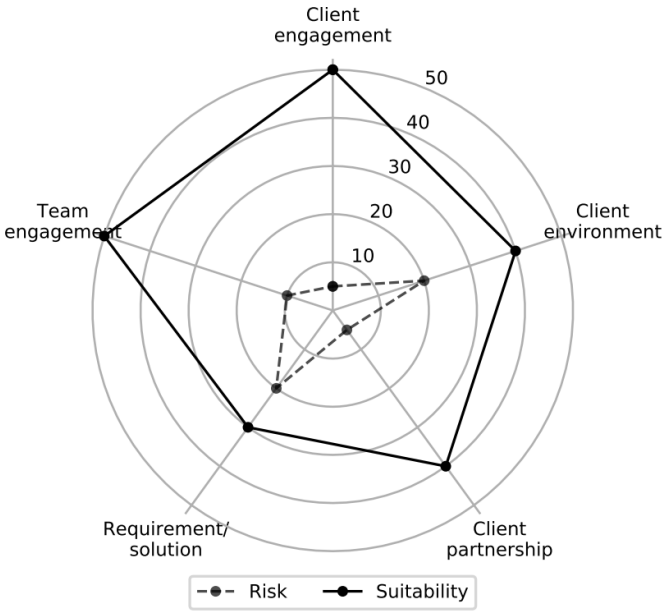

Figure 4.2: Agile suitability estimation using the global IT services and delivery company developed tool.

implementing RPA projects regardless of the client environment's relatively high risk and requirement/solution in RPA projects.

We used the agile suitability tool proposed by [9] further validate the agile suitability tool used by the global IT services and delivery company. The tool was developed based on the results of literature review and validated case-based research. It assesses a project's agile suitability based on eight categories: teamwork, requirement, planning, technical practices, quality, culture, knowledge creation, and outcome. The agile suitability of the selected RPA project measured using the tool is illustrated in Figure 4.3. The solid line illustrates the global standard identified by the authors using an industrial survey. In contrast, the dashed line shows the status of the RPA project, which was assessed with the help of the project manager. Other than culture and requirement, all other categories are comparable to the global standard. Culture and requirements do not meet the global standards, as RPA projects expect

minimal changes to the stable codebase. Therefore, we could conclude that agile practices suit RPA implementation projects.

Given the agile process's potential suitability for RPA projects, we investigate a suitable agile practice to adopt for RPA projects. Under the agile umbrella, several agile practices such as scrum, kanban, and XP are available [28], [17]. We selected the scrum framework, as it helps to manage rapidly changing requirements [10], [11]. Moreover, scrum is a lightweight process, as it reduces the overhead of the process and maximizes the time available to get helpful work done [17]. Furthermore, it has been used in sophisticated software and product development projects and is known to significantly increase the productivity and time to market relative to the plan-driven process [28], [11]. Scrum is the most widely used process framework among agile practices [10], [11]. Therefore, we selected the scrum framework as the basis for the proposed implementation process. The scrum guide [17] explains that each sprint in a scrum starts with backlog grooming and sprint planning. Implementation begins as soon as the sprint backlog is ready. Once the implementation is complete, sprint reviews and sprint retrospectives are conducted. The daily scrum is conducted for 15 minutes. The business process owners and stakeholders participate in the ceremonies except for the daily scrum and sprint retrospective.

However, the scrum framework cannot be used for RPA projects due to the fundamental differences in product development and RPA projects. For example, continuous backlog grooming sessions in scrum are not required in every sprint, as bots typically interact with a stable system/codebase. In product development, the business process owner is concerned with the feature changes made to the product. However, the timely completion of the bot matters in RPA. A partially completed bot does not provide business value, as it is expected to mimic a complete task performed by a user. Furthermore, instead of user stories in the scrum framework, the RPA project team develops use-case backlogs. Backlog grooming, release planning, and sprint planning ceremonies in scrum are handled by the business process owner with the project team. In RPA projects, those meetings are conducted by the scrum master with

the RPA SMEs' involvement and technical or QA leads. Due to the conflicts mentioned above, we cannot use the scrum framework for RPA transformation projects. As a solution for that, we adhere to the Scrum framework to formulate the software implementation process for RPA projects. We elaborate more on that in the next section.

4.5 Formulate the software implementation process for RPA projects

The company has appraised at CMMI (Capability Maturity Model Integration) level five in the product and service delivery process. Hence, the company has well-defined software implementation processes for development, maintenance, testing only, and service delivery projects. While adhering to the scrum framework in building a new software implementation process for RPA projects, we researched challenges, best practices, and malpractices in scrum projects. Focused group discussions were

Figure 4.3: Agile suitability estimation using Jalila's tool.

conducted with an agile practitioner in the company to identify the agile teams' challenges, best practices, and malpractices and eliminate those when formulating the software implementation process for RPA projects. We had separate three focused group discussions with the SMEs to limit the no of participants and maintain the continuous attention and comfortable environment. Focused group discussions were facilized by the company's agile enterprise coach. We identified eighty challenges once we listed all the challenges, best practices, and malpractices. Analysis was conducted using the constant comparative method. *The categories identified include lack of agile mindset, inconsistency in story estimation, client management issues, lack of adherence to agile practices, scope change in requirement freeze, and lack of quantitative measurement*. 10, 11, 8, 36, 7, and 8 challenges were identified, respectively. Before working on process formulation for RPA projects, we researched the 80 challenges to eliminate those in formulating novel processes for RPA transformation projects. As a first step, we identified the 15 prioritized challenges and developed a taxonomy to provide solutions for those challenges, briefly described in the next section.

4.5.1 Taxonomy for Agile Practitioners

To identify best practices and malpractices in the scrum and to identify solutions for the challenges identified in scrum projects, we conducted focus group discussions with the collaboration of agile practitioners in the company. We collected and prioritized eighty challenges and identified 15 burning challenges based on the agile practitioners' preference to provide immediate solutions. Table 4.2 illustrates the 15 burning challenges shortlisted out of 80 challenges. To find solutions for those challenges, we conducted a literature review and then developed an online questionnaire (see Appendix I) to verify any missed challenges and malpractice and capture respective solutions. Then we share an online survey with agile teams and practitioners in the industry. With the data analysis results, face-to-face interviews

were conducted with five agile SMEs with over 10 years of agile experience identified from the industry. Details on survey questions are given in Appendix I.

Table 4.2: Challenges faced by Agile Practitioners in scrum projects.

Challenges
1. Conducting retrospectives at the end of each iteration does not value the software implementation project.
2. As per the mentioned frequency (every week), backlog grooming has not been conducted.
3. Requirement changes accept during the sprint execution.
4. Acceptance criteria in the user story have not been defined adequately.
5. Definition of Done in the project has not been defined adequately.
6. A user story has not been defined adequately.
7. Working in a geographically distributed environment was not much acceptable to the scrum team.
8. Lack of knowledge of pair programming.
9. Lack of knowledge of quality standards (ISO, CMMI, and other maturity models)
10. Lack of knowledge of agile measurement criteria.
11. Team building activities do not conduct iteratively.
12. Impact on internal politics.
13. Lack of required time to complete QA work with the expected quality.
14. Waste of time on updating the automation team and regression test suite.
15. QA spillovers in most of the sprints.

We proposed solutions and best practices to overcome the challenges and malpractices based on the feedback received. Finally, the findings were summarized in taxonomy, as illustrated in Table 4.3. In the taxonomy, challenges were categorized into four categories: lack of awareness, scrum principles violation, time management, and team collaboration. For each category, challenges were listed in the respective row. The solutions are listed under the column named ‘Solutions.’ The solutions for each question were mapped with a number. We conducted semi-structured, face-to-face interviews with five selected SMEs to check the usability of the proposed taxonomy. Once we provided the solutions for 15 burning challenges through

taxonomy, we focused on *client management issues* with eight challenges, elaborated in detail in the next section.

4.5.2 Design thinking to accelerate customer satisfaction.

We selected the qualitative and inductive reasoning methods to find solutions for the six challenges listed in the *client management issues* category. First, we explored the industry's awareness of design thinking practices. We planned to derive solutions based on design thinking practices. SLASSCOM has listed 77 websites [47]. Figure 4.4 explains how seven organizations were identified for research out of 77. Digging through the LinkedIn profiles we were able to identify 50 organizations with IT services. Only 26 organizations found with more than 50 employees. For companies below 50 employees, practicing design thinking techniques is not practical. Very hard if the company is at the start-up stage. We identified 10 organizations that practice design thinking through the snowballing method. In an organization, project managers, quality assurance or technical leads, and business analysts are the main contributors in decision making, project management, and requirement gathering. Therefore, we conducted face-to-face interviews with the employees who play above mentioned roles. Literature review covered five elements in the design thinking practices. Face-to-face questionnaire developed considering the literature review analysis [36]. Human-centered approach, thinking by doing, visualizing, synthesis of diverging & converging, and collaboration are design thinking practices [35]. The [35] study states that design thinking practice is one element out of three elements in the three-dimensional framework. Consequently, we identified that project stakeholders are unfamiliar with the word "design thinking". However, they are following techniques in design thinking. Due to that, we revised the questions to satisfy the expectations based on the feedback received. Table 4.4 describes the age, designation, experience, responsibilities, and no projects handled by the fifteen managerial-level employees. Ten service-based software development organizations identified those applying design thinking practices to the agile process. We selected vital project stakeholders who can negotiate with the customer because they have the decision power. We first

introduced the design thinking concept during the interviews because they were clueless about the design thinking concepts. However, they practice those in their product implementation activities. Straussian grounded theory [41] is used for data analysis because it supports literature review to derive new concepts and theoretical sampling. We immediately completed the open coding process at the end of each interview and analyzed the contents with text. After that, we broke the data into parts then compared it to identify the key terms to add them into the categories. Key terms are compared continuously with the collected data and identify new inclinations till they reach theoretical saturation. Relationships and disaggregation of the open coding were identified using axial coding and finally identified the factors using selective coding.

Next, we focused on solving the seven challenges listed in the *scope change in the requirement freeze* category. In that, we developed a use case estimation model for Raban projects.

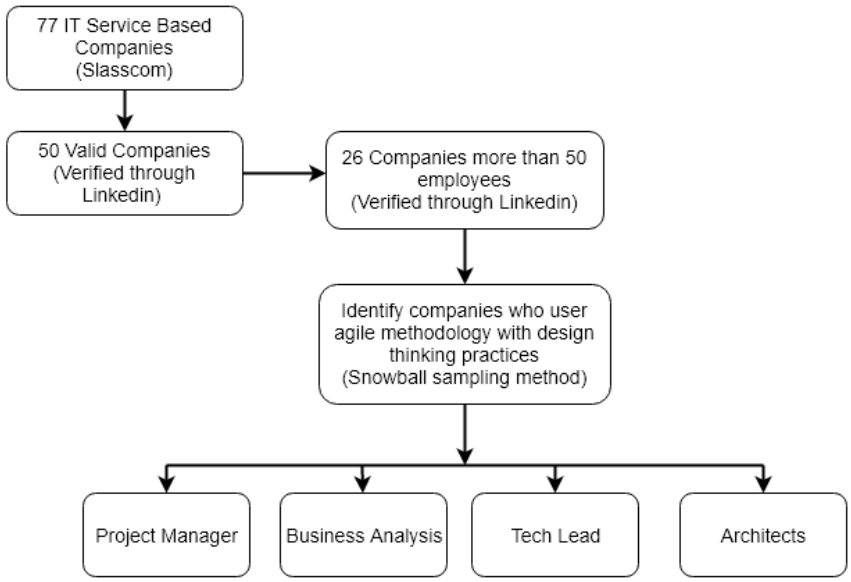

Figure 4.4: Steps followed to sample selection (10 companies) out of 77 IT service-based companies.

4.5.3 Use-case estimation model

During the constant comparative analysis of 80 challenges captured in focused group discussions, we were able to identify nine challenges under the *inconsistency in story estimation* challenge. The issues have been raised, focusing on the scrum process. However, since our focus was on formulating a software implementation process for RPA projects, we studied identifying a proper estimation process for use-case estimation in RPA projects. Because RPA is a new technology, we conducted brainstorming sessions with the RPA SMEs to derive a proper RPA use-case estimation process. As a result, we derived an RPA use-case estimation process. At first, identified the number of automation steps for the selected RPA use case. Then, we estimated the number of automation steps, tool complexity, and logical complexity for each step. Tool complexity was identified based on the automation tool used in RPA transformation. The complexities were introduced as Low, Medium, High, and Ultra High. For each complexity, the effort was derived, considering the tool complexity and logical complexity separately. Moreover, we address the *lack of quantitative measurement* and *scope change in requirement freeze* along with eight and seven challenges, respectively, in chapter 6. Therefore, in the next section, we discuss the execution of the Raban framework in phase III.

4.6 Execute RPA projects in the Raban framework

The project was spanned three months. Six team members that worked in phases I and II in Step 2 was resigned, citing the heavy workload and frustration from the high number of change requests and bug fixes identified during the UAT. However, we still had 23 resources to execute the project using the Raban framework as six new members were recruited to the team, including the project manager who played the role of the scrum master. Moreover, all the team members had no prior experience with the agile process. Therefore, workshops were conducted to provide relevant

training. We formed two teams with 12 members each and selected a business process with similar complexity wherein each project, three bots were developed in phases I and II in Step 2. Figure 4.5 illustrates a high-level overview of the proposed Raban framework. Raban has five phases, namely, selection, ceremony, explore, inspect, and deployment. The client maintains the prioritized business process backlog to identify the business processes to be automated based on their business priorities. Once the client prioritizes the prioritized business process backlog, the selection phase of Raban is initiated. In this phase, the RPA project team verifies and ranks each business process using the RPA suitability DST and environmental readiness. Then the most fitting business process is advanced to the ceremony phase. This phase consists of backlog grooming, estimation, and sprint planning ceremonies. Based on these ceremonies, a sprint backlog is developed, which is used to guide the bot implementation during the explore phase. Next, an inspection of the sprint, sprint retrospective, inspection of the completed sprint work, and sprint reviews are performed during the inspection phase. If the inspection is successful, bots will advance to the deployment phase. Alternatively, based on the inspection outcome, we may need to iterate back to the ceremony phase to update the product backlog or explore the phase to update the sprint backlog.

Next section, we elaborate in detail on our proposed Raban framework.

Table 4.3 : Taxonomy developed for the challenges.

Category	Challenge [Solution]					
Time Management	Allocated time is not sufficient to complete QA work on time. (1)		Additional time required to update the Regression test suite. (2)	Additional time send on to keep automation team UpToDate. (3)	Additional time required to update the automation test suite. (4)	
Lack of awareness	Does not consider on quality standards/ metrics. (5)			Scrum team's lack of knowledge on pair programming (6)		
Scrum Principal Violation	Retrospective ceremony is not conducted at the end of sprint. (7)	Backlog grooming is not conducted to groom the backlog. (8)	Definition of done is not defined correctly. (9)	Accepting changes during the sprint. (10)	Story is not correctly defined. (11)	Acceptance criteria is not correctly defined. (12)
Team Collaboration	Not conducting any team-building activity. (13)		Has internal politics. (14)	Scrum teams are not preferring to work in geographically distributed environment. (15)		
Solutions						
(1)	1. Conduct pair programming. 2. Always automate the early sprint (sprint n-1). 3. Start to build the automation component. However, execute when only code stability is achieved. 4. Use QA test cases for unit test on developer environment.					
(2)	1. Write test cases to align the business flow. (re-usable business component model.) 2. Make sure to automate only the rarely changing user journeys. 3. Spreadsheets are better option to easily update test cases. 4. Automation should start on top of stable environment.					
(3)	1. Better to follow the test case writing structure used by manual testers for Automation team as well. 2. Manual test cases changes should manage in a separate catalog.					
(4)	1. Fully automated environment provisioning process, considering the end-to-end deployment service.					
(5)	1. Conduct knowledge improvement sessions for quality standards and metrics. 2. Get knowledgeable on behavioral driven development. 3. Extend the pair programming knowledge. 4. All the resources should participate for the trainings conducted within company.					

Solutions	
(6)	1. Conduct workshops and training sessions on pair programming.
(7)	1. Retrospective can be conduct in new style. 2. Retrospective can be conduct along with another meeting. (Ex. Conduct metric analysis)
(8)	1. Frequency of conducting Backlog grooming can be reduced with time when project progress. 2. Use overlap time to conduct backlog estimation. 3. Use transparent project management tool for project communication. 4. Pre-backlog grooming meetings can be scheduled with the dev team before conduct with the project stakeholders. 5. Conduct post-backlog grooming meeting with the onsite/dev team. 6. High-quality audio and video recordings of the backlog grooming can be share with offshore team. 7. Business analysts act as an agent. He could conduct backlog grooming with the client and dev team separately. 8. Elaborate user stories with the core team and client at onsite. Then share the same information with offshore team.
(9)	1. At the contract signoff in work agreement document, definition of done should define with the agreement of the product owner.
(10)	1. Good amount of time (e.g., 6 weeks) spend to derive a workable release plan. 2. Core team and client onsite team should engage with story elaboration and then that sync with the offshore team. 3. Product owner should facilitate the Set-up ground rules with the product owner. 4. No requirement changes within the sprint.
(11)	1. Conduct a workshop to improve story writing. 2. Better to have an allocated location to manage the stakeholder feedback during product demonstration.
(12)	1. Better to have an allocated location to manage the stakeholder feedback during product demonstration.
(13)	1. At the end of every sprint or end of each month better to have team-building activity.
(14)	1. Propose innovative sprint. 2. At the release end plan for internal games. 3. Conduct retrospective using innovative games.
(15)	1. Ground rules setup with a geographically distributed team (Ex. if development team and testing team from different geography, agreement made as such, defect is count as a defect if the developer does not fix the defect within a day of recording) 2. Daily standup can conduct on overlap times for respective geo. 3. Keep open the communication bridge to talk any time. 4. Conduct and complete scrum for 15 min. 5. Setup a different meeting to discuss other issues. 6. Introduce games (e.g., crown awarded to the person who completes maximum number of story points). Improve soft skills of the resources. 7. Use collaborative tools (e.g., WebEx)

Table 4.4 : Pain points in customer satisfaction and ways to improve.

Pain Points in Satisfying Customer Expectations
• Unable to prioritize the customer expectations. • Evaluate the competitor market, compare, and frequently update the product features irrespective of features applicability. • Not predict for the future market when deriving features. • Contractually agreed with fix bid projects (budget, cost, and time are fixed). • Understanding about unwanted features only at the completion of the project. • Try to do requirement changes within limited budget. • Considering very new technology without knowing whether they survive or fail. • Working on the product which does not absorb by the market. • Customers want to engage only with domain experts and subject matter experts. • No strong believe on vendor.
Techniques to Understand Customer Expectation
• To list down the customer and end-user wish list at the project initiation it is always advisable to conduct workshops. • Work on roadmap development for project implementation. • Conduct frequent reviews and demos (e.g., proof of concepts and prototypes). • Conduct formal and casual discussions frequently. • Interview an end user of the planned product development to understand his/her paint points. • Analyses customer pinpoints. • Pre-sales is the good place to study the customer. • Small and frequent deliverables with expected features. • Project blockers should communicate to the customer early as possible. • Product implementation should conduct after conducting the market validation. • Enhance the knowledge on simplicity for product implementation in client's mind. • Form the foundation of the solution for the derived business case. • Identify the non-functional requirements. • Update the customer on product implementation progress through artifacts like., Burndown charts, show and tells. • Study the existing process for its drawbacks.

4.7 RABAN: A software implementation process for RPA projects

We derived the name of the proposed framework by combining “RPA” and “ban” (inspired by practices such as kanban and scrumban under the agile umbrella). However, we dropped the letter “P” to make it “Raban”, which refers to a drum used in Sri Lankan traditional music. We consider Raban a framework, as the proposed model can be customized based on the RPA project’s context. As illustrated in Figure 4.6, the Raban framework has five phases: selection, ceremony, explore, inspect, and deployment. These names are chosen to reflect the work carried out within a phase. The process starts with the business stakeholders setting the prioritized business process backlog. We select the candidate's business process and the RPA tool for the project during the selection phase. Next, scrum ceremonies, such as business process elicitation, estimation, release, and sprint planning, are performed in the ceremony phase. However, sprint retrospective and sprint review ceremonies are not performed in this phase, as those are used to inspect and adapt the robotize-sprint work, which happens after the sprint. The development of the bot happens in the explores phase. Inspection activities such as sprint retrospective and sprint review are carried out in the inspect phase. The UAT conducted at the deployment phase. Once the UAT is successful, the bot is deployed to the production. Finally, the PVT is conducted. Figure 4.6 illustrates a more detailed view of Raban, including sub-phases and their outputs.

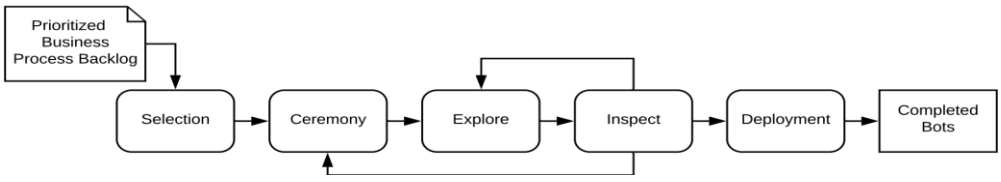

Figure 4.5: High-level overview of the Raban framework.

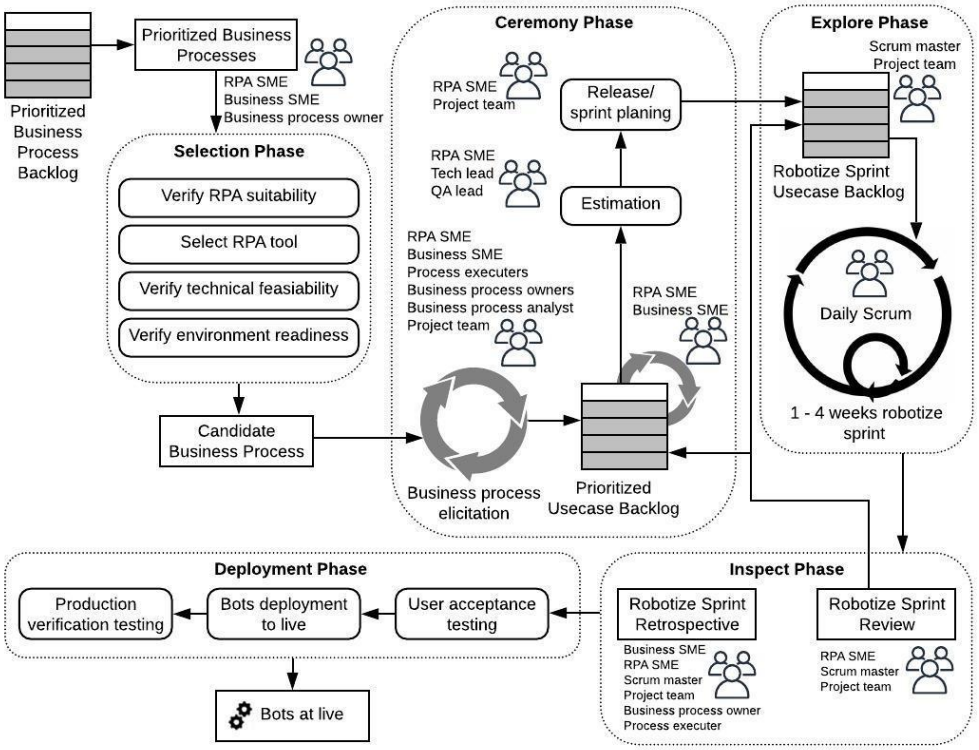

Figure 4.6: Detailed vide of the Raban framework.

The 12th principle in the agile manifesto has a statement such as; “The team become more effective at regular intervals where they tunes and adjust their behavior accordingly” [15]. A sprint in Raban is referred to as a robotize-sprint. Like a sprint in the scrum framework, a robotize-sprint could span two, three, or four weeks of regular intervals of development cycles (see Figure 4.6). In RPA, the business process owner(s) owns only the prioritized business process backlog and prioritized use-case backlog, whereas the project team owns the robotize-sprint use-case backlog. Hence, in a robotize-sprint, the business process owner and the business SMEs join only at the sprint review meeting. The project team prioritizes business process backlog estimation, release planning, robotize-sprint planning, and daily standup. Next, we discuss each phase in detail.

4.7.1 Selection phase

The selection phase's objective is to ensure that the project team transforms the most suitable business processes using a proper RPA tool. Moreover, the selection phase verifies the technical feasibility and the readiness of both test and development environments to start the RPA transformation. We introduced a prioritized business process backlog for business stakeholders to manage and prioritize the business processes considered for the RPA transformation. Selecting on CBP for RPA transformation depends on the volume of transactions, rate of change, and business process complexity. When stakeholders have list of CBP, each needs to be verified for its RPA suitability [48]. Based on the project team's assessment of the RPA suitability of each CBP, the priority of the business processes chosen for the RPA transformation might be different from the initial prioritized business processes. Moreover, as we have already assessed the agile suitability of RPA projects, the agile suitability of each CBP does not need to be verified. Hence, the Raban framework does not dependent on the business process. RPA SMEs, business SMEs, and business process owner(s) could use a suitable DST to verify the RPA's suitability for the prioritized business processes. For example, we presented a DST to determine RPA suitability based on the two-class decision forest classification model [48].

Next, identify the RPA tool for the project. The chosen RPA tool should be capable of converting all the end-to-end steps of the identified business process. The chosen tool needs to fulfil the technical and commercial features such as scalability, rapid deployment, control management/ security, analytics, and flexibility in pricing. The literature identified these features and verified them with RPA SMEs' help. Face-to-face interviews conducted with a project manager, a business process analyst, and two technical leads involved in the RPA project. As a result of that, it was decided that RPA tool selection [49], RPA process feasibility study [50], [3], and environment-readiness guidelines [50], [7] could be used to identify a suitable tool. Next, assess the technical feasibility of the business process with the help of the RPA SME. Business SME and business process analyst present the CBP walk-through to the RPA SME and

answer any questions raised during the session. Then the RPA SME analyzes the feasibility to convert each CBP into a technical solution using the selected RPA tool. If required, RPA SME may develop a proof-of-concept solution with the help of the project team. Considering the technical limitations, the RPA SME analyses the extent to which the business process could be automated. Finally, the RPA SME, in consultation with the business process analyst, discusses with the business process owner(s) areas where process conversion is not technically viable and finalizes the approach to be followed for each business process. It is essential to get the test and development environments ready for the RPA transformation because the bot to be implemented must interact with an existing system. It is the responsibility of the scrum master to get the test and development environments set up like the UAT environment with the help of IT teams (both client and internal) as it requires establishing remote connectivity as per the project requirements and procurement of relevant hardware and software. The project team and RPA SME should ensure that the development, test, and UAT environments have the correct data imported from the production environment.

4.7.2 Ceremony phase

Business process elicitation, estimation, release planning, and sprint planning ceremonies are conducted in this phase. Business process elicitation is conducted to understand the end-to-end functionalities of the CBP. It could be conducted in the form of multi-day workshops in the client environment (i.e., onshore) with business process owner(s), business SMEs, process executors or future bot users, business process analysts, and RPA SMEs. The RPA SME and business process analyst then demonstrate the CBP's end-to-end functionalities to the project team. The project team decomposes each scenario into smaller sub-tasks and develops use-case diagrams for the CBP's end-to-end functionalities. The project team should promptly clarify any grey areas related to the business process with the business process analyst, RPA SME, business SME, process executor, and business process owner(s). Even a minor misunderstanding or omission could lead to an incompatible bot. The project team

should use a transparent project governance tool (e.g., Jira) to develop the use-cases and build the business process backlog. The use of a transparent project governance tool enhances the visibility of the client. Once the business process backlog is developed, business and RPA SMEs should prioritize the use-cases. The business SME should also notify any day-to-day changes made by the client system maintenance team to the UAT environment. Meanwhile, with RPA SME's consultation, the technical lead should develop the solution architecture design while referring to the process definition document developed for the CBP. Similarly, the QA lead should develop the QA test plan. Finally, the solution architecture and test plan are presented to the business SME and business process owner(s) for feedback and initial sign-off. As the project progresses, both documents are updated whenever changes are made to the system's solution architecture and testing procedures.

However, continuous changes to the solution architecture and testing procedures are not expected as the RPA bot is implemented on top of a stable codebase. Next, in the estimation sub-phase, the use-cases in the backlog are estimated with the consultation of the RPA SME, QA lead, and technical lead. Estimate the number of automation steps, tool complexity, and logical complexity for each use case. Then conduct a separate estimation ceremony for each robotize-sprint to estimate the added use-cases or changes made to the use-cases. It is recommended to limit this meeting to an hour. Then the release and robotize-sprint plans are developed by the RPA SME and the project team. The release plan is developed considering the client's priority on bot completion. Subsequently, the robotize-sprint plan is developed based on the finalized release plan. Robotize-sprint is developed to match the project team's capacity. Otherwise, the project team might not complete the committed use-cases at the beginning of the robotize-sprint.

4.7.3 Explore phase

This phase aims to develop and test all the use-cases in the robotize-sprint use-case backlog developed during the ceremony phase. While the frequency of a robotize-

sprint could vary from two to four weeks, it should not be changed within the project life cycle once agreed. Daily stand-up (i.e., daily scrum), facilitated by the scrum master, should be conducted only 15 minutes during the robotize-sprint with the project team's participation. Technical or requirement clarifications should not be discussed during the daily stand-up. If required, such discussions should be conducted separately with the client. Partially completed bots should be demonstrated during the sprint review to get the business SMEs and process executors' feedback to minimize change requests and bugs that may arise later in the UAT. Moreover, sprint demonstration helps the business SME process executors experience and understand bots' behavior before being deployed into the UAT environment.

4.7.4 Inspect phase

Robotize-sprint retrospective and robotize-sprint reviews are conducted in this phase. During the robotize-sprint review, the project team demonstrates fully or partially completed sprint deliverables to the business process owner(s) and business SMEs. Business process owner(s) and business SME then accept the sprint work or provide feedback on any missing functionality. The robotize-sprint use-case backlog is updated based on the suggestions and change requests. Prioritized use-case backlog may also need to be updated when a change request results in a change in priority or severity. Robotize-sprint retrospective is conducted only within the project team to understand the improvements for the subsequent robotize-sprint execution, identify drawbacks or unwanted activities that could be eliminated, and appreciate the team's hard work on-time delivery of the sprint.

4.7.5 Deployment phase

Once the bot is implemented with end-to-end functionalities, the project team deploys the bots in the UAT environment (see Figure 4.6). Process executors conduct the UAT while trained on how to use the bot. Once all the bugs raised during the UAT are fixed, the client's IT team deploys bots into production for PVT. When the PVT is completed, bots go live.

4.8 Results and Analysis

The project team could not manage the change requests count in phases I and II in Step 2. Table 4.5 lists the root cause analysis results of the client-reported defects identified at UAT in the waterfall model process in phase I and the in-house developed model process in phase II. In phases I and II in Step 2, the client only had the opportunity to experience the bot at the UAT. Hence, we received 182 and 52 client-reported defects in phase I and phase II, respectively. The project team must recapture the code for individual change requests, resulting in time and budget overrun. Root cause analysis for phases I and II in Step 2 was conducted with the project teams, process consultants, and RPA SMEs' participation.

We were able to classify the client-captured defects at UAT as the client reported defects, change requests, and other defects. Other defects included the client's mistakes, such as wrong data feed, network failure, incorrect credentials feed to the bots, database errors, other loading failures, and technical limitations. Phase I in Step 2 resulted in 29(16%) change requests, 33 (18%) client reported defects, and 120 (66%) other defects. Though the total number of defects was reduced in phase II Step 2, still 20 (39%) change requests, 10 (19%) clients reported defects, and 22 (42%) other defects were identified. Moreover, 29 (59%) of the change requests in phases I and II in Step 2 were occurred due to the changes directly made to the UAT environment by the client maintenance team without acknowledging the RPA project team. In phases I and II in Step 2, defect identification started only at the SIT stage, and the client identified most of the defects only during the UAT. Process executers had identified a couple of business processes rarely used by the process executers.

Table 4.5 : Defects captured by the client during the UAT of all three phases.

Category		#		
		Phase I	**Phase II**	**Phase III**
Change requests		29	20	3
Client reported defects		33	10	0
Other defects	Data error	43	2	0
	Network failure	17	5	0
	Incorrect credentials	11	3	0
	Database error	7	5	0
	Other loading failures	36	5	0
	Technical limitations	6	2	0

Moreover, business process executers have forgotten to mention rarely used business processes during the requirement gathering. Further, many concerns were raised about the business analyst for missing details and delayed information, which happened in phases I and II in Step 2. Moreover, process executers and process owners had only the UAT to experience the bot functionalities, and they struggled to adhere to the bot behavior within a limited time. However, we noticed that some process executers were reluctant to provide in-depth information on the business processes due to job security concerns.

As illustrated in Table 4.5, 120 (66%) other defects in phase I were due to the unintentional work done by the client. Those are the addition of new features to the system by the client system maintenance team without informing the project team, not updating credentials on time, network failures, and erroneous data updated in the bots. To fix the change requests and defect identified by the client took additional two months and five technical officers' efforts to fix high (8) and medium (12) severity defects. In phase II, the team spent one additional month with four technical leads to

fix high (2) and medium (5) severity defects. The project got delayed and resulted in a significant financial loss to the global IT services and delivery company. The client agreed to pay only for part of the extra effort required to address the defects. Moreover, the client had concerns about on-time delivery and the accuracy and functionality of the delivered RPA bots implemented for a finance application. During phases I and II, the client does not have visibility of the project's progress. The project team's collaboration was limited to the inception and completion, which was highlighted as a weak client and vendor relationship. In both software implementation processes, the business analyst was appointed and stayed with the client to grab the existing system functionalities by engaging with the business process owners and process executers.

We understood that the existing process did not support RPA transformation and decided to formulate a software process based on the agile framework. Section 4.4 describes the steps to verify the agile suitability for RPA transformation projects. The same section verifies the selection of scrum as a base to formulate the Raban framework. Therefore, to eliminate any mal-practices and overcome the challenges in the scrum framework, we have conducted an online survey considering the company's 12 active, agile-based projects. We had two projects run on the kanban framework, and 10 projects run on the scrum framework. Thirty-four responses were collected. 83% of the respondents were in projects run on the scrum framework, and only 17% were in projects based on the kanban framework. One response was discarded out of the 34 responses due to data integrity issues. Each project had two or three software engineers allocated. However, engineers or associate engineers rarely play the tester or developer role. 76% of the respondents had experience with the agile and plan-driven process, and 88% had more than one year of agile experience. 87% of respondents agree that they use user-friendly, transparent tools to track and manage the project. Therefore, we understood that tool usage is not a challenge in agile projects. Further, no issue identified in team members competency, and they are very collaborative.

The solutions and best practices identified for the 15 burning challenges captured with agile practitioners are not challenging to adopt. What mainly requires is to enhance the commitment, communication, and customer awareness allocating more time and resources. Therefore, it is advisable to acknowledge both the Dev and management teams of possible agile practitioners' challenges and solutions. Moreover, recommended to adhere to the best practices. 55% of respondents mentioned that they have sufficient time for quality assurance work, where 36% of respondents does not have. 67% of respondents have sufficient time to write test cases. Moreover, 52% of respondents agree on spillovers in most sprints. 61% of the survey respondents have a separate automation team. 46% of the respondents mentioned that they must spend additional time updating the automation team on requirement changes and regression test suite. Table 4.3 illustrates the taxonomy we developed for the best practices followed by the agile practitioners to overcome the 15 burning challenges. As seen in Table 4.3, challenges were categorized into four groups, and solutions were listed under the 'Solutions' section. Challenges and solutions were mapped with the number mentioned at the end of each challenge. Agile practitioners can understand the solution for a challenge by referring to the number mentioned at the end of the challenge description. In Agile practices, shorter time-boxed sprints, collaboration with the client, and high transparency and visibility make Dev team life challenging compared to the plan-driven process. Therefore, acknowledging the management of the challenges faced by the Dev team and preparing for possible solutions helps reduce the stress in a Dev team. Therefore, the exact solutions were considered in formulating the novel software implementation process for RPA projects.

To understand the pain points and identify the best practices in customer satisfaction, we collected data after conducting an online survey across the industry known for design thinking practices. As per the data analysis based on the Grounded theory, we categorized the best practices to satisfy customer expectations in agile

Table 4.6 : Best practices identified to satisfy customer experiences.

#	Description
1	**Customer's real need identification** • Business analyst should not be the only contact person for requirement gathering. • Before starting the product implementation, it is required to spend a considerable amount of time (e.g., 6 weeks) to capture customer need. • Follow design thinking techniques. (e.g., scenario identification, customer journey mapping, user story mapping, user experience design, heat maps, and customer profiling. • Practice HCA in discussions and conducting workshops.
2	**Transforming customer need into pilot solutions** • To finalize the idea in customer's mind techniques like Proof of concept (POC), prototype, persona identification and wireframes can be used. • Get feedback for the work completed at sprint to make sure project is on track. • Pilot solutions help to fail fast avoiding leading to an unexpected failure. • Validate at the end of every sprint to make sure project team is solving the right solution to avoid expensive development or design occurs.
3	**Visualizing the pilot solution for customer feedback** • Understand stakeholder perception on idea visualization. • Gather feedback for the outcome of the planned solution, mockups, proof of concepts, and prototypes by asking questions. • Evaluate the idea with the end-user. • Let the customer to think on their own requirements and planned solutions.
4	**Idea generation for the pilot solution** • For idea generation divergent approach can be used. Best idea is not always the first idea. • Focused on identifying the most appropriate solution for the customer in the given business context. • Good confidence in the development team.
5	**Brainstorming** • To justify decisions, evaluate ideas, generate ideas, pros and cons to make decisions, and clarification of expertise ideas in brainstorming. • For final decision spread responsibility with everyone. • To provide right solution and to understand the right requirement, stakeholder's involvement is essential. • To clarify unclear areas in product requirements better to get involved with domain and technical experts. • Conflicts raised due to different understanding might avoid among project stakeholders.

Table 4.7 : Challenges faced by Agile teams while practicing design thinking.

Challenges Faced by Agile Teams While Practicing design thinking
• Lack of theoretical knowledge on design thinking. • Daily practice of design thinking with lack of understanding. • Lack of understanding of agile principles. • Not understanding about the common practices for both agile and design thinking (e.g., collaboration is available). • Make the life uneasy when vendor communicated only through product owner, eliminating to contact the end user.

projects into five groups. (i.e., *visualizing the pilot solution for customer feedback*, *customer's real need identification*, *brainstorming*, *transforming customer's real needs into pilot solution*s, and *idea generation for the pilot solution*.) Table 4.5 summarized the list of challenges face by the agile teams face while practicing the design thinking process. Design thinking practices can be supported by the best practices listed in Table 4.6. Proposed design thinking practices/techniques and factors to improve customer satisfaction are listed in Table 4.6. Our contribution to the research is the relationship derived between design thinking practices (i.e., thinking by doing, collaborative work style, synthesis of the diverging and converging approach, visualizing, and human-centered approach) and identified best practices based on the techniques explained in [34]. A human-centered approach can be used to identify customer expectations. Customer needs can be transformed into pilot solutions using Thinking by doing. Proof of concepts, prototypes, and wireframes are standard techniques described in both situations. The stakeholder's idea could be better understood by the feedback received for the pilot solution in Visualizing. A combination of diverging and converging approaches could be used in idea generation for pilot solutions. Best practices can be improved using several other collaborative work techniques discussed in [34]. Figure 4.7 illustrates the framework we developed. Five design thinking practices have been used to improve customer expectations. The proposed framework can be used to improve customer expectations for agile projects.

The vendor needs to develop the product to achieve customer satisfaction. Hence, the vendor should read what is on the customer's mind. The vendor can use Human-centered approach techniques to understand the customer expectations. The combination of divergent and convergent techniques (Synthesis) like pattern finding, ideation, and creating multiple alternatives can be used in collecting product development-related information. Prototype development, wireframe development, and proof of concepts are the techniques that the vendor can use to implement a product in the Thinking by Doing stage. The customer can receive feedback for the workable product developed during the Visualizing stage. The Collaborative work style techniques can be practiced within the four phases of the framework. We incorporated these learnings when we formulated the Raban framework. For example, "Customers think it is privilege for them to work with domain experts and SMEs" Unable to prioritize customer expectations" are listed as a paint points under Table 4.4. To eliminate those in the Raban framework, we introduce the RPA SME role for all the meetings clients get involved with, especially at the business process backlog prioritization stage. To incorporate the best practices under "customer real need identification," we proposed an iterative business process elicitation process with the engagement of business analysts, RPA SMEs, business SMEs, process executers, business process owners, and the project team.

Several analyses were carried out while developing the Raban framework to guide its design and evaluate its ability to reduce the cost and effort of fixing the change requests. The client reported defects identified during the UAT. In Step 4, some team members were new to the project due to team members' resignations in phases I and II in Step 2. Team members' turnover was mainly attributed to the frustration due to the heavy workload resulting from the higher number of change requests and client reported defects in the first two phases. New team members replaced critical roles like project manager, program manager, and QA lead. The same RPA tool was used in Step 3, as none of the issues was identified in phases I and II in Step 2. Next, we described the results from all three phases.

The Raban framework introduced iterative ceremonies, such as prioritized business process elicitation, prioritized use-case backlog grooming, and robotize-sprint review to strengthen the communication between the two teams. The Raban

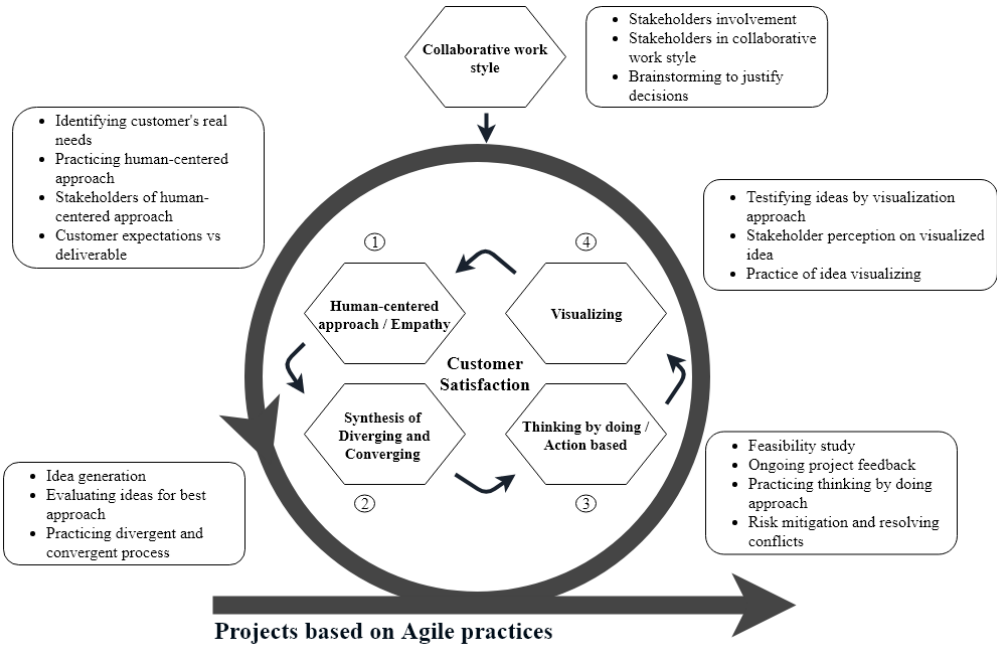

Figure 4.7 : Framework to satisfy customer expectations using design thinking practices in agile practices.

framework adheres to the four agile rules and 12 agile principles. It could harness changes for the customer's competitive advantage by letting the customer experience the bots and provide feedback early in the development cycle. Further, we used a transparent project management tool to manage the project. For example, the prioritized use-case backlog was maintained in Jira. Consequently, the robotized-sprint work progress was visible to the client. Change requests count in phases I to III in Step 2 are illustrated in Figure 4.8. In Step 4, when executing the project in the Raban framework, the project team was able to identify change requests early in the project life cycle, and the change requests count reduced as the project progressed due to the regular meetups and high visibility. In phases I and II, change requests were identified only at the UAT, which was too late and resulted in time and budget overruns.

As illustrated in Figure 4.9 for phases I and II in Step 2, defects identification started only at the SIT stage, and most of the defects were identified during the UAT. In Step 4, defect identification started at the first sprint and continued till the last sprint, and zero defects were identified at the UAT. Enhanced collaboration between the project team and client helped capture the defects early. Rarely used business flows by

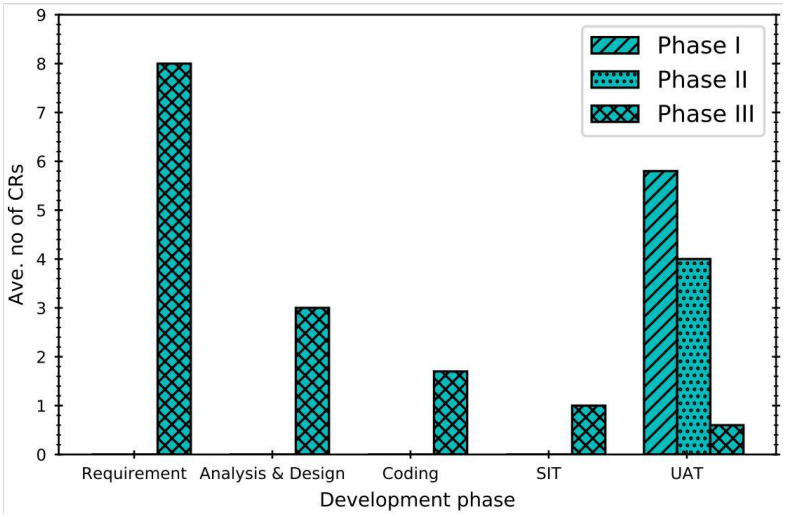

Figure 4.8 : Change requests identification status throughout the project life cycle.

the process executers were identified only during the UAT in phases I and II. Process executers had forgotten to mention those at the requirement elaboration stage. In Step 4, workshops were conducted to elicit the product backlog with the support of process executers, business SMEs, RPA SMEs, project teams, and business process analysts. Two to three workshops were conducted to cover the entire system, and it resulted in the identification of many businesses flows rarely used by the process executers. As the process executers, business SMEs, and business process owners participate in the robotize-sprint review, they have the opportunity to gain hands-on experience on the bot's behavior before deploying to the production environment. Furthermore, some process executers were reluctant to provide in-depth information on the business process as they were worried about their job security. However, by encouraging active and frequent engagement with the process executers, the Raban framework overcame such hurdles in Step 4.

Phase II changed the implementation methodology to the in-house developed test-driven development methodology. Also, the development and testing teams were more engaged with the client. Consequently, time and effort spent on internal defects identified during SIT got reduced compared to phase I. Table 4.5 illustrates the defect identification throughout the lifecycle of all three phases. In phases I and II, while the defect identification started during SIT, most defects were identified during the UAT. Whereas in Step 4, starting from the first iteration, the project team started to identify the defects, and no defects were identified during the UAT. Early defect identification leads to cost reduction by saving time and improving product quality.

Due to the Raban framework's use in phase III, we came across only three change requests during the UAT, and bots were delivered on time. Which was a significant achievement compared to phases I and II, primarily due to the prioritized use-case backlog that helped identify the defects, requirement changes, and missed requirements early in the project life cycle. In phases I and II, the client team could only raise the change requests and defects at the UAT. The complexity of the business processes selected for all three phases was the same. Where we ensured a level of

difficulty, ability to analyze, uncertainty, inter-dependence, variety, and non-routines with other business processes in all three business processes were similar [48]. However, the time spent on project completion is different in all three phases. Eight months in phase I and 12 months in phase II, whereas only three months in phase III. Significant project execution time reduction in phase III is mainly due to the iterations we have in the Raban framework. We had two teams parallelly developing the bots. Each project team member is responsible for selected use-cases completed within the sprint. A new use case would be selected if the selected use cases were completed early. Frequent collaborations with the client reduced the unwanted time spent on issues that arose in bot development. Moreover, significantly reduced time spent fixing the change requests was identified in UAT compared to phases I and II. In phases I and II, process executers had a tough time accommodating the changes to the existing templates required for frequently changing external data (e.g., currency data and table stock auctions) and generating new templates for the business processes due to the newly introduced bots.

As the project team used a big bang approach to deploy the bots to the UAT environment in phases I and II, the project team had little time to prepare the templates to support the bot's functionalities. Further, in those two phases, process executers did not have an opportunity to engage with the bots until deployed to the UAT environment. However, in phase III, with the iterative and incremental development process, executers have the opportunity to get the bot experience before they deploy to the live environment. Raban enabled the client team to start end-user training after the first sprint. Therefore, process executers had the opportunity to acknowledge and prepare the templates early in the project life cycle. Moreover, in the testing phase, the project team had the opportunity to get ready for test data in the subprocess level iteratively, rather than focusing on bulk test data during SIT in phases I and II.

Another challenge in RPA implementation is to build the client's trust in bots' functionality and accuracy. For example, as the bots were implemented for a finance application, even after phases I and II in UAT, the client had doubts about the accuracy of financial transactions, handling outliers, violation of compliance, security policies, and standards. Iterative and incremental implementation in phase III allowed the client

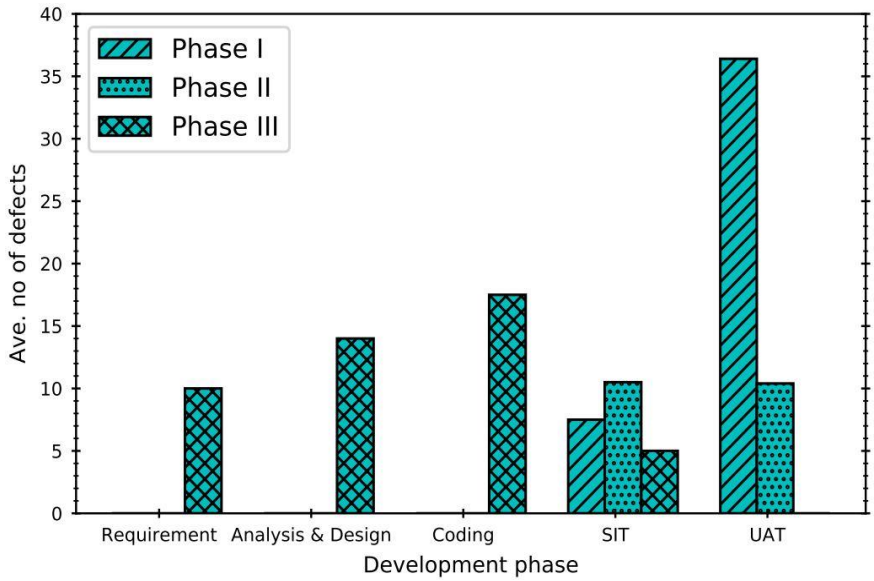

Figure 4.9 : Defect identification status throughout the project life cycle.

to engage with the project team and gain high visibility on the project progress reducing such concerns. Moreover, as all the bots had cognitive capabilities (i.e., speech recognition, natural language processing, and machine learning), additional time is required to train and fine-tune the bots by process executers. Therefore, sufficient time can be allocated to finetune machine learning models by releasing even a primitive version of a cognitive bot early in the development process. The process executers could continue refining the bots until deployed to the production environment. Therefore, the Raban framework's iterative nature resulted in early and comprehensive feedback, proactive training, and more accurate bots.

Further, in phases I and II, the client had no visibility of project progress because the client's collaboration with the project team was limited to the project's inception and completion. Hence, the project team got the client feedback for all the completed bots only during the UAT. In phase III, the client had the opportunity to engage with the project team every fortnight due to the iterative ceremonies conducted within the robotize-sprint (i.e., prioritize use case backlog grooming and robotize-sprint review). During the robotize-sprint review, the project team demonstrated the use-cases completed during the sprint. Therefore, the client could experience the bots' functionality right after they were completed and had the opportunity to raise any issues immediately. Client business value started to grow when the bots were deployed to the production environment. Therefore, in phase III, the business value starts to generate from the first sprint and continues to grow as the project progresses. In phases I and II, the client had to wait until the project team deployed the bot at UAT, which happened at the project closure.

The project failure risk was high in phases I and II because the team was initially clueless about the RPA transformation. Alternatively, the client did not experience the bots until deployed to the UAT environment. Typically, the risk of project failure is high at the project's inception, and it becomes zero only when the bots get deployed to the production and then on the live environment. In Raban, the project risk reduces much faster due to the features we used as in iterative and incremental

development. Which encourages providing high visibility to both parties via a transparent project management tool and recurrent engagement with the client to share project progress and solicit feedback. Moreover, continuous collaboration with the client changed the process executers' negative attitude towards RPA transformation. They had the opportunity to experience the bots during the robotize-sprint review demonstration and voice how the bot should be developed.

We focused on developing a set of bots for a specific client and domain and used a particular combination of tools such as Workfusion and Jira. There could be limitations in generalizing the Raban framework for other RPA projects. The project team needed more time to understand the project's velocity as the projects' sprint sizes were not stable. For example, in one project, three sprint sizes were 97, 33, and 0 story points, whereas, in another project, it was 60, 138, and 73. Hence, the project team required more time to define the project velocity. Also, we needed to identify the specific measurement criteria for the framework. Estimating the use-cases in the backlog was performed with the consultation of the RPA SME, QA Lead, and Tech lead, considering the number of automation steps, tool complexity, and logical complexity. Hence, it is required to have a generalized estimation model for estimation. Further, at the end of phase III, another set of interviews was conducted to identify what went wrong, what went well, and improvements needed in the proposed framework. Open and axial coding was used to analyze the feedback. We categorized the feedback as relating to resolving change requests and defects, applicability of the Raban framework, and suggestions to improve the framework. Feedback towards the Raban framework was mainly positive, where most participants accepted the presented framework. Further, we received two suggestions to improve the robotize-sprint review ceremony to update the ceremony phase only with the significant changes and explore the phase with the changes that require a minimal fixing time of the received feedback. A couple of new ideas to improve the use-case estimation process by developing a specific estimation mechanism to estimate the prioritized business process use-cases were also suggested. Moreover, more suggestions were there to

improve the business process elicitation process, generalizing the framework for other technologies such as Automation Anywhere, UiPath, and BluePrism. Also, can consider business domains such as; health care, customer support, and IT Help desk, and integrating the Raban framework with the DST developed to select CBP.

Based on the feedback, we refined the Raban framework to separate estimation and use case backlog prioritizing sessions, where both were initially conducted under use-case backlog grooming sessions. Moreover, we conducted two workshops with other RPA project teams and the RPA SME to get feedback on the Raban framework and further improved the framework based on the feedback. We introduced two options: update the robotize-sprint use-case backlog or prioritized the use-case backlog for robotize-sprint review based on the feedback. Whereas in the initial version, we updated only the prioritized use-case backlog irrespective of the criticality of the feedback. Based on the feedback, we further understood that updating the prioritized use case backlog is not required for minor feedback received during the robotize-sprint review. Update only the robotize-sprint use case backlog would be sufficient. Further, we only had the process owner represent the client at the initial version of the Raban framework for the robotize-sprint review. After incorporating feedback from RPA project teams and SMEs, we fine-tuned the Raban framework to the final version presented in this paper. Moreover, we compared Raban with scrum, kanban, and extreme programming to highlight the differences in its roles and responsibilities, due dates and delivery timeline, delegation and prioritization, Modifications or changes, measurements of productivity, and best applications. Results were updated in Table 4.8.

Some of the project team members, including the project manager, program manager, and QA lead, were new to the team due to the team member turnover at the closure of phases I and II. We believe that the project team's changes did not contribute significantly to phase III's success compared to the other two phases, as all team members were new to the agile practice. At the inception of phase III, we conducted agile workshops to ensure everyone understood agile principles, values, and scrum

practices. Moreover, experienced RPA professionals' resignations in phases I and II, training was conducted for new joiners on domain knowledge and RPA technology. Even for client engagement, we must introduce new team members. Hence, we believe that the experience gained in phases I and II and familiarity with the client and its business with time had little attribution to phase III's success. Therefore, we believe the proposed framework does enhance the process of developing RPA bots and could be applied to other RPA projects, teams, and organizations.

Table 4.8 : Comparison of Raban with Scum, Kanban, and Extreame (XP) Programming.

Description	Raban	Scrum	Kanban	XP Programming
Roles and Responsibilities	1. Has pre-defined roles and responsibilities for a team. a. Robotize master facilitates the team. b. Business process owner owns only priorotized business process backlog. c. RPA SME to facilitate client on business process prioritization, estimation, and all other activities. d. Business SME to facilitate on business process prioritization, business process elicitation, and virtual bot validation. e. Process executers to facilitate on business process elicitation and virtual bot validation. f. Business process analyst to facilitate on Business process elicitation. g. Robotize team focuses on virual bot implementation.	1. Has pre-defined roles and responsibilities for a project. a. Scrum master facilitates the team. b. Product owner owns the product backlog and defines project goals and objectives. c. Scrum team focuses on product development [17].	1. No pre-defined roles for a team. [69]	1. Has coach, programmer, tester, and a consultant roles. [71]

	Raban	Scrum	Kanban	XP Programming
Due dates/ Delivery timeline	1. Once the candidate business process selected for the RPA transformation, delivery timeline decides by the Robotize team along with client. 2. Robotize team decides on the robotize sprint backlog.	1. Release plan and sprint plan decides on delivery timeline. 2. Sprint backlog decides on deliverables. 3. Product owner and scrum team defines on sprint backlog deliverables [17].	1. Output decides based on the need. Delivery happens continuously [69].	1. Small and short releases. [71]
Delegation and Prioritization	1. Business process backlog should prioritzed based on the business process owner, RPA SME, and Business process SME preference. 2. Usecase backlog prioritized based on the Business SME and RPA SME preference.	1. User stories were prioritized based on the customer preference [17].	1. Team members are allowed take a new task at the completion of existing task [70].	1. Complete the unit test before coding done. 2. Pair programming in development. 3. Integrate and test the developments. [71]
Modifications/ Changes	1. Major changes to the candidate business process are not allowed during the virtual bot implementation.	1. Recommended not to do any changes to the stories in sprint backlog [17].	1. Changes can be made at the project mid-stream. Prior to project completion continuous improvements allowed [70]	1. embrace change early in baby steps. [71]

	Raban	Scrum	Kanban	XP Programming
Measurements of Productivity	1. Estimation done based on the RPA SME, Tech lead, and QA lead of the project assessments.	1. Sprint success relies on the once before it. Productivity measures based on the velocity. Each sprint laid back-to-back [17].	1. “Cycle time” measures the productivity. Which measure the time taken to complete one full cycle, from start to end.	1. Programmers do the estimation c onsidering the effort needs to implement the stories. [71]
Best Applications	1. Best for projects with stable code base with minimal changes to the existing code.	1. Recommend for stable teams with priorities, which may not change everytime [17].	1. Recommended for projects with different priorities.	1. Best for small and medium scale projects. [71]

4.9 Summary

We could not identify an existing software implementation process to execute RPA projects in literature, industry, and global IT services and delivery company. At first, we decided to execute RPA projects in a plan-driven process. As the RPA project team experienced many change requests and defects identified during UAT, we used the in-house developed model to execute RPA projects. However, no significant reduction in change requests or defects was identified in UAT. Hence, we decided to formulate a process for RPA projects. Then we conducted a root cause analysis for the client-reported defects captured at the UAT in both plan-driven and in-house developed model processes. The results revealed that most change requests were due to the client maintenance team's changes made directly to the UAT environment without informing the project team. The same was identified as a communication gap between the client maintenance team and the project team. Rarely used process functionalities were not captured during the requirement gathering, and some features were not captured in the end-to-end functionality of the business process. Which again highlighted the weak relationship between client and vendor. Initially, we tested the agile suitability for RPA projects with the agile suitability tool used by the company and another tool derived from the literature. Both resulted in the agile process being suitable to be used in RPA transformation. Out of existing agile practices, only scrum aligns best with our requirements as it absorbs frequent requirement changes, reduces the overhead of the process, and maximizes the time used for valuable work to be done.

Moreover, we could not use scrum the way it is for RPA transformation due to RPA characteristics. Therefore, we studied the current issues that exist in scrum projects and categorized the identified challenges and malpractices into six groups. After that, each category was researched for better solutions to avoid those issues in deriving the RPA transformation process. We identified 15 burning challenges and taxonomy to identify the solution for those 15 challenges. Further, as a solution for *client management issues*, our study helps us identify the pain points and best practices to understand the customer expectations. Moreover, a list of design thinking best

practices and techniques to improve customer satisfaction were identified. Further, we have derived a framework to be used to satisfy the customer expectations using design thinking. We used those findings in formulating the software implementation process template for RPA projects.

We presented a novel, iterative, and incremental implementation process for RPA projects to reduce the change requests, and the client reported defects captured at the UAT. The agile approach to developing the Raban framework was motivated by the need to accommodate frequent changes, iterative and incremental development, high visibility, early identification, and resolve issues. We validated the Raban framework by applying it to an RPA transformation project for global IT services and delivery company. The resulting project was delivered with only three change requests and no defects within the allocated timeframe. We identified a significant reduction in change requests (85%) and defects (100%) compared to phase II based on test-driven development models. Moreover, the framework encourages a high level of transparency across the project and delivers partially completed bots early and incrementally to solicit client feedback. Furthermore, this also enables the client to proactively optimize the bot's cognitive capabilities to enhance their business processes and user training. We generalized the Raban framework after incorporating the review comments captured at the two workshops conducted with other RPA SMEs and project teams.

We developed and tested the framework with a project team in an organization. However, to validate the novel framework, we need to test the framework in different software development teams. Due to the unavailability of such software development teams in other organizations, to reduce the impact, we conducted workshops within the company and RPA SMEs in the industry where we elaborated the Raban framework and its execution in detail. We used business processes with the same complexity for all three phases (i.e., phases I, II, and III). Though the project team in phases I and II are the same, we had new team members due to high employee

turnover in phase III. We use WorkFusion as an RPA transformation tool in phases I, II, and III. Moreover, in all three phases client improved its maturity in the RPA transformation procedure and virtual bot behavior. Therefore, the defects identified due to lack of domain knowledge continuously decreased from phase I to phase III. Though we engaged with three different business process executers in three phases, their fear of job security was reduced and improved the support in requirement gathering in phases II and III compared to phase I. However, it has minimal impact on our findings as our focus was to reduce the number of clients who reported defects and CRs. In all three phases, the results were evaluated by RPA and Business SMEs, RPA process heads, and PCs. Hence, we can guarantee the research conclusions. Because we set up our research to find the solution for the company burning issue, we did not have any differences between real-world settings and experimental settings.

5 DECISION SUPPORT TOOL TO SELECT CANDIDATE BUSINESS PROCESS (CBP)

RPA mimics user actions on user interfaces and tools. Software robots are not only just screen scraping technology; they may also include cognitive and artificial intelligence capabilities. To the success of RPA transformation is crucial to the selection of appropriate CBP. Because RPA is a new technology, very few process analysts have the necessary RPA expertise. This chapter creates a DST to help process analysts choose CBP for RPA.

Section 5.1 describes the steps carried out to understand the factors, Section 5.2 describes the prediction model developed, and Section 5.3 describes the results and analysis of the research conducted.

5.1 Factor identification

First, we conducted the literature review. Which helps to identify the factors impact on business process suitability for RPA transformation. Limited literature available on selection of CBP. Several articles and whitepapers discussed the challenges of CBP. We identified 25 factors that influence the CBP selection. We then 25 factors classified as financial, external, and process impact factors.

We attempted to identify any missing factors. Then, 16 factors listed in Table 5.1 was finalized as significant for selection of CBP. Five factors were combined with others when identified that all are same set of factors with different names. We get rid of four factors, because they cannot quantify (i.e., client's expected outcome of the process delivery). A new factor was identified (i.e., The requirement of irregular labor) from the interview. All 16 factors were considered in developing the online survey due to unstraightforward feedback of RPA SMEs'. Hence, we decided to conduct an industry survey to track 16 factors in RPA projects/bots developed by surveyed SMEs and their outcomes. As a result, questionnaire developed considering 16 factors. The

questionnaire was shared for a pilot survey with ten RPA SMEs from the company. RPA SMSs played the role of developers, tester, Project managers, and architects were among the SMEs profiled. We still asked survey respondents to rank the factors in the questionnaire. However, after analyzing the pilot survey results, we discovered that the factors to prioritize depended on the survey participants' role in the RPA project. As a result, we removed the factor prioritization option and revised the questionnaire in response to the feedback. An online questionnaire was distributed to industry experts all over the world. SME identification was aided by social media. Three of the 56 respondents stated that they were unable to assess the process's complexity. As a result, those were discarded. Due to lack of professionals with at least have experience completing one RPA project, we had a hard time on collecting required data. Some respondents did not respond due to their personal commitments with their clients. 22 survey participants were involved in a successful RPA projects. Eight participants had experience in failed RPA projects. The remaining 23 respondents were working on different stage of RPA projects.

We used Spearman correlation for 30 (22 + 8) responses, because dataset collected through online survey consist of both ordinal and categorical data. Service Level Agreement (SLA), No of Systems to Access (NOSA), and Workload Variance (WV) were discovered to have a positive correlation with the business process (represented as the CBP Status (CBPS)) in Table 5.2. Simultaneously, the Cognitive Features (CF), Regulatory Compliance (RC), and Volume of Transaction (VOT) are negative correlation with CBPS. Feedback from the SMEs highlighted an additional 10 factors, which are useful in identifying business process characteristics. However, due to the small dataset we use all 16 factors used to develop the prediction model. We used a two-class decision forest classification model to arrive at the RPA suitability decision. We used four-fold cross-validation to measure overall accuracy. The prediction model has an overall accuracy of 90%. Section 6.3 discusses the data analysis, two-class decision forest classification model, and spearman correlation.

Table 5.1 : Factors impact on candidate business process selection in RPA.

Factor Type	Factor Name	Description	Measurement Criteria
Process factor	Volume of transactions (VOT)	No of processing requests that the computing system will receive and respond to within a specified period	High, Medium, or Low
	Business process complexity (BPC)	Levels of ability to analyze, variety, non-routines, difficulty, uncertainty, and inter-dependence with other business processes	High, Medium, or Low
	Rate of change (ROC)	No of changes made to the business process within a month	≤ 2, 3 - 5, 6 - 8, ≥8
	Rule-based business process (RBBP)	Logic-based business process	Yes, with few exceptions, Yes, or No
	Workload variation (WV)	Irregularity of workload where additional staff support is required during peak times	Yes, Sometimes, No
	Regulatory compliance (RC)	Need to satisfy organizational, government, or any other regulations, e.g., ISO or PCI-DSS	Yes, No
External factor	Cognitive features (CF)	Task requires creativity, subjective judgment, or complex interpretation skills (e.g., natural language processing and speech recognition)	Yes, No
	No of systems to access (NOSA)	Automation task involves accessing multiple systems	≤ 2, 3 - 5, 6 - 8, ≥ 8
	Multi-modal inputs (MMI)	Images, audio clips, or videos are given as inputs	Yes, No
	Data-driven (DD)	Decisions during an activity are made based on the analysis of data	Yes, No
	Stability of environment (SOE)	Availability of UAT or any other near-production environment of the target application for automation	Yes, No

Factor Type	Factor Name	Description	Measurement Criteria
Financial impact factor	Operational cost (OC)	Cost of day-to-day maintenance and administration	High, Medium, or Low
	Impact of failure (IOF)	Financial impact to client if the bot goes wrong	High, Medium, or Low
	Human error (HE)	Task is error-prone due to human worker negligence than inability to code as a computer program	High, Medium, or Low
	End of life (EOL)	How soon the bot will become obsolete, e.g., new system is on the road map of client	≤2 years, ≤4 years, ≥ 5 years
	Service level agreement (SLA)	Time taken to complete end-to-end functionalities of a task	Seconds, minutes, hours, days

5.2 Predictive model

We used SPSS for data pre-processing and factor selection. Next, we used Microsoft Azure machine learning studio. We created a two-class classification model to develop the predictive model using the decision forest algorithm. Figure 5.1 illustrates the steps carried out in developing the predictive model. We use SPSS for data pre-processing, and based on Spearman's correlation, we selected the factors for the model. We developed the model after using Microsoft Azure after carrying out the classification and evaluation.

5.2.1 Data pre-process step

We collected 56 responses from the online survey shared across the industry using the snowball sampling method, developed based on the finalized 16 factors shown in Table 5.2. We discarded three responses during the data cleaning task to correct the data inconsistency.

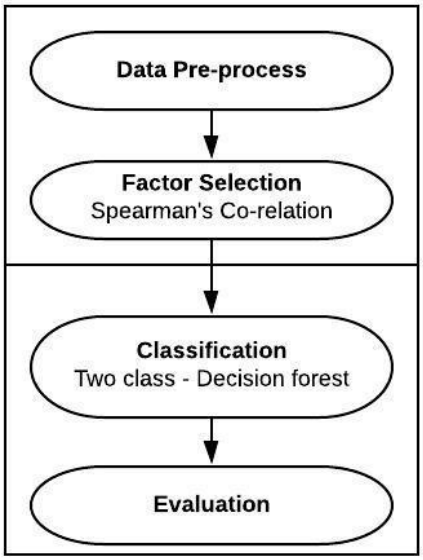

Figure 5.1: A Proposed predictive model.

5.2.2 Factor selection step

The model used spearman's co-relation to identify the significant factors in the dataset to understand the impact on CBP selection. Table 5.2 lists the factors and Spearman's correlation among the 16 factors.

Pearson's correlation is used when the data set is interval or ratio data. Spearman's correlation was used to define the strength of the monotonic relationship between variables because we have a categorical and ordinal data set. As illustrated in Table 5.2, "CBPS" is the dependent variable, and the rest of the variables are independent variables. The coefficient correlation of variables showcases that there is no interrelationship. The Volume of Transaction (VOT), Business Process Complexity (BPC), Rule-Based Business Process (RBBP), Regulatory Compliance (RC), Cognitive Features (CF), Multi-Model Input (MMI), Data-Driven (DD), Impact of Failure (IOF), End Of Life (EOL) variables have a negative correlation with the dependent variable,

Table 5.2: The CBPS and Spearman's correlation among 16 factors.

	CBPS	**VOT**	**BPC**	**ROC**	**RBBP**	**WLV**	**RC**	**CF**	**NOSA**	**MMI**	**DD**	**OC**	**IOF**	**HE**	**EOL**	**SOE**	**SLA**
CBPS	1.00	-0.62	-0.16	0.000	-0.01	0.41	-0.32	-0.37	-0.31	-0.19	-0.06	-0.18	-0.02	-0.15	-0.02	0.05	0.40
VOT	-0.62	1.00	0.16	0.006	0.14	-0.53	0.06	0.05	-0.19	0.00	0.00	0.05	0.10	0.19	0.14	0.00	-0.29
BPC	-0.16	0.16	1.00	0.19	0.16	-0.07	-0.25	-0.04	0.13	-0.23	-0.53	0.35	0.32	0.12	-0.37	0.16	0.14
ROC	0.00	0.06	0.19	1.000	-0.19	0.22	-0.27	0.13	0.03	-0.27	-0.34	0.09	0.30	0.00	0.25	0.24	0.25
RBBP	-0.01	0.14	0.16	-0.19	1.00	-0.08	-0.12	-0.15	0.06	0.15	-0.07	0.30	0.19	0.00	0.01	0.47	-0.11
WV	0.41	-0.53	-0.71	0.22	-0.77	1.00	-0.23	-0.30	0.11	0.07	-0.14	-0.29	0.22	0.00	-0.07	-0.16	0.38
RC	-0.32	0.06	-0.25	-0.27	-0.12	-0.23	1.00	0.24	-0.15	0.11	0.33	-0.43	-0.19	-0.16	0.25	-0.25	0.05
CF	-0.39	0.05	-0.04	-0.13	-0.15	-0.30	0.24	1.00	-0.07	0.61	0.21	-0.29	-0.17	0.00	-0.19	0.11	-0.42
NOSA	-0.31	-0.19	0.13	0.003	0.06	0.11	-0.15	-0.07	1.00	-0.22	-0.35	0.24	0.07	0.26	-0.20	0.12	-0.18
MMI	-0.19	0.00	-0.23	-0.27	0.15	0.07	0.11	0.61	-0.22	1.00	0.28	-0.19	0.02	0.00	-0.19	0.11	-0.24
DD	-0.06	0.00	-0.53	-0.34	-0.07	-0.14	0.33	0.21	-0.35	0.28	1.00	-0.38	-0.52	0.00	0.17	-0.15	-0.08
OC	-0.18	0.05	0.35	0.09	0.30	-0.29	-.43	-0.29	0.24	-0.19	-0.39	1.00	0.51	0.21	0.01	0.14	-0.10
IOF	-0.02	0.10	0.32	0.30	0.19	0.22	-0.19	-0.17	0.07	0.02	-0.52	0.51	1.00	0.00	-0.02	0.01	-0.04

	CBPS	VOT	BPC	ROC	RBBP	WLV	RC	CF	NOSA	MMI	DD	OC	IOF	HE	EOL	SOE	SLA
HE	-0.14	0.19	0.12	0.000	0.00	0.00	-0.16	0.00	0.26	0.00	0.00	0.21	0.00	1.00	0.37	-0.19	-0.02
EOL	-0.02	0.14	-0.37	0.25	0.01	-0.07	0.25	-0.19	-0.20	-0.19	0.17	0.00	-0.02	0.37	1.00	0.02	-0.02
SOE	0.05	0.00	0.16	0.24	0.47	-0.16	-0.25	0.11	0.12	0.11	-0.15	0.14	0.01	-0.19	0.02	1.00	-0.24
SLA	0.40	-0.29	0.14	0.25	-0.11	0.38	0.05	-0.42	-0.18	-0.24	-0.08	-0.09	-0.04	-0.02	-0.03	-0.24	1.00

CBPS' Rate of Change (ROC), WorkLoad Variance (WLV), No of Systems to Access (NOSA), Operational Cost (OC), Human Error (HE), Stability Of Environment (SOE), Service Level Agreement (SLA) variables have a positive correlation with the dependent variable, CBPS. VOT (-0.625), WLV (0.413), and SLA (0.400) have a moderate correlation with the CBPS, and those are statistically significant, were 2-tailed values respectively .000, 0.002, and 0.003. Further, RC (-0.317), CF (-0.396), NOSA (0.307) has a weak correlation with the CBPS, where the variables are statistically significant, and the 2-tailed values are respectively 0.021, 0.003, 0.025. Following variables have a very week correlation with the dependent variable CBPS, BPC (-.159), ROC (0.000), RBBP (-0.007), MMI (-0.191), DD (-0.062), OC (0.175), IOF (-0.018), HE (0.147), EOL (-0.018), SOE (0.051) and statistically not significant. (2-tailed values respectively 0.256, 0.998, 0.961, 0.170, 0.660, 0.210, 0.898, 0.292, 0.898, 0.716). However, face-to-face interviews with RPA SMEs have agreed to add the same variables to build the model, though 10 variables have a weak correlation with the dependent variable. Because we had only 53 sample sizes, RPA SMEs' recommendation was to train the model with more data and further study factors not listed within the 16. Hence, as per the RPA SMEs recommendation, we developed the model, illustrated in Figure 5.2, with overall model development and testing workflow. A two-class classification model out of multiclass classification was selected to keep the model simple. Moreover, we used the decision forests algorithm [51] because it is an ensemble learning method used for classification tasks. These models provide better coverage and accuracy compared to single decision trees. We used the decision forests framework in Microsoft Azure Machine Learning Studio, which extends several forest-based algorithms and unifies classification, regression, density estimation, manifold learning, semi-supervised learning, and active learning under the same decision forest framework. Further, the model was developed using the response received for the 16 factors identified. The prediction model was published on the Microsoft Azure platform as a web service where the SME can access it to validate the model. Further studies will be carried out for a bigger sample size adding missing

factors if any are available. RPA technology is still not correctly stabilized in the industry. Due to that, the researcher needs more time for a bigger sample size.

5.2.3 Classification step

We selected a two-class classification model out of multiclass classification to keep the model simple because this is the first phase of the model development. We plan to improve the model for multiclass classification in the second phase of the development to improve the developed model. The decision forests algorithm is used

as an ensemble learning method for better coverage and accuracy in the classification tasks.

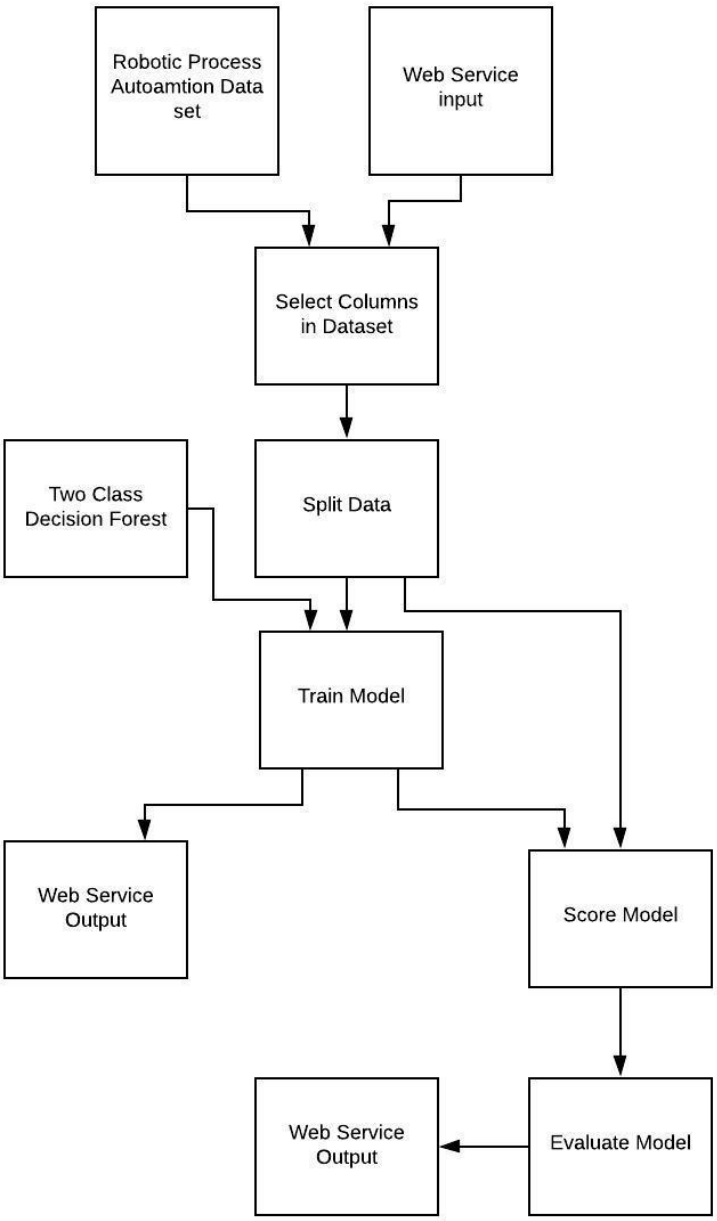

Figure 5.2: Model development and testing workflow.

5.2.4 Evaluation step

Evaluation in Azure machine learning studio is concluded with the help of evaluating the model module. It measures the accuracy of the trained model. Accuracy, F-score, precision, and recall are a couple of metrics out of many metrics available for classification models. Table 5.3 showcases the metrics values for the developed classification model. The proposed model achieves 90% overall accuracy. To validate the overall performance accuracy of the predicted model, four-fold cross-validation test was used. 70% were randomly used in the training data set, and 30% were used for testing.

Table 5.3: Predictive model accuracy.

Accuracy	Precision	Recall	F Score
0.900	1.00	0.870	0.930

5.2.5 Model evaluation

Following that, we used decision forests algorithm to assess the accuracy of the trained two-class classification model. Accuracy measures the classification model quality as the proportion of actual results to all model cases. To assess the accuracy of a classification model, common metrics such as recall, accuracy, F-score, and precision were used. Precision denotes the proportion of actual results that are positive overall. The recall is the percentage of correct answers returned by the model. The F-Score is calculated as the weighted average of precision. Recall values ranging from 0 to 1 (higher the better). Table 5.3 displays the results of four-fold cross-validation. The model's overall accuracy is 90%, and the model has good precision and recalls too. A higher F1 score further indicates good accuracy of the test.

The proposed DST consist of trained two-class classification model and the values of 16 factors. SMEs to access the predicted model to validate, we published it

as a web service using Microsoft Azure platform. To verify the proposed DST, we apply it of three business processes in a financial services company.

5.3 Results and analysis

Five RPA SMEs validated the proposed DST by applying it to three different projects in global IT services and delivery company. Figure 5.3 illustrates the output of the DST for a failed RPA transformation project. The results shown as 0 for No and 1 for Yes, along with the status of each factor. The output is explained in the second sentence. The other two projects were predicted by the DST to be suitable for RPA transformation, and they were indeed successful in actual operation at the client site.

According to our findings from the interviews and data analysis, RPA projects fail when the selected business processes exhibit the following characteristics:

1. The business process is extremely complex.
2. Workload has a high degree of variation.
3. Compliance with regulations is required.
4. The business process must have access to at least three to five different systems.
5. multi-model inputs must be supported.
6. The operating costs are high.
7. The system environment is insecure.

Analysis proved that having two or more characteristics out of seven characteristics mentioned above will lead to RPA project failure.

```
    <- 'DSS in RPA 3 [Predictive Exp.]' test returned
    ["0",    "Medium",    "High",    "X<=2",    "With    few    exceptions",
"No","Yes","Yes","X<=2",        "No","Yes","        Medium","        High","
Medium","2<X<5=5","Yes","Days", "-]
     Results
    {"Results" :
      {"Output"
            {"type":"table":"value":
                  {"Column Names":
                              {"What is the state of candidate business
                        process that you have selected?", "Volume of
                        Transactions", "Business Process Complexity",
                        "Rate of Change", "Rules-Based Business Process",
                        "Workload Variation", "Regulatory Compliance",
                        "Cognitive Features", "No of Systems to Access",
                        "Multi-Model Inputs", "Data Driven", "Operational
                        Cost", "Impact of Failure", "Human Error", "End
                        of Life" , "Stability of Environment" , "Service
                        Level Agreement" , " Scored Label Mean" , " Scored
                        Label Standard Deviation" },
                  " ColumnType" :
                              {"String", "String", "String", "String",
                        "String", "String", "String", "String", "String",
                        "String", "String", "String", "String", "String",
                        "String",   "String",   "String",   "Double"   ,
                        "Double"} ,
                  "Value":
                              {
                                  {"0" , "Medium" , :"High" , "X<=2" ,
                              "With few exemptions" , "No" , "Yes" , "Yes"
                              , "X<=2" , "No" , "Yes" , "Medium" , "High"
                              , "Medium" , "2<X<=5" , "Yes" , "Days" , "-
                              0.127917257182435" , "0.676987962011746"}
                              }
                        }
                  }
            }
      }
    }
```


Figure 5.3: prediction model results.

Demographic analysis of participants helps to understand the RPA industry and its influence on research outcome and conclusion. As shown in Figure 5.4, survey respondents held various roles in RPA project, including business analysts (30%), technical leads (3%), transformation heads (3%), project managers (21%), testers (14%), developers (26%), and architects (3 %). Analysis showcased that no agile-specific role like scrum master was not within RPA projects. Which proves RPA teams do not use agile process for project execution.

As seen in Figure 5.5, 96% and 4% of survey respondents had less than five years of experience, and over 5-years' experience in RPA projects, respectively. Which prove that RPA is new to the industry. As illustrated in Figure 5.4, 3% of respondents are business analysts, whereas 26% are developers, 21% are project managers, and14% are testers. Only 3% of respondents are heads of transformation, technical leads, and architects. Furthermore, we collected the respondents' experiences to understand the respondents' maturity over RPA technology. It was identified that only 4% of respondents have 5+ years of experience in the RPA industry because RPA technology is a new technology. 53% and 32% of respondents have 3+ and 1+ years of experience, respectively. Only 11% of respondents have less than one year of experience in the RPA industry, as illustrated in Figure 5.5.

Respondents are allowed to select three stages in bot implementation: already completed and in action, in the middle of development, and at the initiation process. We found that 42% of the respondents were satisfied with the survey results' RPA transformation. Only one to three of the seven characteristics listed in section 6.4 were present in these successful projects. Only 15% of the bots were unable to determine whether the client/business was dissatisfied with the resulting bot. 32% of respondents are still in the middle of bot implementation, and 11% of respondents are at the RPA transformation planning stage illustrated in Figure 5.6.

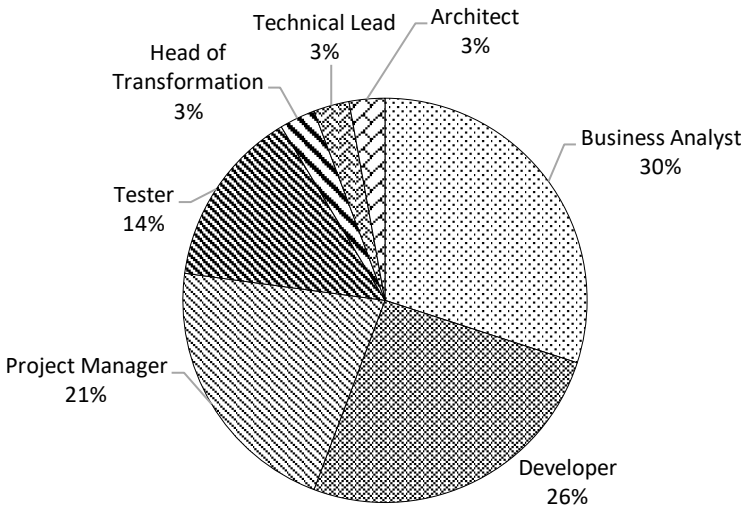

Figure 5.4: Different roles played by survey respondents in RPA projects.

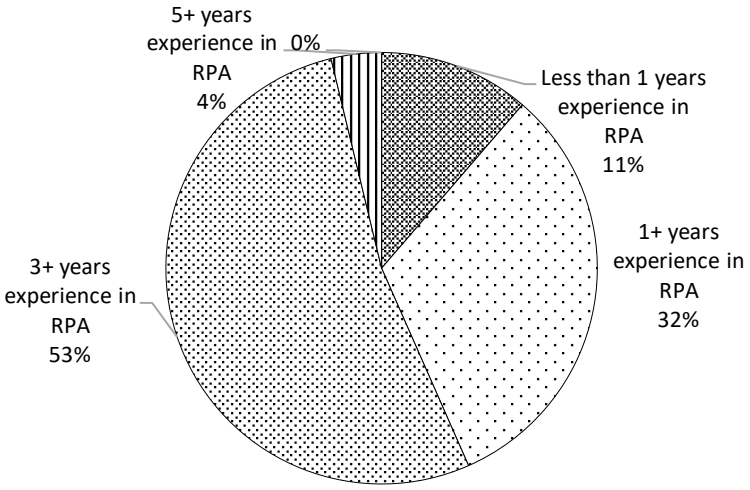

Figure 5.5: Respondents' experience in RPA projects.

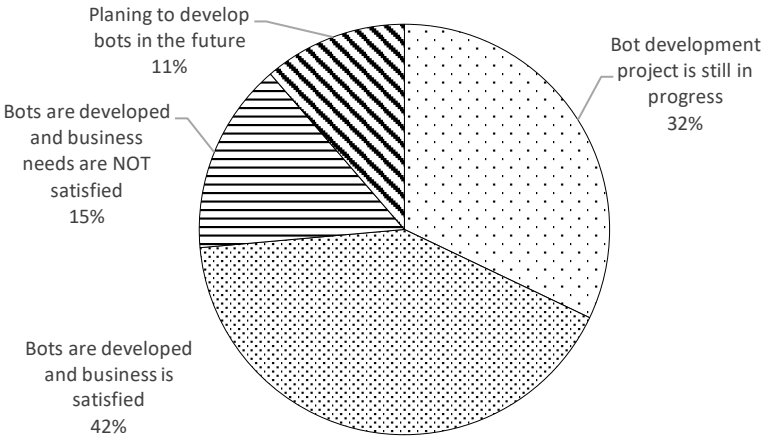

Figure 5.6: Survey respondents' bot development status.

5.4 Summary

To select CBP for RPA projects, we created a prediction model. This model is built around 16 factors that define a business process. A combination of a literature review and interviews with RPA SMEs was used to identify the factors. Then, we used an online survey to determine the impact of those factors on the success of an RPA project. Workload variance, number of systems to access, and service level agreement all correlated positively with the RPA outcome of the business process, whereas volume of transactions, regulatory compliance, and cognitive features all correlated negatively. Even though the other ten factors have an insignificant correlation, we proceeded with all 16 factors to develop the model because RPA SMEs thought these factors were still valuable, and our dataset was small. The outcome of a CBP was then predicted using the two-class decision forest classification model. Model's overall accuracy was validated in four-fold cross-validation process. The trained model showed 90% accuracy. Five RPA SMEs validated the resulting DST by successfully applying it to three completed RPA projects.

Now we have formulated the Raban framework and improved the decision on identifying the business process for RPA transformation. In the next step, we focused on identifying quantitative measurements for the Raban framework to measure the RPA projects quantitatively.

6 SOFTWARE METRICS FOR RABAN FRAMEWORK

Weekly metric analysis guides the project team to achieve successful project completion because of frequent metric analysis within a short time alert even a slight deviation of the predicted product quality. However, the project team should know which metrics to measure for successful predictions. Though many metrics are tracked, analyzed, and researched, limited resources discussed metrics to be tracked and managed in scrum projects to improve product quality, team productivity, and project predictability. Therefore, to derive metrics for RPA projects, we first identify metrics for scrum projects and based on that, we derived metrics for RPA projects. This chapter describes the steps we followed to identify metrics in the scrum and to derive metrics for the Raban framework.

Section 6.1 and 6.2 describe the steps followed to identify quantitative measurements for scrum projects and the Raban framework, respectively. Data analysis and results are described in Section 6.3.

6.1 Metrics for scrum projects

Out of the 80 challenges captured during the focus group discussion conducted at the global IT service and delivery company, we identified 36, seven, and eight challenges in *lack of adherence to agile practices*, *scope change in requirement freeze*, and *lack of quantitative measurement* categories, respectively.

We conducted face-to-face interviews with nine scrum masters out of 12 projects based on scrum frameworks within the company. As illustrated in Table 6.1, there were nine agile development, two agile maintenance, and an agile testing project. All are ongoing projects in different sprints. Projects defined sprint duration as mentioned under the ‘number of weeks per sprint’ column in Table 6.1. However, we focused only on the development projects in agile because RPA projects are

Table 6.1 : Description of Scrum Projects.

Project Name	**No of Sprints conducted**	**No of weeks per sprint**	**Project Type**
Project 1	5	2	Agile development project
Project 2	8	2	Agile development project
Project 3	4	2	Agile development project
Project 4	10	2	Agile development project
Project 5	12	2	Agile development project
Project 6	8	3	Agile development project
Project 7	11	2	Agile development project
Project 8	12	2	Agile development project
Project 9	15	2	Agile development project
Project 10	25	4	Agile maintenance project
Project 11	19	3	Agile maintenance project
Project 12	5	1	Agile testing projects

development projects. Interviews with the scrum masters re-assured the unsuitable metrics usage, improper metrics analysis, lack of awareness of metrics, and negative attitude on metric tracking in scrum projects. However, scrum teams are supposed to track and conduct weekly metric analyses. During face-to-face interviews with the scrum masters, 32 metrics were captured and listed in Table 6.2. Moreover, detailed descriptions of the metrics are listed in Appendix IV. The literature review identified 49 metrics and is listed in Table 6.3. After combining both findings, we conducted the data analysis using the constant comparative method in Straussian grounded theory. The constant comparative method identifies themes and sub-themes among the qualitative data [41]. To identify the themes, we first reviewed the notes taken during the focus group discussion, interviews with nine scrum masters, literature review, and interviews conducted with SMEs. Once all the data were coded, we again read through the themes to identify sub-themes through axial coding because each quote identified can be coded under multiple categories. When developing themes and sub-themes

Table 6.2 : Metrics captured in face-to-face interviews.

Metrics captured in face-to-face interviews	
1. Definition of readiness/ Requirement clarity index	17. Cost of poor quality
2. Unit test coverage for the developed code	18. Cost of quality
3. Erainsight	19. Productivity
4. Delivered defect density	20. Open defect aging
5. Defect density (sprint Defect Density, Project Defect Density)	21. Defect burn down
6. UAT DD Benchmark metric	22. Test execution productivity/ Test execution progress/ Test execution rate
7. Open defect severity index	23. Test design productivity
8. Defect rate	24. Effort Variance
9. Defect slippage rate	25. Defect removal efficiency
10. Defect severity index	26. Time to find a defect
11. Sprint velocity	27. Defect to remark ratio
12. Commitment ratio	28. Customer delighted index
13. Test case burndown	29. Ontime delivery index
14. Test coverage	30. Delivery ratio
15. Spillover per sprint	31. Delivery maturity index
16. Sprint-burndown chart	32. Release burndown

initially, we removed irrelevant quotes. After that, we removed duplication in identifying quotes. Correspondingly, we accurately understand the quotes through brainstorming sessions with SMEs. Finally, we categorized the quotes into the most appropriate theme or sub-theme. We finalized the sub-themes for the agreed core categories during the selective coding. We conducted brainstorming sessions with three process consultants, who are Certified Scrum Masters (CSM) and two scrum masters in the company. Table 6.5 illustrates a detailed description of the SMEs who validated the research findings. Some SMEs do not want to expose their names, and we use numbers to represent them. ‘SME Role’ column explained their work carried out in the company and experience clearly described under the ‘SME Experience’ column. Once we have finalized the metrics for Scrum projects, we use those metrics to derive the metrics for the Raban framework, as Raban is novel and derived based on Scrum framework.

6.2 Metrics for Raban framework

To identify the RPA projects' metrics, we first collected what they wanted to measure from the team. For this, we conducted two brainstorming sessions with the project team and RPA SMEs in the company. The best-fit metrics in scrum projects we identified were beneficial for the metric identification for the Raban framework. We conducted focus group discussions with the project team, RPA SMEs, and process consultants of the company and finalized metrics for RPA projects run on the Raban framework. We tested the metrics for two RPA projects. Because this is the first stage in metric identification, the RPA team needs more time to finalize the best-fit metrics for RPA projects to improve product quality, team productivity, and project predictability.

6.3 Results and analysis

We analyzed the data collected through face-to-face interviews with nine scrum masters. We studied how scrum teams follow agile principles and values

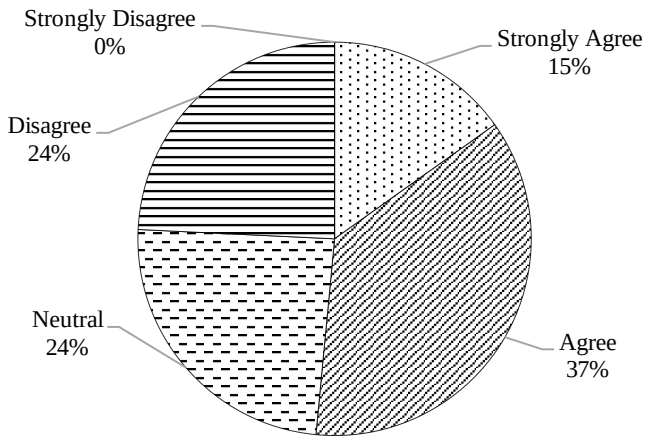

Figure 6.1: Probability of conducting backlog grooming ceremony (Every week).

through demographic analysis. 80% of the respondents conducted daily standup for 15 minutes as instructed in scrum principles. As in the scrum guide sprint planning session conducted before start the sprint by 79% respondents. In Scrum, some respondents do not conduct backlog grooming, and sprint reviews. Figure 6.1 illustrates the likelihood of conducting backlog grooming by 52% of the respondents. 18%, 24%, and 21% of the development teams conducted sprint reviews, backlog grooming, and sprint retrospectives, ceremonies, respectively. Consequently, we understood that scrum teams were unaware of how vital scrum ceremonies are or did not consider it a value-adding task to the project. Definition of done and acceptance criteria were not practiced by the respondents who do not conduct backlog grooming. Backlog grooming and sprint review ceremonies facilitate customer satisfaction. Sprint retrospective ceremony is there to discuss draw backs and improvements on scrum team [52]. Based on the project requirement backlog grooming will conduct either weekly or biweekly with the engagement of scrum team and product owner. During the backlog grooming backlog items decomposed and elaborate to improve the clarity. Once the requirements were decomposed for smallest story, estimated with everyone's agreement. The sprint review is to get feedback from the client for the work completed. A SME, stated during the follow-up interview "*If team works in comfortable zone, they think conducting retrospective is a waste of time"*. Therefore, it is advisable to be creative and interactively conduct the session and get rid of common questions like *what are the good things happen*, *was there any issues*, and *what the improvements are*, teams can interactively conduct retrospectives [52]. For an example mature scrum team can utilize the time spent on retrospective to review the metric behavior. Use of transparent tool to conduct retrospective suggest by an expert when the team is distributed. As illustrate is Figure 6.2 43% of respondents accept changes at sprint execution. Agile is to be used in often in changing environments. However, changes are not absorbed at the sprint execution. As per one expert stated, "No change allowed during the sprint execution.". Scrum teams does not like to work geographically distributed environments where 79% of the respondents were in that. However, teams do not like

to work in a distributed environment because 37% are neutral. 15% of the respondents' responses highlighted their dislike of working in geographically distributed environments. Moreover, the scrum teams did not use the pair programming concept, and only 45% of respondents knew. Agile teams' life got both more challenging and exciting than in a traditional waterfall model process due to high visibility and transparency.

Table 6.1 illustrates a brief description of the scrum projects. It described project type, number of sprints conducted in the project, and number of weeks per sprint at the time of data collection. There were nine Agile development projects, two Agile maintenance projects, and one Agile testing project. Five agile development projects have executed 10+ sprints, whereas four agile development projects have less than 10 sprints. Both maintenance projects are long-running projects. At the data collection, 15+ sprints had been completed. Further, agile testing projects have executed only five sprints. In agile development projects, other than project 6, the rest

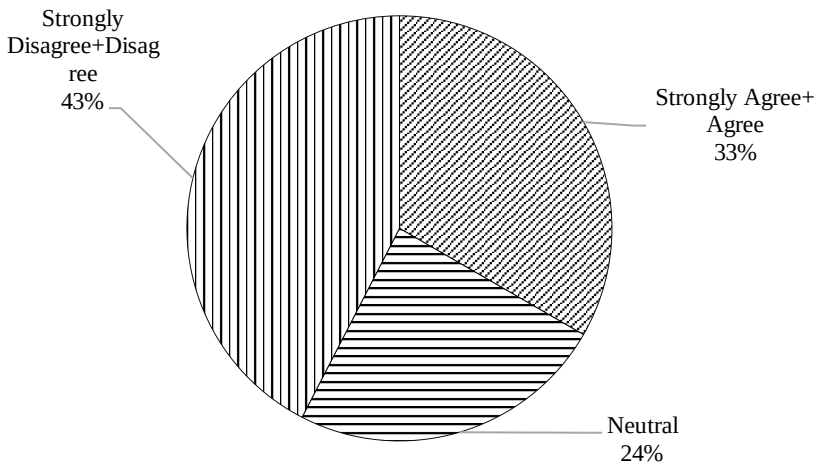

Figure 6.2: Probability of accepting changes during the sprint.

of the projects have two weeks sprint, whereas project 6 has three weeks sprint. In agile maintenance projects, sprints have three weeks and four weeks in projects 11 and 10, respectively. Agile testing projects have a one-week sprint. However, we focused only on nine agile development projects as the first step. Interviews with the nine scrum masters highlighted the usage of unsuitable metrics, improper metrics analysis, lack of awareness, and negative attitude on metric tracking in scrum projects.

Further, the data analysis conducted for the face-to-face interview with nine scrum masters showcased that 94% of the respondents are known to the timeline of the project delivery, project's critical success factors, and criticality of the project business complexity. Results showcased that software metrics helps to improve agile project performance. Eighty-one metrics were collected through literature review and face-to-face semi-structured interviews with nine scrum masters. We removed the Delivery Maturity Index (DMI) because the metric track to identify the project maturity does not relate to the scrum metrics. After that, we removed 14, 28, and 21 metrics that measure for the same purpose, violate the scrum principles and are good to have metrics (not must have), respectively. The respective metrics are listed in Table 6.6. There are 14 metrics. Though the name is different, the purpose of the measurement is the same. ERA insight is a metric tracked using a company-developed tool to measure the code quality. UAT defect benchmark metric, Customer reported bugs and Delivered defect density measure to understand the variation of the defects captured by the client during the UAT for different deliverables. Defect count, Total reported bugs, Fixed/ Solved bugs, Number of critical bugs, Outstanding bugs, and Defect density are to track and manage defects captured by the QA in a sprint. We have identified Sprint burndown chart was addressed in many ways, such as Sprint burn-down, Sprint burn down, Progress chart - kind of kanban board, Burndown charts represent Sprint burndown chart. Team velocity and Velocity is to represent Sprint velocity. Further, Release burndown is there to represent the Release burndown chart.

Under the “metric violates the scrum principles” category in Table 6.4 listed 28 metrics tracked against the agile principles and values. Closing the defect by fixing it within the sprint is the main priority in a scrum project irrespective of the counting number of defects captured [28]. Agile practitioners are uninterested in bugs if they are fixed immediately. Therefore, the Defect severity index, Open defect ageing, Defect to remark ratio, Defect removal efficiency, and Time to find a defect were disregarded. Defect slippage rate, Defect rate, Test success rate, Test failure rate, Test growth ratio, Number of broken builds, Hours spend on bugs, Test coverage, Test execution productivity, Test design productivity, Cost of poor quality, Cost of Quality, Productivity, Effort variance, Team total effective available hours, Team effort remaining, Remaining task effort, Hours spend on tasks, Team total available hours, Number of remaining tasks, Number of complete tasks, and Delivery ratio metrics are measured based on the effort. In that text execution productivity also considered as the test execution progress, and test execution rate. In agile, we do an estimation based on relative size. There were metrics to measure testing efficiencies such as Test execution progress, Test execution rate, Number of tests, and Test design. Agile encourages team collaboration and teamwork rather than measuring individual tasks efficiently. At the same time, we have identified 21 metrics that can be tracked based on the relevancy when the velocity is stable. Tracking many metrics from the start is not advisable in an agile project because the sprint is short, and things happen very fast [17]. Those metrics are Bugs coming from the previous release, Defect burn down, work capacity, Focus factor, Success at scale, Win/Loss record, Reliability, Accuracy of the forecast, Targeted value increase, Ratio of successful sprints, Team happiness, True sprint length, Total available capacity, Test coverage, Accuracy of estimation, Percentage of adopted Work, Percentage of found work.

As illustrated in Table 6.6, we identified basic 17 metrics and three artefacts to track scrum projects on product quality, team productivity, and project predictability. Moreover, we categorized the metrics and artefacts based on the project execution state of the scrum. Definition of readiness metric and Release burn down artefact tracked in

speculate stage. Defect density, Scope change index, Technical debt, Unit test coverage for the developed code, Open defect severity index, Sprint goal status, Commitment ratio, Spillover stories per sprint, Accuracy of estimation, Velocity, Delivered defect density, and On-time delivery metrics tracked in inspect and adapt stage. Scrum maturity index, Definition of done metrics tracked monthly and Net promoter score, Customer delighted index tracked quarterly. We validated the metrics and artefacts in five multiple cases in global IT services and delivery company.

Table 6.3 : Metrics captured in literature review

Metric Name	
1. Customer reported bugs [73]	25. Progress Chart - Kind of kanban board [66]
2. Defect Count [62]	26. Accuracy of estimation [18]
3. Total reported bugs [63]	27. Targeted Value Increase (TVI+) [18]
4. Number of critical bugs [63]	28. Ratio of successful sprints [62]
5. Outstanding bugs [63]	29. Team happiness [73]
6. Fixed/ solved bugs [63]	30. True sprint length [73]
7. Bugs coming from previous release [63]	31. Burndown [67]
8. Hours spend on bugs [63]	32. Team total available hours [63]
9. Test success rate [63]	33. Team total effective available hours [63]
10. Test failure rate [63]	34. Team effort remaining [63]
11. Technical debt [64]	35. Team Velocity [18]
12. Number of broken builds [64]	36. Total available capacity [63]
13. Percentage of Adopted Work [18]	37. Sprint Burn-down [18]
14. Number of Tests [65]	38. Sprint Velocity [18]
15. Test growth ratio [65]	39. Number of complete tasks [63]
16. Test coverage [65]	40. Number of remaining tasks [63]
17. Velocity [18]	41. Remaining task effort [63]
18. Focus Factor [18]	42. Hours spend on tasks [63]
19. Success at Scale [18]	43. Release Burndown [68], [62]
20. work capacity [18]	44. Release burn up chart [68], [62]
21. Win/Loss Record [18]	45. Forecast horizon [73]
22. Reliability [62]	46. Lead time [73]
23. Percentage of Found Work [18]	47. Working delivery software success rate [63]
24. Sprint burn down chart [18]	48. Net Promotor Score [18]

Table 6.4 : Metric Analysis.

Metrics with the different name measure for the same purpose			
#	**Metrics Names [Considered Metric]**		
	Product Quality	**Team Productivity**	**Project Predictability**
1	ERA insight [Technical debt]	Sprint burn down, Sprint burn-down, Progress chart - Kind of kanban board, Burndown [Sprint burndown chart]	Release burndown [Release burndown chart]
2	UAT DD benchmark metric, Fixed/ Solved bugs [Delivered defect density]	Team velocity, Velocity [Sprint velocity]	
3	Customer reported bugs, Defect count, Total reported bugs, Number of cirical bugs, Outstanding bugs [Defect density]		
Metrics violate the scrum principles			
#	**Product Quality**	**Team Productivity**	**Project Predictability**
1	Defect severity index	Cost of poor quality	Delivery ratio
2	Hours spend on bugs	Cost of quality	
3	Number of broken builds	Productivity	
4	Defect slippage rate	Open defect aging	
5	Defect rate	Effort variance	
6	Test success rate	Defect removal efficiency	
7	Test failure rate	Time to find a defect	
8	Number of Tests	Defect to remark ratio	
9	Test growth ratio	Team total available hours	
10	Test coverage	Team total effective available hours	
11		Team effort remaining	
12		Number of complete tasks	
13		Number of remaining tasks	
14		Remaining task effort	
15		Hours spend on tasks	
16		Test execution productivity (Test execution progress or Test execution rate)	
17		Test design productivity	

Good to have Metrics [not must have]			
#	**Product Quality**	**Team Productivity**	**Project Predictability**
1	Bugs coming from previous release	Defect burn down	Release burn up chart
2		work capacity	Forecast horizon
3		Focus factor	Lead time
4		Success at scale	Working delivery software success rate
5		Win/Loss record	
6		Reliability	
7		Accuracy of forecast	
8		Targeted value increase	
9		Ratio of successful sprints	
10		Team happiness	
11		True sprint length	
12		Total available capacity	
13		Test coverage	
14		Accuracy of estimation	
15		Percentage of adopted work	
16		Percentage of found work	

Table 6.5 : Agile SME role and experience.

SME Name	SME Role	SME Experience	Company type and head count
SME 1	Head of QA	13+ years of experience specializing on Testing and Quality Engineering disciplines. Currently heading the Quality Engineering Practice as the head of Quality Engineering.	200+ product-based company
SME 2	Scrum Master	10+ years of experience as a project manager in agile projects. Currently handling scrum-based projects as a scrum master.	300+ product-based company
SME 3	Head of Process	10+ years of experience in ERP development and support spanning a multitude of business domains. Agile process definition and streamlining of existing processes working with multiple teams on organizational improvement initiatives.	400+ product-based company
SME 4	Associate manager Process	10+ years of experience in project management industry.	10000+ employees, product and service base company
SME 5	Manager-Software Engineering	12+ years of experience in project management in software engineering industry.	10000+ employees, product base company
SME 6	Senior quality assurance engineer	8+ years of experience in quality assurance in software engineering industry.	200+ employees, product base company
SME 7	Solution Engineer	11+ years of software development experience across Financial, CRM, Hospital, Clinic, Event Management, and E-commerce etc.	300+ product and service-based company
SME 8	Technical Lead	10+ years of software development experience	300+ product and service-based company
SME 9	Head of delivery	2+ years of experience in project management in software engineering	500+ service-based company

Table 6.6 : Best-fit metrics and artifacts in scrum.

Frequency	**Speculate**	**Explore**	**Inspect and Adapt**	**Monthly**	**Quarterly**
(PQ-Product Quality, TP-Team Productivity, PP-Project Predictability)					
Metrics (M)	1. Definition of Readiness (DoR) - PQ	1. Defect Density (DD) - PQ	1. Open Defect Severity Index (ODSI) - PQ	1. Scrum Maturity Index (SMI) - PP	1. Net Promotor Score (NPS) - PP
		2. Technical debt (TD) [Era Insight (EI)] - PQ	2. Sprint Goal Status (SGS) – PQ	2. Definition of Done (DoD) - PQ	2. Customer Delight Index (CDI) - PP
		3. Unit Test Coverage for the DevelopedCode (UTDC) - PQ	3. Scope Change Index (SCI) - PQ		
			4. Commitment Ratio (CRa) - TP		
			5. Spillover Stories Per Sprint (SSPS) – TP		
			6. Accuracy of Estimation (AoE) – TP		
			7. Velocity (V) - TP		
			8. Delivered Defect Density (DDD) - PQ		
			9. On Time Delivery Index (OTDI) - PP		
Artifacts (A)	1. Release burndown - PP	1. Test Case burndown - TP			
		2. Sprint burndown - TP			
M #	1	3	9	2	2
A #	1	2	0	0	0

In metric identification for the Raban framework, the team mainly based on the best-fit metrics identified for scrum projects because the Raban was a novel process we introduced. ‘Thumbs up rule’ and ‘path flow coverage’ metrics were suggested during the brainstorming session. Further, 12 metrics were identified from the existing scrum metrics and customized to satisfy the RPA project to improve product quality, team productivity, and project predictability. As illustrated in Table 6.7 we identified seven metrics in product quality, three metrics and one artifact in team productivity, and two metrics and one artifact in project predictability. When considering the metric tracking stage, one metric and artifact measure in the speculate stage, two metrics and one artifact measure in the explore stage, five metrics measure in the inspect and adapt stage, one metric measure monthly, and two metric measure quarterly. Then we executed the same for five robotize-sprints in a project.

Definition of readiness metric and Release burndown artifact measured during speculate stage. Defect density, Unit test coverage for the developed code metrics, Technical debt, Path flow coverage, and Sprint burndown chart artifact measured in explore stage. During inspect and adapt stage commitment ratio, the accuracy of estimation, robotize-sprint, delivered defect density, on-time delivery index metrics, and thumbs up rule tracked. The definition of done and quarterly net promoter scores and customer delight indexes is measured every month. Moreover, there are six product quality metrics, four-team productivity metrics, and four project predictability metrics. Finally, we validated the metric model in five RPA transformation projects in global IT services and delivery company.

Table 6.7 : Best-fit metrics and artifacts for Raban.

Frequency	**Speculate**	**Explore**	**Inspect and Adapt**	**Monthly**	**Quarterly**
(PQ-Product Quality, TP-Team Productivity, PP-Project Predictability)					
Metrics (M)	1. Definition of Readiness (DoR) - PQ	1. Defect Density (DD) -PQ	1. Commitment Ratio (CRa) – TP	1. Definition of Done (DoD) - PQ	1. Net Promotor Score (NPS) - PP
		2. Technical debt (TD) [Era Insight (EI)] - PQ	2. Accuracy of Estimation (AoE) - TP		2. Customer Delight Index (CDI) - PP
		3. Unit Test Coverage for the Developed Code (UTDC) - PQ	3. Robotize Sprint Velocity (V) - TP		
		4. Path flow coverage - PQ	4. Delivered Defect Density (DDD) - PQ		
			5. Thumbs-up rule - PQ		
			6. On Time Delivery Index (OTDI) - PP		
Artifacts (A)	1. Release burndown - PP	1. Robotize Sprint burndown - TP			
M #	1	4	6	1	2
A #	1	1	0	0	0

6.4 Summary

This section presents the steps to identify metrics and artifacts for scrum projects. We identified 17 metrics and three artifacts. Those metrics were categorized based on product quality, team productivity, and project predictability. Moreover, they were categorized based on the execution state in scrum, such as speculate, explore, and inspect and adapt. Finalized metrics and artifacts were tracked in five scrum projects to validate. Then, to derive metrics for the Raban framework, brainstorm sessions were conducted with the RPA project team and RPA SMEs to understand measurement points. After that, we customized the identified scrum metrics and artifacts to derive 14 metrics and two artifacts to track and manage RPA projects on the Raban framework.

We have formulated the Raban framework and derived metrics to track and manage RPA projects run on the Raban framework. Hence, we developed the process template for the RPA projects incorporating the Raban framework and the metrics. Then, we discuss the research implication, limitations, and future works.

7 CONCLUSIONS AND RECOMMENDATIONS

This research focused on forming a process template (i.e., software implementation process and metrics to measure the process) for RPA projects to fill the research gap of no process template to execute RPA projects in literature and the industry. Consequently, we conducted our research on the RPA projects in a global IT service and delivery company to formulate the process template for RPA projects. We identified five research objectives such as, 'To investigate the applicability of existing software implementation processes for RPA projects.', 'To formulate a software implementation process to execute RPA projects to reduce additional costs due to change requests identified during the UAT.', 'To derive a tool to identify CBP for RPA transformation.', 'To identify methods to capture customer needs and best practices to improve customer satisfaction.', 'To derive software metrics to track RPA projects to improve product quality, team productivity, and project predictability.'

Initially, the company conducted RPA projects in the waterfall model process in phase I and an in-house developed process model in phase II. However, due to the significant number of change requests and the client-reported defects captured in phases I and II, we formulated a novel process template for RPA projects. Once we formulated Raban's novel process model, we further derived metrics to be used in the RPA projects execute using Raban. For that, initially, we identified best-fit metrics to track and manage scrum projects in product quality, team productivity, and project predictability as a support to identify metrics for Raban. Further, we derived a DST to decide on the CBP for RPA transformation. We developed an estimation process identifying the steps to follow in RPA project use-case estimation. Moreover, we developed a taxonomy to find solutions for the problems and challenges faced by agile practitioners. Further, a framework was developed to enhance the customer satisfaction. We adopt design thinking practices in agile projects. Further, the same was considered when deriving the Raban.

Section 7.1 describes the research implications. Section 7.2 describes the limitations that came across during the research. Finally, in Section 7.3, we discuss future work.

7.1 Research implications

To achieve the first objective (i.e., to investigate the applicability of existing software implementation processes for RPA projects.), initially, we conducted a literature review, and semi-structured interviews with RPA SMEs confirmed that no process template is defined for RPA projects. Then, we examined the applicability of the existing process for RPA transformation projects. As a result, we conducted Phase I of the project using the waterfall model and phase II using the company developed process based on test-driven development. We experienced existing software processes are not suitable for RPA transformation projects. Further, we studied the applicability of the scrum framework for RPA projects and identified though scrum is suitable for product development, the same cannot be used as it is for RPA project execution. Therefore, we identified that existing software implementation processors are not suitable for the RPA project execution.

Then, we focused on achieving the second objective (i.e., to formulate a software implementation process to execute RPA projects to reduce additional costs due to change requests identified during the UAT.) by formulating a software implementation process for the RPA projects named ‘Raban’. Raban reduced additional costs due to change requests identified during UAT. The steps followed in formulating Raban was described in detail in Chapter 4. Once the Raban was formulated, we focused on achieving the third objective (i.e., to derive a tool to select the business processes for RPA transformation.) to improve the selection phase of Raban. We derived a DST tool to identify the CBP. Chapter 5 described in detail the formulation of the DST tool to select CBP for RPA transformation. Then to improve customer satisfaction, we identified customer pain points and best practices to be used in Raban. With that, we

achieved the fourth objective, ‘identify a method to capture customer needs and best practices to improve customer satisfaction’. Section 4.5.2 described in detail the steps followed in capturing customer needs and best practices to improve customer satisfaction. To achieve the fifth objective (i.e., derive software metrics to track RPA projects to improve product quality, team productivity, and project predictability), we first focused on identifying the metrics for scrum projects. After that, we focused on deriving metrics for the Raban framework, because Raban was new and developed based on the agile practice, scrum. We derived 14 metrics and two artifacts to track and manage the RPA projects in the Raban framework. Chapter 6 described in detail the steps followed in metrics identification for projects run on scrum and Raban.

Raban encourages establishing a solid relationship with the client team by conducting recurring ceremonies, such as prioritized business process elicitation, prioritized use-case backlog grooming, and robotize-sprint review. Prioritized business process elicitation was conducted as a workshop attended by the project team, business process owners, process executors, and business SMEs. Prioritized use case backlog grooming was conducted in three workshops to capture the selected business process requirements. At the end of each 2-weeks long robotize-sprint, business process owners, process executors, and business SMEs participated in the robotize-sprint review. Feedback was given for the work completed by the project team. The client had the opportunity to highlight the changes required for the existing requirements at the robotize-sprint review.

Further, continuous collaboration with the client changed the process executors’ negative attitude towards RPA transformation. They had the opportunity to experience the bots during the robotize-sprint review demonstration and had their voice on how the bot should behave. Iterative engagement between the project team and process executors, business SMEs, and business process owners made it easier to capture the changes made to the existing business process by the client system maintenance team. The client has no business value in participating in sprint planning estimation and daily

standup meetings. Therefore, the client maintenance team and other required stakeholders are not needed to participate in those meetings. The client is concerned about the bot execution within a shorter timescale. Considerably, the client could start end-user training early and help change the process executers' negative attitude towards RPA bot implementation. Therefore, rather than going for the big bang approach, iterative and incremental development helps start early training to improve bots' awareness with the process executers and reduce surprises at the UAT.

Typically, the risk of project failure is high at the project's inception. The risk becomes zero only when the bots are deployed to the production and the live environment. In Raban, the project risk reduces much faster because of the iterative and incremental process, which provides high visibility to both parties via a transparent project management tool and recurrent engagement with the client to share project progress and solicit feedback. Proper tool usage improved clear communication between the client and the project team.

Our work was more towards cognitive RPA, where cognitive interaction takes time. Any changes to cognitive RPA bots require model training that takes time. Therefore, capturing changes at the beginning helps train the machine learning models better. Rather than getting a significant volume of test data at the last moment for testing, the iterative implementation process helps the team to get the chunks of data at the subprocess level. Moreover, with the bot's introduction to the business process, new templates might be required to prepare for frequently changing external data (i.e., currency data or table stock auctions), which the client needs to prepare. Hence, training early can help the client's teams get familiar with bots quickly.

Another challenge in RPA implementation is to build trust with the client. When a bot is implemented for a finance module, the client is usually concerned about incorrect transactions, compliance, security policies, standards, and exception scenarios. However, the iterative and incremental implementation in Raban eliminates

those issues by letting the client experience the work completed, even a workable bot, within a short time. Because the project team initially had doubts about the RPA transformation, the risk of project failure was high in phases I and II. The client did not experience the bots until they were deployed to the UAT environment.

Though 100+ scrum metrics can be found in the literature, no one has identified best-fit metrics to be tracked in scrum projects to effectively improve product quality, team productivity, and project predictability. Therefore, as illustrated in Table 6.6, we identified 17 metrics and three artifacts to track and manage projects run on the scrum framework. We track them in the different project execution stages, such as speculating, exploring, inspecting, and adapting. The Definition of readiness metric and release burndown artifact are tracked in the speculate stage. Technical debt, defect density, unit test coverage for the developed code metrics, test case burndown, and sprint burndown artifacts are tracked during the explore state. Open defect severity index, sprint goal status, scope change index, commitment ratio, spillover stories per sprint, the accuracy of estimation, velocity, delivered defect density, and on-time delivery metrics are tracked in the inspect and adapt stage. Metric tracking frequency, monthly or quarterly, needs to be decided based on the metric. For example, while scrum maturity index and definition of done metrics were tracked monthly, net promoter score and customer delighted index were tracked quarterly. Moreover, we track 17 metrics and three artifacts in scrum projects to improve product quality (nine metrics and one artifact), team productivity (six metrics), and project predictability (five metrics and two artifacts).

Further, from those 17 metrics and three artifacts, we derived 14 metrics and two artifacts to track and manage the RPA projects run on the Raban framework as there were no such metrics used within the industry. During the Speculate stage Definition of readiness metric and release burndown chart artifact measured. In Explore stage, Technical debt, defect density, unit test coverage for the developed code metrics, and sprint burndown chart artifact measured. During the inspect and adapt stage commitment ratio, the accuracy of estimation, robotize-sprint, delivered defect

density, and on-time delivery index metrics should be measured. Each month's definition of the done metric will be measured. Quarterly measure of the Net promoter score and customer delight index metrics. Out of the 14 metrics, two artifacts, six metrics focus on product quality, four metrics and one artifact focus on team productivity, and four metrics and one artifact focus on project predictability.

All the business processes are not suitable for RPA transformation. Selecting the wrong process could lead to RPA project failure. However, we could not identify literature related to CBP selection. RPA SMEs from the industry were also unaware of such a tool used. Therefore, we developed a model to predict the business process for RPA transformation, a critical factor determining the RPA project's success. The model was developed based on the 16 factors identified from the literature and semi-structured interviews with RPA SMEs. To predict and train the model for 90% accuracy, two-class decision forest classification model was used. Snowball identified RPA SMEs (5) from the industry to validate the prediction model. Then the prediction model was successfully executed in three projects. Validation confirmed that the proposed tool assists the client in selecting the most suitable CBP.

We developed a taxonomy to provide solutions and best practices for the 15 burning challenges prioritized during the focused group discussions conducted with the agile practitioners in the industry. While the literature had addressed challenges and solutions separately, no one seems to have studied the solutions and best practices for the identified challenges and malpractices of agile practitioners. Moreover, to improve the awareness of agile principles, values, and the scrum framework in the RPA project team, we conducted a four-day workshop followed by a one-hour assessment. RPA project team was formed only with the workshop participants who received the certification issued by the company. Further, we developed an estimation process to estimate RPA use cases. We identified the steps to follow in use-case estimation in RPA projects. RPA SMEs, QA leads, and technical leads were consulted to finalize those steps to follow in the estimation process. RPA team members who

lack experience in RPA projects cannot understand the complexity of an RPA use case in RPA transformation.

The design thinking practices are thinking by doing, a human-centered approach, synthesizing the diverging and converging approach, visualizing, and collaborative work style. We derived a framework for understanding and improving customer expectations by adopting design thinking practices to solve the challenges listed under the client management issues. Other than that, we have identified the best practices to be followed in the software development process to satisfy customer expectations, as illustrated in Table 4.6. We used those practices in formulating Raban, especially at the requirement gathering with the business analysts. Raban framework encouraged RPA SMEs, business SMEs, process executers, business process owners, business process analysts, and RPA project team to participate in the requirement gathering workshops. Which is an iterative process where we develop prioritize use-cases backlog incrementally to make sure the team is spending a reasonable amount of time to capture customer needs. Further, we get feedback during the robotize-sprint review from business SMEs, process executers, business process owners, and business process analysts.

7.2 Research limitations

We completed an RPA transformation project for a multinational financial services company's consumer division. We only had the opportunity to engage with the RPA projects in the banking industry, although it is not the only domain suitable for the RPA transformation. Industries such as technology, manufacturing, healthcare, food and beverage, insurance, and logistics, can be benefited from RPA transformation. However, we could not identify RPA projects run on those industries to support our research. Therefore, it is essential to understand to what extent the Raban framework can apply to such domains and improve. Even in the banking domain, we only had the opportunity to work with a single client. Therefore, we need

to test Raban for other banks. Moreover, we worked only with the RPA project team in the service and delivery company. Hence, we can further improve the Raban framework by executing it in another company's RPA projects. Further, we had the opportunity to test the Raban framework using the Workfusion tool. However, we could not identify RPA projects that operate using other RPA automation tools, such as UiPath, Blue Prism, Automation Anywhere, and Softomotove. The project team can select any RPA automation tool with Raban. The metrics we derived to be measured in RPA projects were only tested on five RPA projects because we could not come across any other RPA projects to validate the metrics. Therefore, we must test them for RPA projects in different domains to generalize the metrics in different circumstances. Better to validate the applicability of the Raban framework and respective metrics for different RPA tools, which we have not tested so far. Moreover, we can test the Raban framework for RPA tools, domains, and companies' combinations.

In the Raban framework, we developed use-cases to reflect the business process scenarios. In RPA projects, expert knowledge and experience are required to estimate the use-cases. Hence, we have developed an estimation process to reduce RPA SMEs' workload in use-cases estimation. However, it was verified only within the global IT services and delivery company's RPA project because we could not identify RPA project execution in the industry to test the estimation process. Therefore, to generalize the proposed estimation process, we need to test the estimation process with another set of RPA project teams.

We developed DST to identify the CBP for RPA transformation. Which we tested only in RPA projects at a multinational financial services company. Those RPA projects were in the banking domain with limited support from the client. We could not identify another RPA project to test the tool from the industry. Hence, we need to test and validate the RPA projects run on other domains. Moreover, we can improve the taxonomy we developed to provide solutions or best practices for the 15 priority challenges identified. We can extend the solutions and best practices for the rest of the

65 challenges. We can also generalize the framework developed to understand and improve customer satisfaction using design thinking practices by executing the framework in more projects. Our findings might be limited because we could identify only 10 companies familiar with design thinking practices. However, if we can execute the research across more companies in the IT sector, we might capture more solutions and best practices to enhance the taxonomy.

Though the author played a role as a process consultant for the projects based on the research, there was no impact on the research outcome. Because process consultants in the company assess the project execution against company standards and guide the project team, they are not involved with the project tasks carried out against project delivery.

7.3 Future work

As stated in the research limitations above, we tested the Raban framework for RPA projects run on the Workfusion tool. The industry has different RPA transformation tools. While Raban has no dependency on the tool used for RPA transformation, we have not verified it empirically. Therefore, we can verify whether the Raban framework can execute the RPA transformation tool in RPA projects.

Moreover, we executed the Raban framework for projects in the banking domain. Banking and finance are not the only industries enabled for RPA transformation. Technology, manufacturing, healthcare, food and beverage, insurance, and logistics are the other industries we can do RPA transformation. We can improve the Raban framework by executing it in other domains.

Furthermore, use-cases estimations in RPA projects are critical as they require expert knowledge and experience to understand the complexity. The RPA team is using the estimation process defined by the RPA SMEs. We need to test the estimation

process for RPA projects in other domains or the exact domains in another company and technologies. Consequently, we can standardize the RPA estimation process.

RPA project selection for business process transformation is a critical task. Because we worked with the client who engaged in the banking domain, we got limited intervention for CBP selection for RPA transformation. Hence, we worked with limited client support for DST verification. We can validate the tool and improve accuracy by executing it for more projects. Other than that, we can enhance the tool by integrating artificial intelligence. The DST does not depend on the RPA transformation tool. However, we have not validated it through empirical study, which was a pending task. We could execute the tool in another domain to generalize. Likewise, the model we developed to improve customer satisfaction by adopting design thinking practices also improves after executing it in another domain.

REFERENCES

[1] T. Reuner, “HfS Blueprint Report”, Excerpt for Accenture, 2018.

[2] A. Asatiani and E. Penttinen, “Turning robotic process automation into commercial success - Case OpusCapita,” *Journal of Information Technology*, p. 67-74, 2016.

[3] C. Kroll, A. Bujak, V. Darius, W. Enders, and M. Esser, “Robotic process automation - Robots conquer business processes in back offices,” Capgemini Consulting and Capgemini Business Services, 2016.

[4] S. Z. Jovanović, J. S. Đurić Tatjana, and V. Šibalija, “Robotic Process Automation: Overview and opportunities,” *International Journal Advance Quality*, vol. 46, pp. 3-4, 2008.

[5] S. Aguirre, A. Rodriguez, "Automation of a business process using robotic process automation (RPA): A case study," in *Workshop on Engineering Applications*, 2017.

[6] L. Willcocks, M. Lacity, and A. Craig, “Robotic process automation at Xchanging,” The London School of Economics and Political Science, 2015.

[7] C. Lamberton, A. Gillard, and G. Kaczmarskyj, “Get ready for robots – Why planning makes the difference between success and disappointment,” EYGM Limited, 2016.

[8] J. Watson and D. Wright, “The robots are ready. Are you? Untapped advantage in your digital workforce,” Deloitte University Press, 2017.

[9] S. Jalali, C. Wohlin, and L. Angelis, “Investigating the applicability of agility assessment surveys: A case study,” *Journal of Systems and Software*, 2014.

[10] R. Holler *et el*., “11th annual state of agile survey,” VersionOne, 2016.

[11] R. Hoda, N. Salleh, and J. Grundy, “The rise and evolution of agile software development,” IEEE Software.

[12] L. A. Cooper, D. K. Holderness, and T. L. Sorensen, “Robotic process automation in public accounting,” Accounting Horizons, 2019.

[13] R. Dilla, H. Jaynes, and L. Livingston, “Introduction to robotic process automation - A primer,” Institute for Robotic Process Automation, 2015.

[14] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in Proc. 9th international conference on Software Engineering, 1987.

[15] K. Beck, J. Sutherland, K. Schwaber, M. Fowler, and A. Cockburn, “Mani-Festo for agile software development,” 22 04 2020. [Online]. Available: URLhttp://agilemanifesto.org/.

[16] T. Dyb˚a and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," in information and software technology, 2008.

[17] K. Schwaber and J. Sutherland, “The scrum guide - The definitive guide to scrum: The rules of the game,” 2020.

[18] S. A.S. J. Downey, “Scrum Metrics for Hyper-productive Teams,” In Proc. System Sciences (HICSS), 2013 46thHawaii International Conference,Hawaii, 2013.

[19] S. L. Eeger, “Software Metrics: Progress after 25 Years?,” IEEE Software, 2008.

[20] M. S. Rawat, A. Mittal and S. K. Dubey, “Survey on Impact of Software Metrics on Software Quality,” *(IJACSA) International Journal of Advanced Computer Science and Applications*, 2012.

[21] K. Hass, “The Blending of Traditional and Agile Project Management,” PM World Today, 2007.

[22] E. Kupiainen, M. V. Mantyla, and J. Itkonen, “Using Metrics in Agile and Lean Software Development – A Systematic Literature Review of Industrial Studies,” *Journal of Information and Software Technology,* 2015.

[23] C. Jones, “Software Defect Removal Efficiency,” Computer Journal,Vol. 29 ,PP. 94-95, 1996.

[24] J. Purcell, “Comparison of Software Development Lifecycle Methodologies,” in Information Systems Security Professional Consortium, 2007.

[25] J. Gustafsson, “Model of Agile Software Measurement: A Case Study: Master of Science Thesis in the Programme Software engineering,” Chalmers University of Technology, Sweden, 2011.

[26] N. Oza and M. Korkala, “Lessons Learned In Implementing Agile Software Development Metrics,” in Proc. UK Academy for Information Systems Conference Proceedings, 2012.

[27] M. Agarwal and P. R. Majumdar, “Tracking Scrum projects tool, Metrics and Myths about agile,” *International Journal of Emerging Technology and Advanced Engineering*, 2012.

[28] K.V. J. Padmini, H. M. N. Dilum Bandara, and G.I.U.S. Perera , "Use of Software Metrics in Agile Software Development Process," in Proc. Moratuwa Engineering Research Conference (MERCon), 2015.

[29] K. E. Emam, “A Methodology for Validating Software Product Metrics,” in NRC Publications Archive (NPArC), 2002.

[30] M. Kunz, R. R. Dumke, and N. Zenker, “Software Metrics for Agile Software Development,” in Proc. 19th Australian Conference on Software Engineering (ASWEC '08), 2008.

[31] D. S. Vacanti, “Actionable Agile Metrics for Predictability,” 2016.

[32] C. Melo, D. S. Cruzes, and F. K. R. Conradi, “Agile Team Perceptions of Productivity Factors,” In Proc. Agile Conference (AGILE), 2011.

[33] A. Hussain, E. Mkpojiogu, and F. Kamal, “The Role of Requirements in the Success or Failure of Software Projects,” International Review of Management and Marketing 6(2016), 2016.

[34] H. Lotta and M. Lasskso, “Design Thinking in the Management Discourse: Defining the elements of the Concept,” In Proc. 4th World Conf. on Design Research (IASDR ‘11), 2011.

[35] L. Tilmann, C. Meinel, and R. Wagner, “Design thinking: A fruitful concept for it development?,” *Design* Thinking, Springer, 2011, pp. 3-18.

[36] C. Kevin and R. Smith, “Unleashing the Power of Design Thinking,” Design Management Review, vol. 19, no. 3, pp. 8-15, 2008.

[37] M. Bethany, “Corporate implementation of design thinking for innovation and economic growth,” *Journal of Strategic Innovation and Sustainability,* vol. 10, no. 2, 2015.

[38] D. Toro, B. Jiménez, R. Cortés, and T. Bonilla, “A requirements elicitation approach based in templates and patterns,” 1999.

[39] Piccardo, *et el.*, “QualiCEFR: A quality assurance template to achieve innovation and reform in language education through CEFR implementation.,” Learning and Assessment: Making the Connections, 2017.

[40] J. Münch, O. Armbrust, M. Kowalczyk, and M. Soto, “Software Process Definition and Management London,” Springer, 2012.

[41] S. Kapferer, “Empirical Research in Software Engineering,” Doctoral dissertation, HSR, 2019.

[42] A. Borges *et el.*, “Support mechanisms to conduct empirical studies in software engineering: a systematic mapping study,” in Proc. of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE' 15), 2015.

[43] Li Zhang et el., "Empirical Research in Software Engineering — A Literature Survey," Journal of Computer Science and Technology, vol. 33, no. 5, p. 876–899, 2018.

[44] C. Wohlin and A. Aurum, “Towards a decision-making structure for selecting a research design in empirical software engineering,” *Empirical Software Engineering,* vol. 20, no. 6, pp. 1427-1455, 2015.

[45] A. Devine, “Smart process automation: The why, what, how and who of the next quantum leap in enterprise productivity,” Institute for Robotic Process Automation and Artificial Intelligence, 2017.

[46] G. B. Glaser and A. L. Strauss, “The discovery of grounded theory: Strategies for qualitative research,” Chicago: Aldine, 1967.

[47] A. Strauss and J. Corbin, “Basics of qualitative research: Techniques and procedures for developing grounded theory,” in *Saga Publication*, Thousand Oaks, 1998.

[48] KVJ Padmini, GIUS Perera, and HMND Bandara, “A decision support toolto select candidate business processes in robotic process automation (RPA):An empirical study,” in Proc. *2019 3rd SLAAI - Int. Conf. on Artificial Intelligence*, 2019.

[49] N. Merican, “Exploring robotic process automation as part of process improvement.,” 2016.

[50] M. Lacity, L. P. Willcocks, and A. Craig, “Robotic Process Automation at Telefónica O2” The London School of Economics and Political Science, 2015.

[51] J. Bernes, Azure machine learning: Microsoft Azure Essentials, Microsoft Press, 2015.

[52] E. Derby and D. Larsen, “Agile retrospectives: Making good teams great!” Agile 2007, 2007.

[53] D. R. Greening, “Agile Enterprise Metrics,” in Proc. 48th Hawaii International Conference on System Sciences, 2015.

[54] “Agile Metrics : Let the Numbers tell the tale,” Prowareness [Online], December 5 2021. Available: http://www.scrumdayeurope.com/: http://www.scrumdayeurope.com/prowareness/website/scrumday.nsf/Agile_Metrics.pdf

[55] T.H. Cheng, S. Jansen, M. Remmers, “Controlling and Monitoring Agile Software Development in Three Dutch Product Software Companies,” in Proc. *ICSE’09 Workshop*, 2009.

[56] P.S.M.D. Santos *et el.,* “Visualizing and Managing Technical Debt in Agile Development: an Experience Report,” in Proc. *International Conference on Agile Software Development*, 2013.

[57] J. Johns *et el.,* “The 3C Approach for Agile Scrum Software Methodology,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 3, 2014.

[58] M. Agarwal and R. Majumdar, “Tracking Scrum projects Tools, Metrics and Myths About Agile,” *International Journal of Emerging Technology and Advanced Engineering,* vol. 2, no. 3, 2012.

[59] Y. Dubinsky, D. Talby, O. Hazzan, and A. Keren, “Agile metrics at the Israeli Air Force,” in *Agile Conference,* 2005.

[60] W. Hayes *et el.*, “Agile Metrics: Progress Monitoring of Agile Contractors,” Software Engineering Institute, 2014.

APPENDIX I

Appendix I

*Required

Survey on Challenges Faced by Agile Team Members

Dear Friends,

We are conducting a research study to analyse the challenges faced by the Agile teams in Sir Lanka. We are inviting you to participate in this study by completing the following questionnaire. It will take about ~15 minutes to complete the survey.

This survey is stipulated confidential and anonymous. Your responses will not be identified with you personally and all findings will appear in aggregated form. You and your organization will not be linked in any manner.

Your participation in the research would be greatly appreciated. If you have any queries or wish to know more please feel free to contact us using the details provided below.
Thank you very much for your time and help in making this study possible.

Sincerely,
Jeeva Padmini, Pavithra Kankanamge, and Dilum Bandara
Dept. of Computer Science and Engineering,
University of Moratuwa
Sri Lanka.

If you are a Scrum team member, please answer the following questions.

1. My project is conducting; *

Tick all that apply.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
daily stand-up ceremony on time for ~15 minutes every day.	☐	☐	☐	☐	☐
sprint planning ceremony before the start of sprint in every sprint.	☐	☐	☐	☐	☐
sprint retrospective ceremony at the end of each sprint.	☐	☐	☐	☐	☐
sprint review ceremony at the end of each sprint with the product owner.	☐	☐	☐	☐	☐
backlog grooming ceremony every week.	☐	☐	☐	☐	☐

Metrics tracked in your project

2. What are the metrics tracked in your project?

3. Rate the value those metrics bring to the project:

Mark only one oval.

	1	2	3	4	5	
No Value	◯	◯	◯	◯	◯	Very High Value

4. List down any metric you propose

Scrum team

5. As a scrum team *

Tick all that apply.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
we are not accepting any changes within the sprint.	☐	☐	☐	☐	☐
we always maintain the project velocity.	☐	☐	☐	☐	☐
our story acceptance criteria is properly defined in the story.	☐	☐	☐	☐	☐
"definition of done" for our project is properly defined.	☐	☐	☐	☐	☐
we properly define the story.	☐	☐	☐	☐	☐

6. My project scrum team *

Tick all that apply.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
is geographically distributed.	☐	☐	☐	☐	☐
likes to work in a geographically distributed environment.	☐	☐	☐	☐	☐
is using user friendly tools to track and manage the project.	☐	☐	☐	☐	☐
is using transparent tools such as JIRA, RALLY, VERSIONONE, etc., effectively.	☐	☐	☐	☐	☐

7. My project scrum team members *

Tick all that apply.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
are collaborative.	☐	☐	☐	☐	☐
are skillful.	☐	☐	☐	☐	☐
have internal politics.	☐	☐	☐	☐	☐
participate for team building activity every month.	☐	☐	☐	☐	☐
physical presence at office is not much important.	☐	☐	☐	☐	☐
emphasize more on quality standards (CMMI, ISO, and other maturity models)	☐	☐	☐	☐	☐
are good in pair programming.	☐	☐	☐	☐	☐

8. Agile testing team members *

Tick all that apply.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
have sufficient time to complete QA work on-time with expected quality.	☐	☐	☐	☐	☐
have sufficient time to write comprehensive test cases.	☐	☐	☐	☐	☐
include a separate team to automate the manual test cases.	☐	☐	☐	☐	☐
spend additional time on updating the automation team.	☐	☐	☐	☐	☐
spend additional time on updating the regression test suite.	☐	☐	☐	☐	☐
are experience QA spill overs in most of the sprints.	☐	☐	☐	☐	☐

9. I know about my project's *

Tick all that apply.

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
critical success factors.	☐	☐	☐	☐	☐
high-level plan and its milestones.	☐	☐	☐	☐	☐
complexity and business criticality.	☐	☐	☐	☐	☐

10. Please list down any other challenges that you may have experienced/observed?

Demographic data

Please provide following information:

11. What is your age? *

Mark only one oval.

- below 20 years
- 20 - 30 years
- 31 - 40 years
- 41 - 50 years
- above 50 years

12. What is your gender? *

Mark only one oval.

- Male
- Female

13. How many years of experience you have in Software Industry? *

Mark only one oval.

- ◯ Less than 1 year
- ◯ 1-3 years
- ◯ 3 – 5 years
- ◯ 5-10 years
- ◯ 10-15 years
- ◯ More than 15 years

14. For how long you have worked on projects based on agile framework? *

Mark only one oval.

- ◯ Less than 1 year
- ◯ 1-3 years
- ◯ 3 – 5 years
- ◯ 5-10 years
- ◯ 10-15 years
- ◯ More than 15 years

15. What software development methodologies were adopted by projects you were involved in the past? *

Tick all that apply.

- ☐ Waterfall
- ☐ agile
- ☐ both

16. What is your role in the current agile project? *

APPENDIX II

Face-to-face Questions for Design thinking practices to satisfy customer expectations

Geographical Information:

1. Name
2. Designation
3. Gender:
4. Experience in Agile projects:
5. No of Agile projects engaged with:

Common interview questions:

1. Could you explain your role? Focus on day-today activities and responsibilities of the project. Focus to identify what he is doing in the project.
2. What type of agile methodology you have experienced with (Scrum, XP, TDD, Hybrid)?

Human-centered approach / Empathy

1. What are the type of human centered approach techniques used to get to know about the customer's real needs (expectations) when you are gathering requirements?

   Possible answers:
   • Persona techniques to identify the characteristics of the end-user (age, location, education etc.) look into who using this product?
   • Scenario based approach is product related scenarios, how they use the product.
   • Use cases to understand how the end-user interact with the real world
   • Consumer journey mapping to understand the customer experience when using the product

2. How are you practicing the above mention technique? Who are the participants?
   <Ex: Product owner, Dev team, Scrum master, BA >

3. Did you achieve any advantages using above mentioned techniques and what is your recommendations to use this techniques?

Thinking by doing / Action based

1. What are the type of "thinking by doing practices" for getting feedback before going for actual implementation

Ex: Create prototypes, POCs for the requirements and for the initial idea and getting feedback before going for actual implementation?

2. Advantage and benifits of following "thinking by doing approach" instead of directly going for actual implementation?

3. What kind of situations that can use "thinking by doing approach"? Ex: Prototypes and POCs

Visualizing

1. What are the techniques that you are using to visualize the idea as tangible in your project?

Ex: Prototypes, POC or graphical format (UX Design)

2. What are the advantages and benifits of using visualizing your idea as tangible?

3. What kind of situations that you have applied thius visualizing DT practice in your project?

Synthesis

1. Are you using divergent approach to come up with multiple alternative ideas instead of taking initial idea as the best?

2. How about using the convergent approach used to identify the best solution out of several solutions?

3. What kind of situations have you applied this divergent and convergent approaches in your project?

1. Are you brainstorming the work with your team? The situations that brainstorming is necessary?
Ex: KJ method (group consensus technique)

2. Who are the participants for practicing collaborative work style? How they are involving?

3. Have you involved your customer to your work to help you understand the problem and what kind of situations?

Satisfy customer expectation projects based on agile

1. According to your understanding, what are the pain points of the customer journey
Ex: understand real problem, maximize the value of the product, competitive advantage or user experience etc.

2. Have you considered that both internal (time, cost and scope) and external (customer and the end users expectations and social) factors should be focus on to satisfy customer expectations in your project?

3. According to your understanding, what are the ways and means to satisfy customer in your projects?

APPENDIX III

Appendix III

*Required

Decision support tool to select candidate business process for Robotic Process Automation (RPA)

Dear Sir/Madam,

We are conducting a research study to verify a decision support tool developed to select candidate business processes for Robotic Process Automation (RPA). As a RPA expert, we are inviting you to participate in this study by completing the following questionnaire. It will take about ~10 minutes to complete.

This survey is stipulated confidential and anonymous. Your responses will not be identified with you personally and all findings will appear in aggregated form. You and your organization will not be linked in any manner.

Your participation in the research would be greatly appreciated. If you have any queries or wish to know more please feel free to contact us using the details provided below.

Thank you very much for your time and help in making this study possible.

Sincerely,
Jeeva Padmini and Dilum Bandara
Dept. of Computer Science and Engineering,
University of Moratuwa
Sri Lanka.
{dilumb, jeeva.12}@cse.mrt.ac.lk

In answering the following questions please consider one of the business processes that you have contributed to automate or will be contributing.

1. What is the state of candidate business process that you have selected? *

Mark only one oval.

- ◯ Bots are developed and business is satisfied
- ◯ Bots are developed and business needs are NOT satisfied
- ◯ Bot development project is still in progress
- ◯ Planing to develop bots in the future

As per my own assessment the selected candidate business process has the following process factors:

2. Volume of Transactions *

i.e., no of processing requests that the computing system will receive and respond to within a specified period of time

Mark only one oval.

- ◯ High
- ◯ Medium
- ◯ Low

3. Business Process Complexity *

e.g., levels of ability to analyze, variety, non-routines, difficulty, uncertainty, and inter-dependence with other business processes

Mark only one oval.

- ◯ High
- ◯ Medium
- ◯ Low

4. Rate of Change *

i.e., no of changes make to the business process within a month

Mark only one oval.

- ◯ 2 or less
- ◯ 3 to 5
- ◯ 6 to 8
- ◯ More than 8

5. Rules-Based Business Process *

i.e., logic-based business process

Mark only one oval.

- ◯ Yes, with a few exemptions
- ◯ Yes
- ◯ No

6. Workload Variation *

i.e., irregularity of workload where additional staff support is required during peak times

Mark only one oval.

- ◯ Yes
- ◯ Sometimes
- ◯ No

7. Regulatory Compliance *

i.e., need to satisfy organizational, government or any other regulations, e.g., PCI-DSS or ISO

Mark only one oval.

- ◯ Yes
- ◯ No

As per my own assessment the selected candidate business process has the following external factors:

8. Cognitive Features *

i.e., task require creativity, subjective judgment, or complex interpretation skills (e..g., natural language processing and speech recognition)

Mark only one oval.

- ◯ Yes
- ◯ No

9. No of Systems to Access *

e.g., automation task involves accessing multiple systems

Mark only one oval.

- ◯ 2 or less
- ◯ 3 to 5
- ◯ 6 to 8
- ◯ More than 8

10. Multi-modal Inputs *

e.g., images, audio clips, or videos are given as inputs

Mark only one oval.

- ◯ Yes
- ◯ No

11. Data Driven *

i.e., decisions during an activity is made based on the analysis of data.

Mark only one oval.

- ◯ Yes
- ◯ No

12. Stability of Environment *

i.e., availability of UAT or any other near-production environment of the target application for the automation.

Mark only one oval.

- ◯ Yes
- ◯ No

As per my own assessment selected candidate business process need to consider following financial impact factors:

13. Operational Cost *

i.e., cost of day-to-day maintenance and administration

Mark only one oval.

- ◯ High
- ◯ Medium
- ◯ Low

14. Impact of Failure *

i.e., financial impact to client if the bot goes wrong

Mark only one oval.

- ◯ High
- ◯ Medium
- ◯ Low

15. Human Error *

e.g., task is error prone due to human worker negligence than inability to code as a computer program

Mark only one oval.

- ◯ High
- ◯ Medium
- ◯ Low

16. End of Life *

i.e., How soon the bot will become obsolete, e.g., new system is on the road map of client

Mark only one oval.

- ◯ 2 years or less
- ◯ 4 years or less
- ◯ More than 5 years

17. Service Level Agreement *

i.e., Time taken to complete end-to-end functionalities of a task.

Mark only one oval.

- ◯ Seconds
- ◯ Minutes
- ◯ Hours
- ◯ Days

Thank you so much for your support!

Please also answer the following questions for us to better interpret the results.

18. I am having; *

Mark only one oval.

- 5+ years experience in RPA
- 3+ years experience in RPA
- 1+ years experience in RPA
- Less than 1 years experience in RPA
- No experience in RPA

19. I have played/playing the following role(s) in RPA project; *

Tick all that apply.

- Business Analyst
- Project Manager
- Developer
- Tester

Other: _____

20. Please share your e-mail if you would like to be informed about our findings.

_____

_____

_____

_____

_____

21. Any comments you like to share about RPA and selecting a candidate business process?

This content is neither created nor endorsed by Google.

Google Forms

APPENDIX IV

#	Metric Title	Metric Description	How to measure	Reference
1	Accuracy of Estimation	Rreflects the Team's ability to correctly estimate the body of work during Sprint Planning	1 - (Total (Estimate Deltas)/ Total Forecast)	[18]
2	Cost Of Quality	Percentage of effort spent on all activities other than core development activities such as requirements, architecture, design, and coding over the total project effort	Cost of Quality = Total Quality Effort [Submitted Time] in the internal release * 100 / Total Effort [Submitted Hrs] in the internal release	Interview
3	Cost of Poor Quality	Percentage of effort spent on fixing the defects in the software products (such as incorporating review comments in the artifacts and fixing defects discovered in testing) over the total project effort	Cost of Poor Quality = Total Rework Effort [Submitted Hrs] in the client release * 100 / Total Effort [Submitted Hrs] in the client release	Interview
4	Commitment Ratio	Ratio of number of story points delivered (as per Definition of Done (DoD)) to the total number of story points committed during the sprint planning.	Total number of story points delivered (as per DoD) in an iteration (or Sprint) / Total number of story points committed for an iteration (or sprint) during the Planning * 100	Interview
5	Customer Delight Index	Customer satisfaction survey	(Online survey questionnaire developed by the company.)	Interview

#	Metric Title	Metric Description	How to measure	Reference
6	Defect burndown chart	A chart representing the number of defects captures against the sprint	-	Interview
7	Defect Density	Number of defects per story point of the sprint	Defect Density = Total Defects in the iteration * 1000/ Measured Size in SP for the iteration	Interview
8	Delivery maturity index	Metric to measure how much project are compliance with I	-	Interview
9	Defect Removal Efficiency	To understand the efficiency of the team in detecting the defects induced into the system and take necessary steps to improve the same	Total Number of company found, product Defects in the client release * 100 ÷ (Total Number of company found, product Defects in the client release + Total Number of Client Reported product Defects in the client release)	Interview
10	Defect Rate	Total number of internal engineering defects per 1000 Person Hours of project engineering effort (including the development of Requirements, Architecture, Design and Coding)	Number of Defects found during all engineering reviews and developer, integration and QA testing in the Release / Project	Interview
11	Defect Severity Index	Number of defects added during the sprint	Defect Severity Index (Product) = Weighted Count of product Defects in the project / Total number of product defects in the project	Interview
12	Delivered defect density	Number of client reported defects per unit size of the application	Count of Defects = Show stopper Defects + High Defects + Medium Defects + Low Defects	Interview

#	Metric Title	Metric Description	How to measure	Reference
13	Defect slippage rate	Number of production defects	-	Interview
14	Delivery ratio	Ratio of features done in planned release schedule	-	Interview
15	Defect to remark ratio	Ratio of number of Defects (Remarks accepted as Defects by the development team) to the total number of Remarks raised by the Test Team	Defect to Remark Ratio = Total number of defects in the project * 100 / Total number of Remarks reported in the project	Interview
16	Effort Variance	Percentage deviation of actual effort spent on the project over planned effort	Actual Effort (Submitted time) = Total actual effort (Submitted) for the release (internal / client / Project)	Interview
17	Era insight	Company developed metric to measure product code quality	-	
18	Open defect severity index	Number of open defects at the time of sprint end.	Open Defect Severity Index (Product) = Weighted Count of open Defects in the project / Total number of product defects in the project	[17], Interview
19	Open defect aging	The time difference between the time of defect open to fix.	-	Interview
20	Ontime delivery(OTD) index	Percentage of total client milestones that are met (delivery made) on or before the planned dates.	OTD Index = Total number of client milestones delivered on or before the current baseline delivery date till date in the project * 100 / (Total client milestones planned till date for the project + completed client milestones in the project)	Interview

#	Metric Title	Metric Description	How to measure	Reference
21	Productivity	Amount of functionality developed / delivered in terms of number of function points per person month of effort spent on the project sans the training effort.	Productivity (Development) = (Total effort (submitted hours) logged in the project - Training Effort (submitted hours) logged in the project) / Size in Function Points developed in the project	Interview
22	Release burndown			Interview
23	Requirement clarity index / Definition of readiness	Average of requirement clarity index measured for each of the requirements on a scale of 1 to 5	Sum of requirement clarity index measured for each of the requirements on a scale of 1 to 5 for internal releases / Number of requirements for internal releases	Interview
24	Sprint burndown chart	Effort burndown Chart developed for relevant sprint	-	Interview
25	Spill over per sprint	Number of unfinished stories move to another sprint.	-	Interview
26	Sprint Velocity	∑ of original estimates of all accepted work	∑ of original estimates of all Completed work in the sprint	[18] / Interview
27	Time to find a defect	Average time taken to detect a defect during the code review	Time to find a defect - Code review = Effort [Submitted Hrs] put in for in the project / Code review defects in the project	Interview
28	Test case burndown	Chart developed number of test case execution against the dates	-	Interview

#	Metric Title	Metric Description	How to measure	Reference
29	Test coverage	Number of test cases executed	Test coverage – Number of test cases executed/ Number of test cases developed	Interview
30	Test design productivity	Efficiency of designing the test cases.		Interview
31	Test execution productivity/ Test execution progress/ Test execution rate	Test case execution efficiency.	Number of test cases executed	Interview
32	UAT DD Benchmark	Number of defects left	-	Interview
33	Unit test coverage for the developed code	Verify whether the product code was tested by writing unit test cases.	Number of unit test cases written for the developed code	Interview