Ankit Bhutani

Recommendations to Improve the Software Testing Process of the Case Company

Helsinki Metropolia University of Applied Sciences

Master’s Degree

Industrial Management

Master’s Thesis

01 Jun 2021

Preface

This has been a tough year for everyone and balancing work with studies made it a little more challenging, but it was a great learning experience without any doubt. The whole process of conducting the study has been quite a journey and I have learned a lot about research and academic work.

I am thankful to the case company management for providing me the support to balance my work and student life. Without their support, this wouldn’t have been possible. I want to thank my wonderful colleagues for providing their valuable inputs to the study and their time for the interviews despite being so busy at work. I also thank Abhishek Mohan for providing his time and expert insights for the study.

I would like to thank my instructor Dr. Thomas Rohweder for his guidance throughout the study. I would also like to thank Sonja Holappa for helping me with academic writing and Dr. Juha Haimala and Dr. James Collins for their valuable feedback at each stage of the study.

I would also like to thank my family and friends for their continuous support every day.

Ankit Bhutani
Helsinki
June 1, 2021

Author Title Number of Pages Date	Ankit Bhutani Recommendations to Improve the Software Testing Process of the Case Company 58 pages + 1 Appendix 01 june 2021
Degree	Master of Engineering
Degree Programme	Industrial Management
Instructors	Dr Thomas Rohweder, Principal Lecturer

The case company is a global IT solution provider in the life sciences industry. Market competition is forcing organizations to deliver better quality products faster. The case company has invested a lot in test automation to be able to release faster and facing some of the challenges brought by test automation. The objective of the thesis is to generate recommendations to improve the software testing process in the case company.

The study utilizes design research methodology and is conducted in four stages. In the first stage, the strengths and weaknesses of the process are identified by conducting a current state analysis of the process. In the second stage, ideas from the literature are searched to tackle the identified weaknesses. The third stage focuses on the co-creation of recommendations with the key stakeholders in the process and an external expert. In the fourth and the final stage generated recommendations are validated by the senior managers of the research and development department. Also, a recommendation plan was created.

The study uses qualitative methods for data gathering. Stakeholder interviews and internal process documents were the main sources of data. The identified weaknesses were divided into four categories namely process operations, knowledge transfer, key performance indicators, and process improvement. The recommendations are suggested to tackle weaknesses in each category and a redesigned process is suggested.

The outcome of the thesis is a list of recommendations that provide practical solutions to tackle inefficiencies in the current testing process. The proposed process improvements will help the case company develop better quality software products faster when implemented. Some of the recommendations require detail planning while others can be taken into use with minimum time investment. Some elements of the recommendations are transferable and can be utilized by other I.T organizations to improve their software testing process.

Keywords	Software Testing, Testing Process, Test Automation, Agile

Contents

List of Figures

1 Introduction

Software testing is an important part of the entire software development life cycle. Software errors cost enterprises around $61 billion in salary costs alone in North America (Cambridge Judge Business School, 2020). Software testing ensures that products meet the quality standards and work as expected. Poorly tested software not only can compromise the safety and security of the products but also can put the business at risk.

Market competition is forcing organizations to develop better software and deliver it faster to market which requires organizations to optimize their software development including the testing process which can account for almost half of the overall effort to develop the software (Boris Beizer, 2002). Many organizations have invested in automation testing of the software products which reduces the overall testing times but brings some challenges due to lack of a proper automation strategy and poor implementation.

The case company recognizes the urgent need for optimizing the software testing process as it is putting the strategic aims of releasing the product faster to market at risk. The case company is also dealing with some of the challenges brought by automation testing and looking for ways to improve it. An inefficient testing process can also put the quality of the products in jeopardy. Therefore, improvements in the testing process are needed.

1.1 Business Context

The case company is a global IT solutions provider for the highly regulated life sciences industry. The Company develops software products that help sponsors and pharmaceutical companies conduct clinical trials.

The Global Research and development team is responsible for developing, maintaining, and testing the software products. Testing of the products is carried out using Manual and Automated tests and the Quality Assurance team is mainly responsible for all the testing activities that include creation, execution, and maintenance of these tests.

1.2 Business Challenge, Objective and Outcome

The case company has invested a great deal of money in the development of new software products over the last few years and has been utilizing automation testing for the testing of those products but still some projects are heavily dependent on manual testing. Projects that do utilize automation testing are observing an increase in the time it takes to execute those tests as the complexity of the products increase. These issues are impacting the overall time it takes to release a newer version of the products which is typically done every 4-6 weeks.

The Increased complexity of the products has also made it difficult to maintain these tests and the tests are even showing false negatives at times and thus development hours are wasted in rewriting these tests and debugging issues are impacting the team's confidence as well. These challenges are hindering the company's plan to release products more frequently to maintain the competitive edge in the market.

The objective of this thesis is to generate recommendations on how to improve the testing process and the outcome of the thesis is the recommendations for improving the testing process in the case company. The outcome is intended to enable the case company to improve the testing process.

1.3 Thesis Outline

The thesis has four stages to address the business challenges introduced above. The first stage is to analyze the current state to identify the strengths and weaknesses of the testing process. Quality management system documents, interviews with stakeholders, past project data, and regulatory requirements are the main source of data for the current state analysis.

After the current state analysis existing literature knowledge is researched for tackling the identified weaknesses. The next stage is to generate recommendations with the help of internal stakeholders and external expert interviews. The fourth and last stage is to validate the initial recommendations based on feedback from senior management to come up with final recommendations to improve the process.

The thesis includes 7 sections starting with the introduction to the subject. Section 2 explains the project plan, research design, and data collection plan to carry out the study. Section 3 describes the current state analysis and summarizes the strengths and weaknesses of the process. Section 4 explores the existing literature focusing on ideas to deal with the weaknesses identified in the current state analysis and outline the conceptual framework. Section 5 focuses on generating initial recommendations for improvements. Section 6 gathers feedback on the initial recommendations with the help of stakeholders to generate the final recommendations. Section 7 is for self-evaluation and conclusion of the study.

The study does not include the implementation plan for these improvements due to time constraints. It is limited to analyzing the current process and recommending improvements for the said process. The next section describes the project plan, research approach, and data collection for the study.

2 Method and Material

This section describes the selected research approach, data collection, and analysis methods that were used to carry out the study. It includes a research design that describes different stages of this study.

2.1 Research Approach

Different approaches exist to carry out quality research depending on the purpose and context of the challenges being addressed. For some research projects, the needed purpose could be just to examine the impact of something within an individual organization while some other projects may focus on multiple organizations. Saunders et al. (2016) describe all the business and management project can be put on a continuum. Basic, fundamental, or pure research which is done purely to expand knowledge of business processes, management, and their outcomes with little focus on its practical applications are at one end of the continuum and applied research that focuses on issues that have immediate relevance to managers and limited to solving specific problems in organizations are at the other end (Saunders et al., 2016: 8-9). Several research strategies can be used to carry out applied research, case study, design, and action research (Kananen 2013: 27-28).

According to Kananen (2013) design research focuses on producing functional and practical solutions by combining development and research and it is conducted in organizations to improve operations (Kananen 2013: 20-21). Furthermore, research methods can be classified into qualitative and quantitative research methods. Qualitative methods are used where the purpose of research is to explore and understand an existing phenomenon. While the quantitative method relies on numbers and closed-ended questions, qualitative methods use words and open-ended interview questions, observations, etc. for data collection and analysis (Creswell, 2014).

Therefore, to carry out this study design research was selected as the most suitable approach as it aims to find a solution for a specific problem faced by an organization. The objective of the study is to suggest recommendations for improving a specific process in the case company and this knowledge is relevant only in the case company's context. Qualitative methods are used in the study for data collection using stakeholder interviews and documents to understand the current state and for generating improvement ideas.

2.2 Research Design

This section describes the research design of the study as illustrated in figure1 in a flow diagram. As shown in figure 1, the study is conducted in 4 main phases and the first phase starts with a current state analysis of the testing process to understand the existing state of the process. The current state analysis was done by reviewing existing quality management system documents, regulatory guidelines, and existing process descriptions. Then interviews were conducted with key stakeholders and members of the testing team to identify the strength and weaknesses of the process (Data 1).

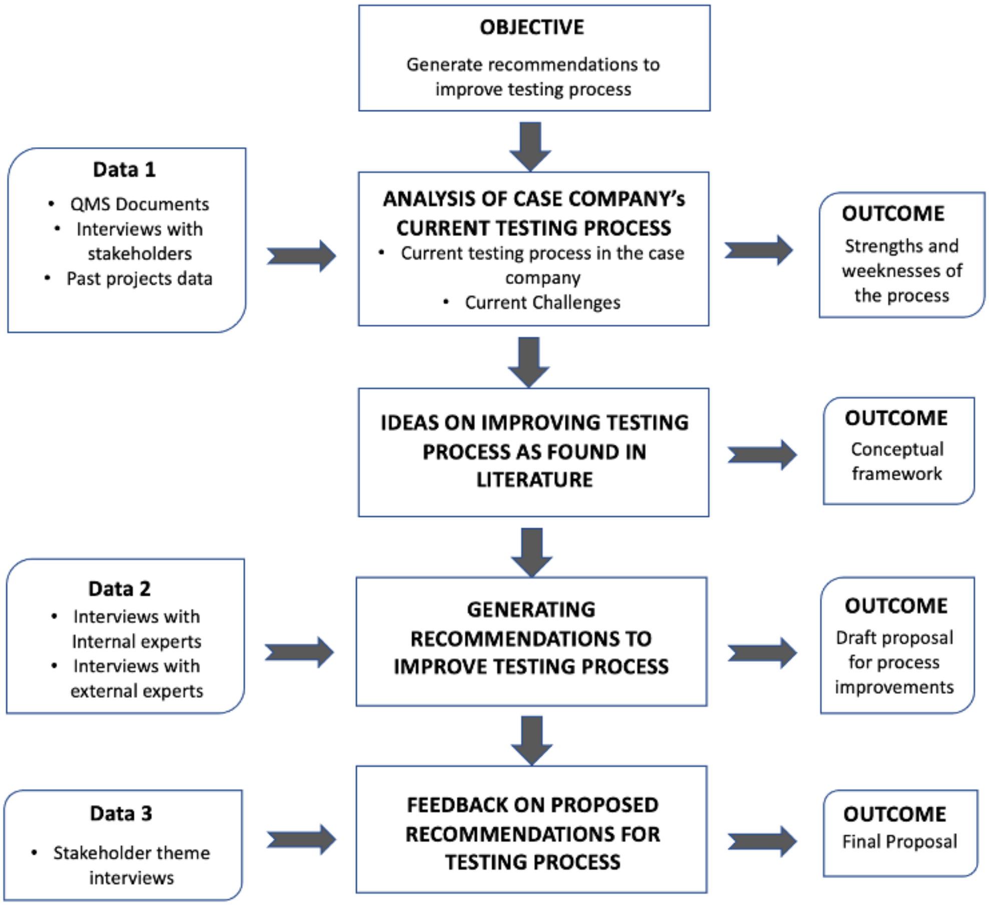

OBJECTIVE
Generate recommendations to improve testing process
Data 1
• QMS Documents
• Interviews with stakeholders
• Past projects data
ANALYSIS OF CASE COMPANY's CURRENT TESTING PROCESS
• Current testing process in the case company
• Current Challenges
OUTCOME
Strengths and weeknesses of the process
IDEAS ON IMPROVING TESTING PROCESS AS FOUND IN LITERATURE
OUTCOME
Conceptual framework
Data 2
• Interviews with Internal experts
• Interviews with external experts
GENERATING RECOMMENDATIONS TO IMPROVE TESTING PROCESS
OUTCOME
Draft proposal for process improvements
Data 3
• Stakeholder theme interviews
FEEDBACK ON PROPOSED RECOMMENDATIONS FOR TESTING PROCESS
OUTCOME
Final Proposal

Figure 1. Research design

The second phase is to review existing literature to gain knowledge of available best practices focusing on the weaknesses identified in the current state analysis of the process resulting in the conceptual framework for this study.

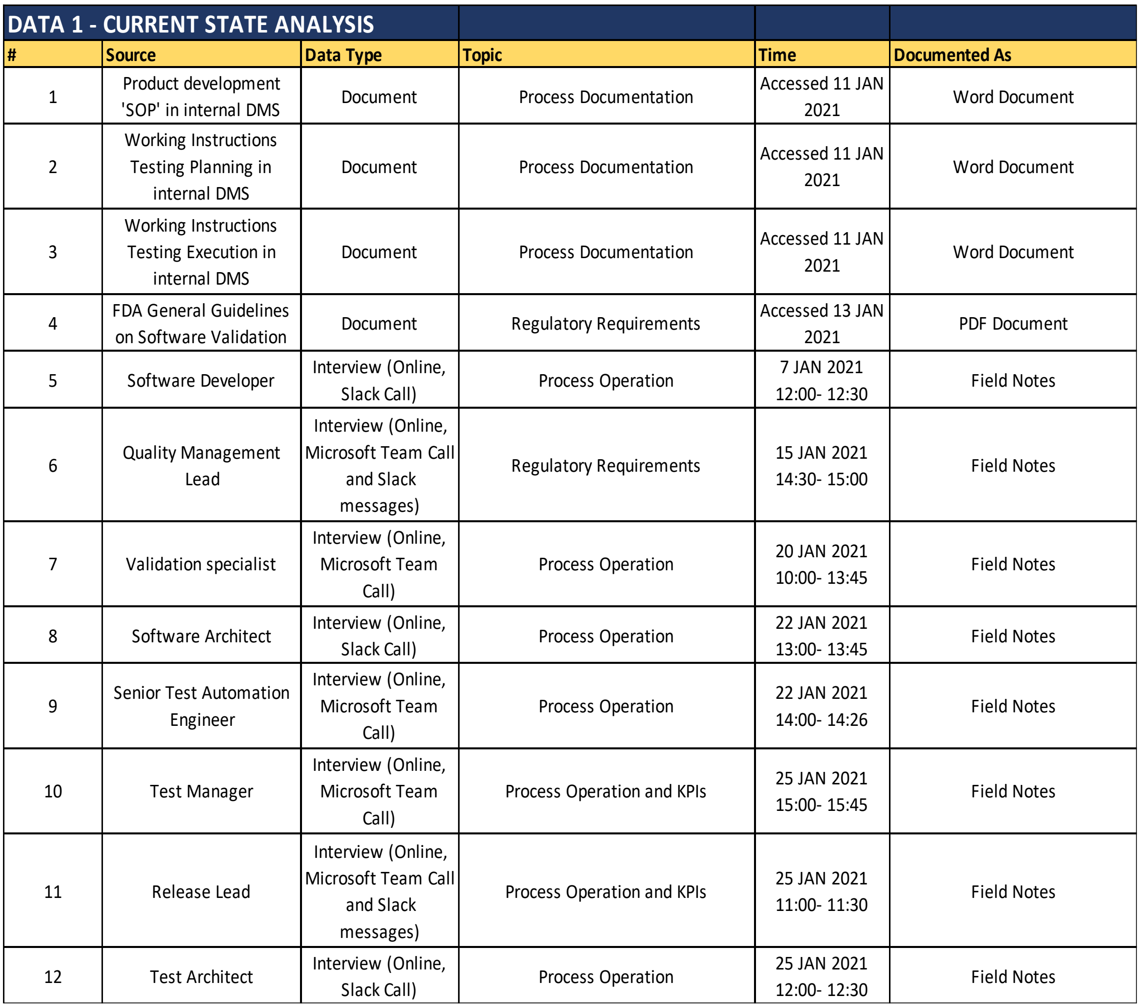
Stage 3 was carried out to generate recommendations based on the conceptual framework and with the help of internal and external experts (DATA 2). Interviews were conducted to generate the initial set of recommendations and the outcome was the draft proposal of process improvements.

The last stage as seen in Figure1 was the validations of the generated recommendations with the help of management (DATA 3). Based on the feedback final recommendations were formulated.

2.3 Data Collection and Analysis

Data for this study were collected from a variety of sources such as process documentation, regulatory guidelines, and organized interviews with the stakeholders. Data was collected in three rounds which are presented in the following three tables.

Table 1. Data 1 collection

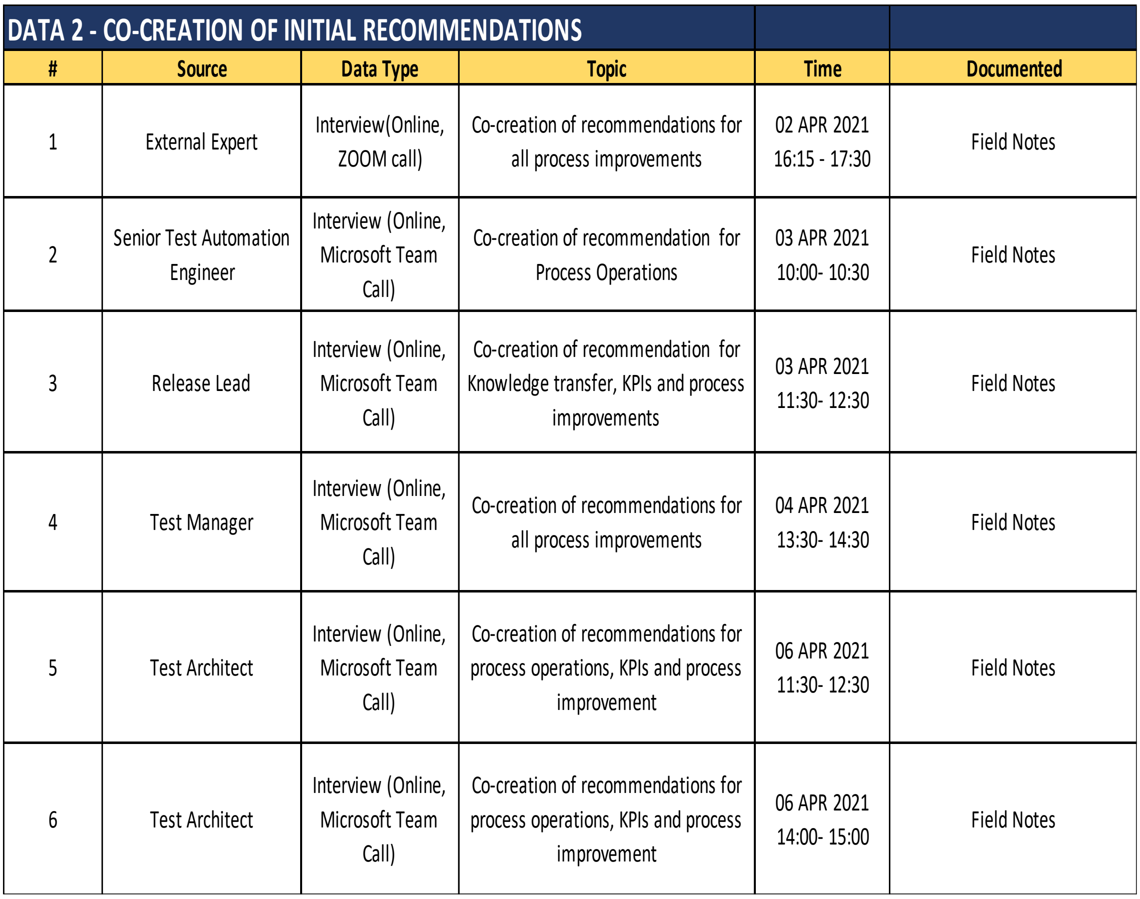


DATA 1 - CURRENT STATE ANALYSIS					
#	Source	Data Type	Topic	Time	Documented As
1	Product development 'SOP' in internal DMS	Document	Process Documentation	Accessed 11 JAN 2021	Word Document
2	Working Instructions Testing Planning in internal DMS	Document	Process Documentation	Accessed 11 JAN 2021	Word Document
3	Working Instructions Testing Execution in internal DMS	Document	Process Documentation	Accessed 11 JAN 2021	Word Document
4	FDA General Guidelines on Software Validation	Document	Regulatory Requirements	Accessed 13 JAN 2021	PDF Document
5	Software Developer	Interview (Online, Slack Call)	Process Operation	7 JAN 2021 12:00- 12:30	Field Notes
6	Quality Management Lead	Interview (Online, Microsoft Team Call and Slack messages)	Regulatory Requirements	15 JAN 2021 14:30- 15:00	Field Notes
7	Validation specialist	Interview (Online, Microsoft Team Call)	Process Operation	20 JAN 2021 10:00- 13:45	Field Notes
8	Software Architect	Interview (Online, Slack Call)	Process Operation	22 JAN 2021 13:00- 13:45	Field Notes
9	Senior Test Automation Engineer	Interview (Online, Microsoft Team Call)	Process Operation	22 JAN 2021 14:00- 14:26	Field Notes
10	Test Manager	Interview (Online, Microsoft Team Call)	Process Operation and KPIs	25 JAN 2021 15:00- 15:45	Field Notes
11	Release Lead	Interview (Online, Microsoft Team Call and Slack messages)	Process Operation and KPIs	25 JAN 2021 11:00- 11:30	Field Notes
12	Test Architect	Interview (Online, Slack Call)	Process Operation	25 JAN 2021 12:00- 12:30	Field Notes

Table 1 shows the data sources for round one which includes various product documentation available in internal document management systems, regulatory guidelines, and interviews with the key people involved in the process. Interviews were the primary source of data at this stage and were conducted in a semi-structured fashion. Due to COVID 19 restrictions, all the interviews were carried out online using Microsoft Teams and SLACK calls and the responses were recorded in the field notes. The questions for the interviews can be found in Appendix 1.

The second round of data collection is presented in Table 2 below, Data 2 was collected from the interviews with internal and external experts which were held online using TEAMS and SLACK calls to create the initial set of recommendations for the process improvement.

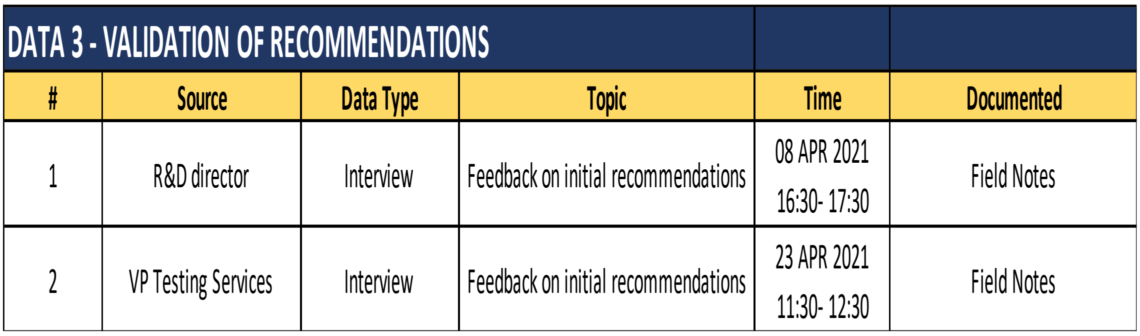
Table 2. Data 2 Collection for Initial Recommendations



DATA 2 - CO-CREATION OF INITIAL RECOMMENDATIONS					
#	**Source**	**Data Type**	**Topic**	**Time**	**Documented**
1	External Expert	Interview(Online, ZOOM call)	Co-creation of recommendations for all process improvements	02 APR 2021 16:15 - 17:30	Field Notes
2	Senior Test Automation Engineer	Interview (Online, Microsoft Team Call)	Co-creation of recommendation for Process Operations	03 APR 2021 10:00- 10:30	Field Notes
3	Release Lead	Interview (Online, Microsoft Team Call)	Co-creation of recommendation for Knowledge transfer, KPIs and process improvements	03 APR 2021 11:30- 12:30	Field Notes
4	Test Manager	Interview (Online, Microsoft Team Call)	Co-creation of recommendations for all process improvements	04 APR 2021 13:30- 14:30	Field Notes
5	Test Architect	Interview (Online, Microsoft Team Call)	Co-creation of recommendations for process operations, KPIs and process improvement	06 APR 2021 11:30- 12:30	Field Notes
6	Test Architect	Interview (Online, Microsoft Team Call)	Co-creation of recommendations for process operations, KPIs and process improvement	06 APR 2021 14:00- 15:00	Field Notes

The last set of data collected was Data 3 as shown in Table 3 to validate the initial set of recommendations with the help of interviews with senior management and was conducted online using the same tools.

Table 3. Data 3 Validation of Initial Recommendations



DATA 3 - VALIDATION OF RECOMMENDATIONS					
#	Source	Data Type	Topic	Time	Documented
1	R&D director	Interview	Feedback on initial recommendations	08 APR 2021 16:30- 17:30	Field Notes
2	VP Testing Services	Interview	Feedback on initial recommendations	23 APR 2021 11:30- 12:30	Field Notes

The next section of the study presents the findings of Data 1 collection as the current state analysis of the testing process in the case company.

3 Current State Analysis

This section describes the current state of the testing process in the case company and discusses the findings of the analysis as a summary of the strength and weaknesses of the process. Data collection methods to carry out the analysis was introduced in the previous section.

This section has three subsections starting with an overview of the current state analysis, a detailed description of the testing process and at the end, a summary of identified strengths and weaknesses are included.

3.1 Overview of the Current State Analysis Stage

The current state analysis was carried out in three different stages.

First, existing process documents from internal quality management systems and regulatory requirements were studied to formulate a basic understanding of the process and key stakeholders were identified for the second stage.

In the second stage, interviews were conducted with the identified stakeholders to better understand the current working of the process. The interviewees were from the quality, Sprint, and Validation teams. During the interviews, interviewees were asked to (a) describe their role in the testing process, (b) what is the biggest challenge they see which blocks them from doing their work efficiently, (c) identify strengths and weaknesses of the process. Based on that a process map was developed describing the current flow of the testing process.

In the third stage data collected during the interviews was analyzed and a summary of key strengths and weaknesses was created. The strength and weaknesses were highlighted in the process map for better visualization.

3.2 Description of the Current Testing Process

The process flow of the testing process in the case company is presented in Figure 2 which was redrawn from various figures available in the process documentation. A detailed description of the various stages of the process is provided in the following sub-sections.

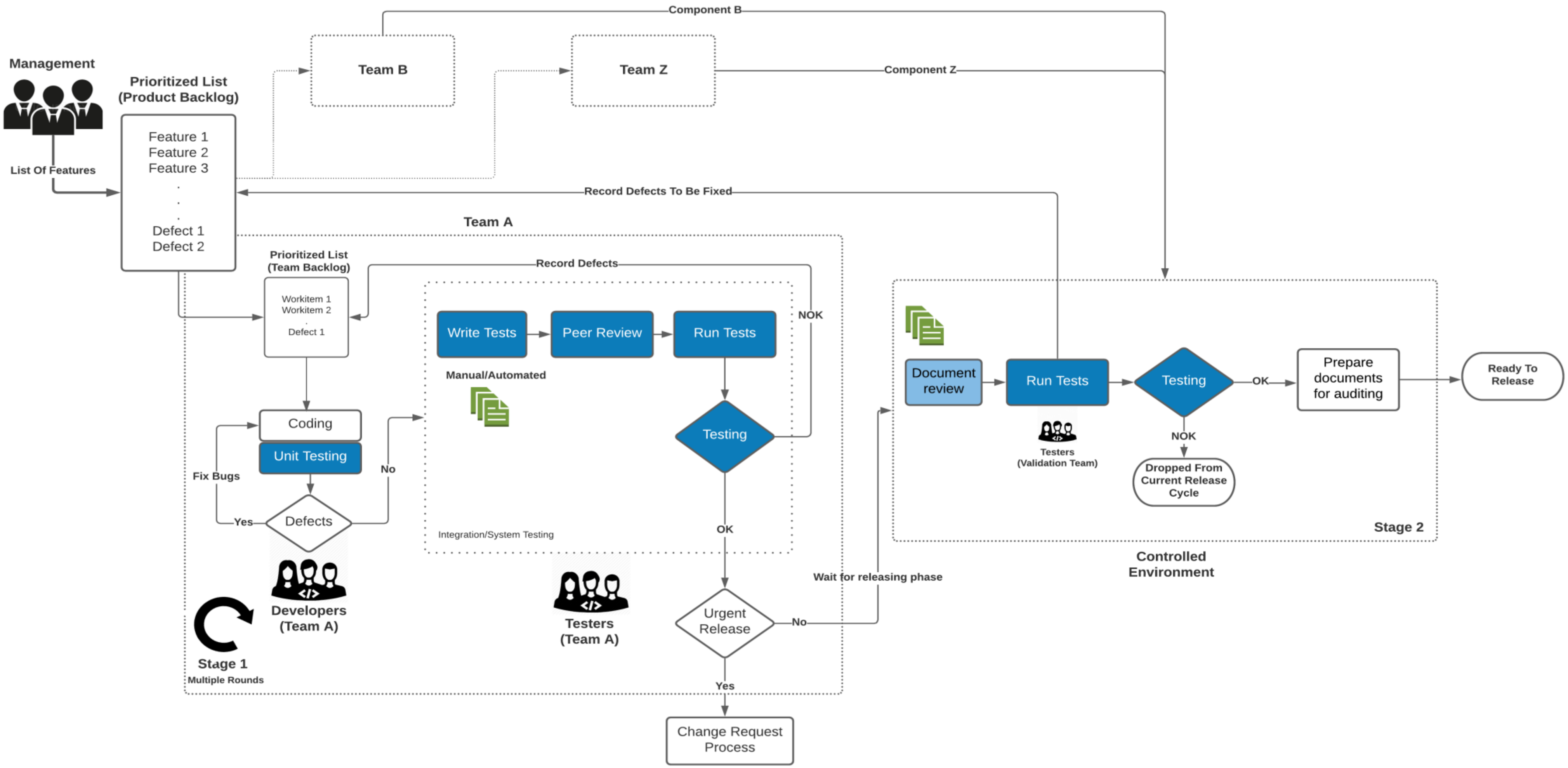


Figure 2. The testing process flow diagram.

As shown in figure 2, the process flow starts with the management team creating a list called product backlog that contains a list of features that needs to be developed in the next 8-12 weeks called a program increment. The product backlog is a prioritized list that also contains all the previously known issues that need to be fixed in this increment. During the increment planning, different teams commit to delivering these features and these items are put into the team's backlog. Some of these features are delivered by multiple teams while some features are independent.

After the increment planning is done, different teams do their planning for the next 2-4 weeks called one SPRINT and these teams are called SPRINT teams. The case company follows a 2-week SPRINT model. So, A program increment has multiple SPRINTS. The product owner is responsible for prioritizing the team's backlog and delivery of the promised feature during the program increment.

3.2.1 Testing in SPRINT teams (Stage 1)

Once the team backlog for the current increment is finalized, the team's do the SPRINT planning in which items for the next two weeks are planned. The actual implementation of the new features then starts by the developers of the team.

After the code is complete developers do the lowest level of testing called unit testing before submitting it to test engineers in the team called SPRINT testers. SPRINT testers then write the manual and automated tests to test these new changes in the code. Test cases are then reviewed by other testers in the teams. Manual and automated testing is done at this stage depending on the component being tested but usually, it involves both.

The issues which are found at this stage are recorded by the team members. These issues are then prioritized with the product owner and pushed into the team's backlog. Depending on the severity and criticality of these issues, these will be fixed either in the future SPRINT or in the same SPRINT if there is time left. Very low priority issues will stay in the team's backlog so that teams can focus on more important work and deliver as promised.

One of the major weaknesses highlighted by the SPRINT team testers was that testing is slow and testers never find time to do the exploratory testing and work on the test automation improvements. One of the respondents expressed this as follows:

> *We don't have time to do exploratory testing and focus on our test automation improvements.*
>
> *Data 1: Senior test automation engineer*

After testing is done by the SPRINT team testers that change is marked as done. Sometimes there might be an issue in the previous release of the software component that needs immediate attention, then those are pulled into the SPRINT. All the urgent changes and fixes release through a separate process the change management process. The team then continues to plan for the next SPRINT and work on the remaining items in the team backlog. All the release dates are agreed upon during the program increment planning and on the release date, all the approved features in the SPRINT teams move to the next stage of testing that is validation testing.

3.2.2 Testing in Validation teams (Stage 2)

The second stage in the testing process is called the validation phase. Before any component can be released to production it must go through the validation phase. New features that are developed by different SPRINT teams for different components will now be tested by another team of testers called validation testers. The environment for this testing is controlled and no development happens in this environment. The validation team is responsible for testing all the features that come from different teams for different software products. During the validation phase, the testers from the team run the same tests that were developed by the SPRINT team testers during phase 1.

The issues found during this stage are reported as defects and those go back to product backlog. If some critical issues are found and cannot be fixed in the current release cycle those features are dropped from the current release cycle. These dropped features those go to production in the next release cycle or through the change management process as shown in Figure 2.

The validation testers often face challenges during execution and need help from the SPRINT testers to run these tests which leads to a lot of waiting in the process. Also, the test automation is utilized only by a few SPRINT teams so there is a lot of manual testing involved. One of the interviewees highlighted:

There is no formal knowledge transfer happening between the two teams

DATA 1: Validation specialist.

3.3 Summary of findings from the Current State Analysis (Data Collection 1)

This section provides an overview of the strengths and weaknesses identified during the current state analysis of the testing process in the case company. The current state analysis was based on data mentioned in Table 1.

Based on the data collected a total of three strengths and seven weaknesses were identified in the testing process. The strengths are listed in Table 4.

Table 4. Strengths of the current testing process

Summary of strengths		
#	**Strength**	**Source**
1	Process is well defined and documented	QMS and Interviews
2	Process is effective in maintaining quality of software products	Stakeholder interviews
3	Process has clearly defined roles	Interviews

Internal documents and interviews with key stakeholders highlighted that the process is very well defined and documented which was not surprising as the company is operating in a highly regulated environment. Also, the interviews highlighted that there are clearly defined roles and at each stage of the process.

The biggest strength that was highlighted by multiple stakeholders was that the process is quite effective in making sure that the quality of the developed products is high.

> *I would say that the process strength is that it protects the quality of the products.*

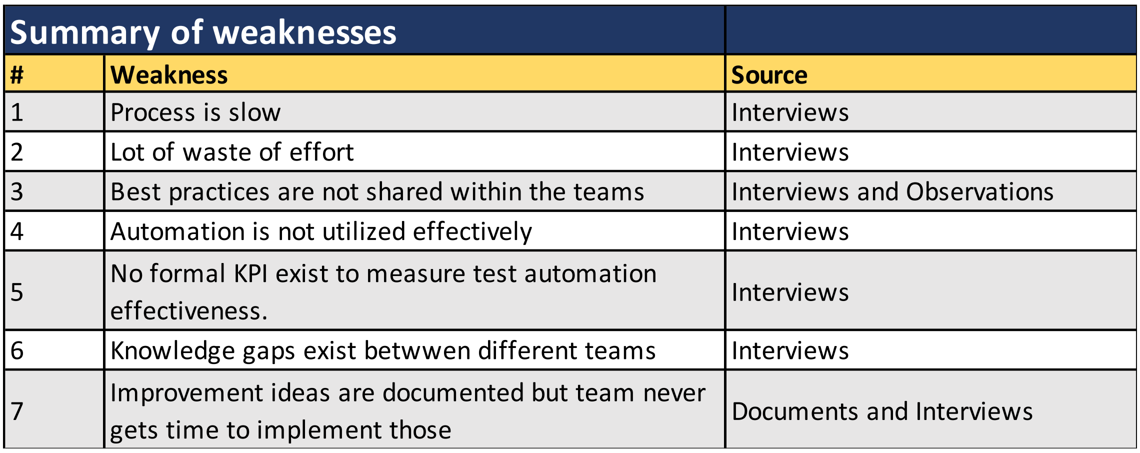
Data 1: Release lead

> *Process is quite clear, and it works as it is supposed to.*

Data 1: Test manager

Despite being an effective process, several weaknesses were also identified as shown in Table 5 below.

Table 5. Weaknesses of the testing process



Summary of weaknesses		
#	Weakness	Source
1	Process is slow	Interviews
2	Lot of waste of effort	Interviews
3	Best practices are not shared within the teams	Interviews and Observations
4	Automation is not utilized effectively	Interviews
5	No formal KPI exist to measure test automation effectiveness.	Interviews
6	Knowledge gaps exist betwwen different teams	Interviews
7	Improvement ideas are documented but team never gets time to implement those	Documents and Interviews

Many stakeholder interviews highlighted that the process is a little slow and there is a lot of waste of effort and duplicate work especially for projects where test automation is high.

> *We often need help from SPRINT testers for test execution so there is a lot of waiting involved and it gets worse when people are on vacation.*
>
> *DATA 1: Validation specialist.*

Another set of weakness was related to the test automation which only a few teams utilize but the way it is utilized can be improved and there are no existing KPIs to measure its effectiveness.

> *For many projects, test automation doesn't exist one of the reasons being some teams have more skilled people than others. Test automation should act as quality gates for automatic decision-making.*
>
> *DATA 1: Test architect*

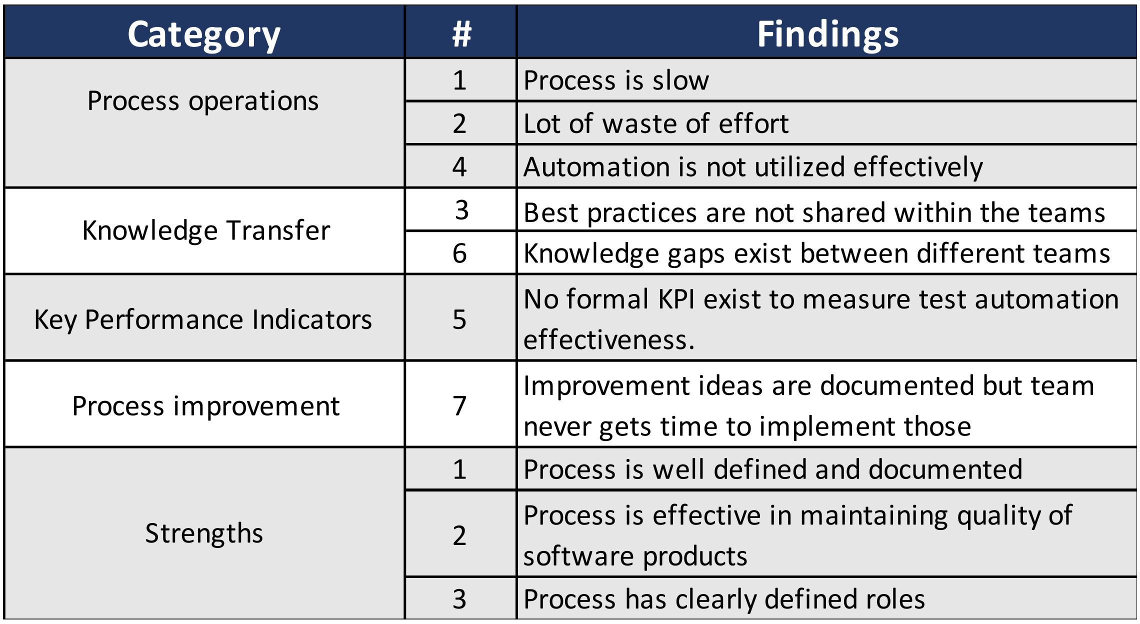
> *We don't have any formal KPI to track test automation performance.*
>
> *DATA 1: Test manager*

Also, there is a knowledge gap that exists between the two teams which leads to more delays and SPRINT testers sometimes must jump to help with the testing. One of the common issues that was highlighted across teams was that improvement ideas are discussed and documented but no practical steps are taken and often the team does not even get time to implement those.

3.4 Key Focus Areas

Three strengths and seven weaknesses were identified during current state analysis which are further divided into categories as shown in Table 6.

Table 6. Findings divided into categories for literature review



Category	#	Findings
Process operations	1	Process is slow
	2	Lot of waste of effort
	4	Automation is not utilized effectively
Knowledge Transfer	3	Best practices are not shared within the teams
	6	Knowledge gaps exist between different teams
Key Performance Indicators	5	No formal KPI exist to measure test automation effectiveness.
Process improvement	7	Improvement ideas are documented but team never gets time to implement those
Strengths	1	Process is well defined and documented
	2	Process is effective in maintaining quality of software products
	3	Process has clearly defined roles

The first four categories represent the identified weaknesses in the current process which will be the focus area for the next stage of literature review and the last category of strengths is preserved to create the initial recommendations.

The first category is related to weaknesses in process operation and has the highest number of weaknesses. The second category includes weaknesses in the knowledge transfer category where best practices are not shared between different teams and knowledge gaps exist in the validation team. The third category highlights the missing key performance indicators for test automation and the last category includes weaknesses in the process improvement area.

All the findings of the current state analysis are marked with circled numbers in the process diagrams in Figure 3 below at stages where those were identified. Some weaknesses are general weaknesses in the process not occurring at a specific stage so those are marked separately. The Numbers refer to Table 4 and Table 5 where the green numbers represent strengths and red numbers the weaknesses identified.

In section 4 existing knowledge in literature is searched to get improvement ideas. The Literature review is focused on identified weakness categories and target improvement ideas specifically for the weaknesses identified in the current process.

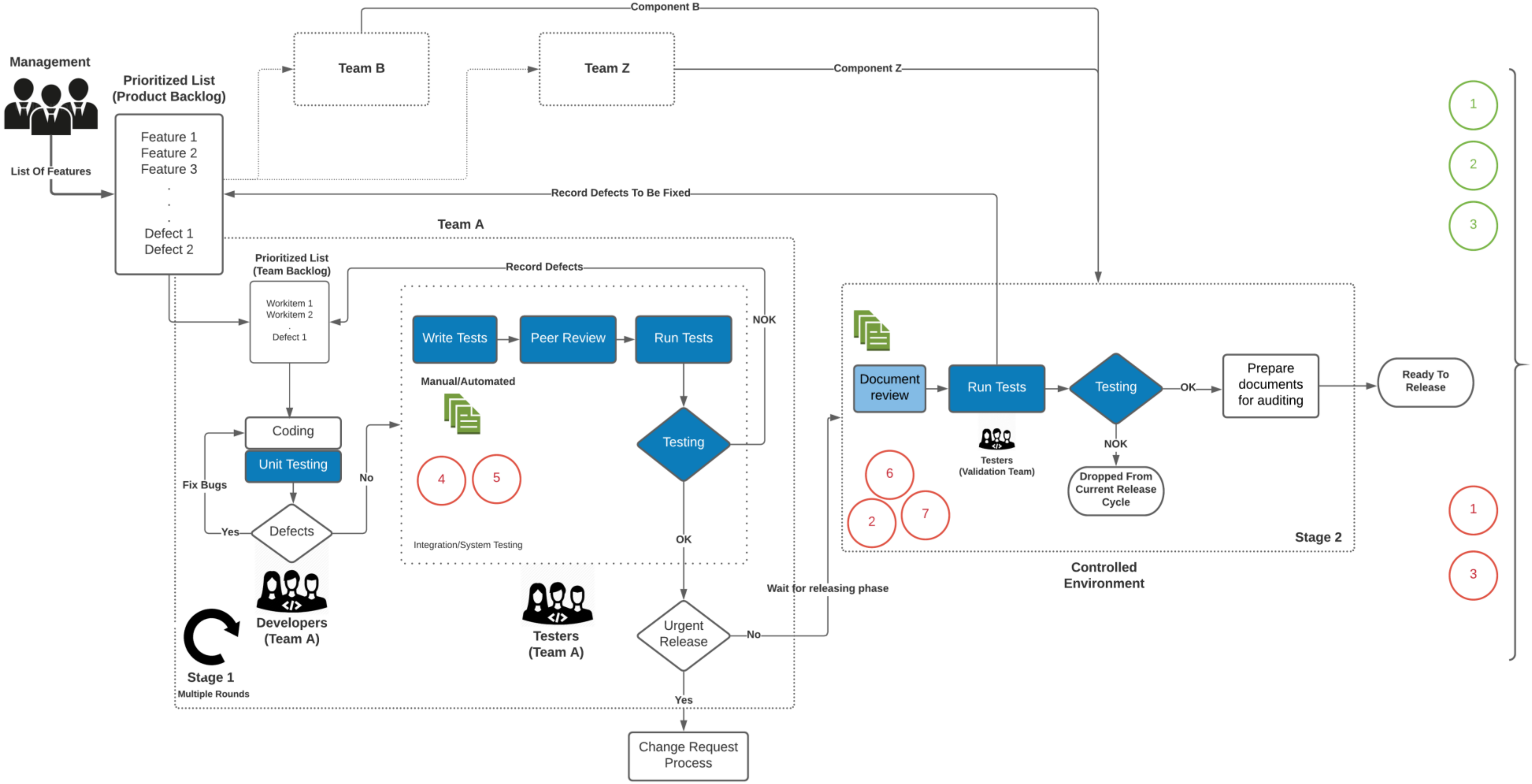


Figure 3. Identified Strengths and Weaknesses in the Process Flow.

4 Existing Knowledge and Best Practices on Test Process Improvement

Section 4 discusses the existing knowledge and practices for test process improvement found in the relevant literature and puts it into a conceptual framework. Relevant literature was referred to based on the current state analysis of the test process done in Section 3. The four categories of identified weaknesses were the basis of a thorough literature search to obtain ideas on tackling these weaknesses.

This section is divided into four sub-sections. Each section starts with the description of the relevant study, idea, or model, followed by a discussion and rationale behind the selection. Finally, the key findings are summarized into a visual format, the conceptual framework for the study.

4.1 Lean Process Operations

According to Staats and Upton (2011), the “Toyota” or Lean principles can be useful in operations involving judgment and expertise. They claim that there are always certain activities in knowledge operations that have nothing to do with applying judgment and expertise and can significantly benefit from lean principles. May (2005) emphasizes that focusing on value and continuous improvement are two important lean principles that have the most relevance in knowledge work.

Many people underestimate the amount of waste that can be removed from their daily work and organizations should focus on systematically making the waste visible and teaching people how to eradicate that. They also suggest that teaching everyone “The five why’s “is an effective way to get to the root cause of every activity that is performed (Staats and Upton, 2011: 4).

Poppendieck (2002) suggests that four principles of lean thinking which are most relevant in software development are:

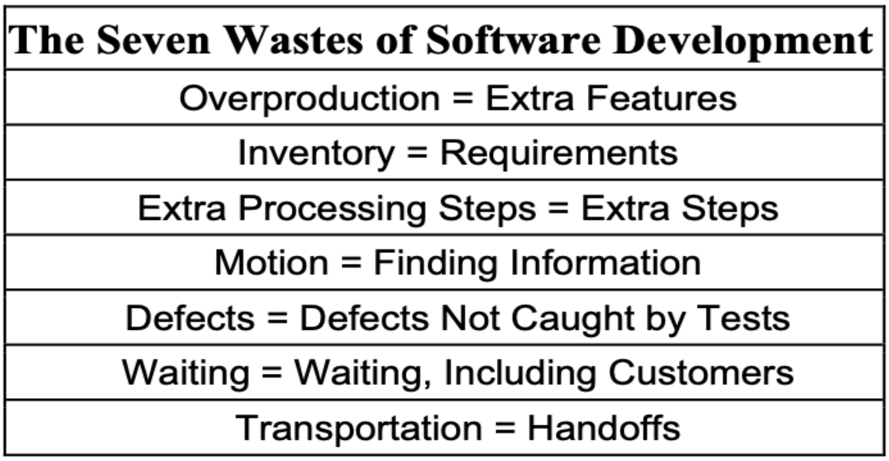
1. Eliminate waste to maximize value creation

2. Focus on people who add value

3. Delay Commitment

4. Optimize at all levels

Poppendieck also emphasizes that to utilize the lean principles the first step is to learn how to identify waste which is largely invisible in knowledge work (May, 2005). Seven types of waste exist in manufacturing (Ohno, 1988) and how different types of waste would translate from manufacturing to software development (Poppendieck, 2002) is shown in figure 4 below.



The Seven Wastes of Software Development
Overproduction = Extra Features
Inventory = Requirements
Extra Processing Steps = Extra Steps
Motion = Finding Information
Defects = Defects Not Caught by Tests
Waiting = Waiting, Including Customers
Transportation = Handoffs

Figure 4. Seven types of waste in software development

Nidagundi and Novickis (2017) went one step ahead and investigated sources of waste specifically in Scrum Software testing as described below:

- **Transport:** Unnecessary testing and inefficient scrum meetings.
- **Inventory:** Not able to finish testing in the same sprint and testing is slipped into next sprint.
- **Motion:** Incorrect or unnecessary task estimations.
- **Waiting:** Testers waiting for code to be ready and other team members due to process dependencies.
- **Overproduction:** Task completions before the estimated time and waiting for another developer to finish before continuing with the next stage.

- **Over Processing:** Being too strict with scrum rules and inappropriate scrum ideology.

- **Defects:** Missed defects and errors due to wrong interpretation of requirements by testers.

Test Pyramid

According to Crispin (2008) test automation is the key to be successful in agile software development. Test automation not only acts as a safety net but also gives feedback early and often, but test automation requires thoughtful investment and strategy. Cohn (2009) claims that one of the reasons why teams struggle with writing fast tests was that teams automate at the wrong levels. Effective test automation requires automating at three different levels which makes the test automation pyramid as shown in figure 5.

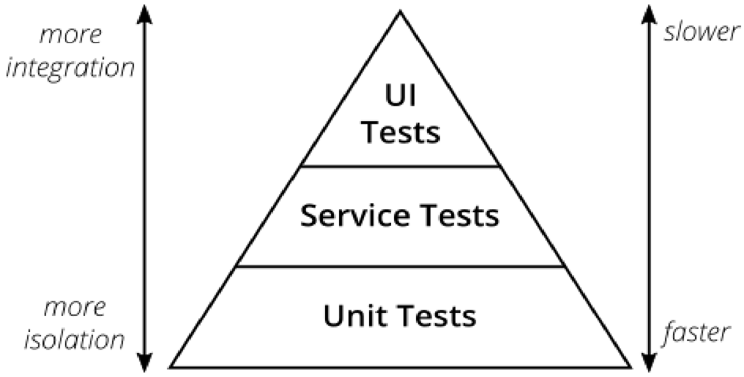


Figure 5. The Test Pyramid. Retrieved from https://martinfowler.com/articles/practical-test-pyramid.html

In the figure, each layer represents how automation effort should be divided between different types of tests. The higher the level of the tests the slower those tests would be which in result will increase the feedback time. Crispin (2008) later added another layer of manual testing at the top of the pyramid.

Unlike Crispin, Vocke (2018) argues that the test pyramid presented above might be overly simplistic and might not be ideal but still due to its simplicity the test pyramid acts as a rule of thumb while creating automation test suits. Two things that can be taken from the Cohn test pyramid is to write automation tests with different levels of granularity and the higher the level of testing the fewer tests you should have (Vocke,2018).

4.2 Knowledge Transfer

According to Seppänen et al. (2002) efficient creation, distribution, and reuse of up-to-date knowledge are critical success factors in software organizations. Software development is a collaborative process that is highly dependent on effective knowledge sharing (Ghobadi and Mathiassen, 2015). Ghobadi et al. (2015) conceptualized thirty-seven different types of knowledge sharing barriers in agile teams divided into the following seven categories team diversity, team perception, team capabilities, project communication, project organization, project technology, and project setting.

According to Riege (2007, 48-67), effective knowledge transfer is more than the movement of knowledge from one location to another. For any knowledge-sharing initiatives to be effective, it must be introduced by senior and middle-level managers who not only understand the need to align knowledge management and business strategy but also understand the barriers. Riege (2007) in his research created some guidelines on how to overcome the barriers in knowledge transfer. Some of those are summarized below:

- Acknowledge time pressure and allocate free time for knowledge transfer for example an hour per week.
- Provide formal sharing settings.
- Offer social gatherings and occasions for people to communicate.
- Recognize people for their contributions.
- Ask people to be proactive and hold regular sessions.
- Provide suitable training programs.
- Make sharing knowledge part of the culture and people’s individual KPIs.

- Identify who needs to share what, when, why, when, and to whom.

Sarka (2014) in his study identified six criteria for efficient knowledge transfer which are summarized below:

1. Customization of messages for a different audience to build trust.

2. Quick and genuine feedback on the communicated message.

3. Involvement of more senses while transferring knowledge especially hearing and vision.

4. Spontaneity in communication.

5. Balancing the dialogue with the involvement of other parties.

6. Making the message more interesting by including humor, stories, and personal context.

4.3 Key Performance Indicators

Measurement has always been a central subject in any engineering discipline and software testing is no different. Software measurement plays a key role in an effective testing process (Farooq et al., 2014).

Neely (2000) suggests below mentioned characteristics of effective KPI’s:

- KPI’s must be aligned with business goals.

- The purpose of KPI must be clear.

- Data collection methods and calculating performance must be clear.

- KPI’s should be selected through discussions with the stakeholders.

- KPI’s should change with the circumstances.

- KPI’s should provide a way to continuous improvement rather than just monitor.

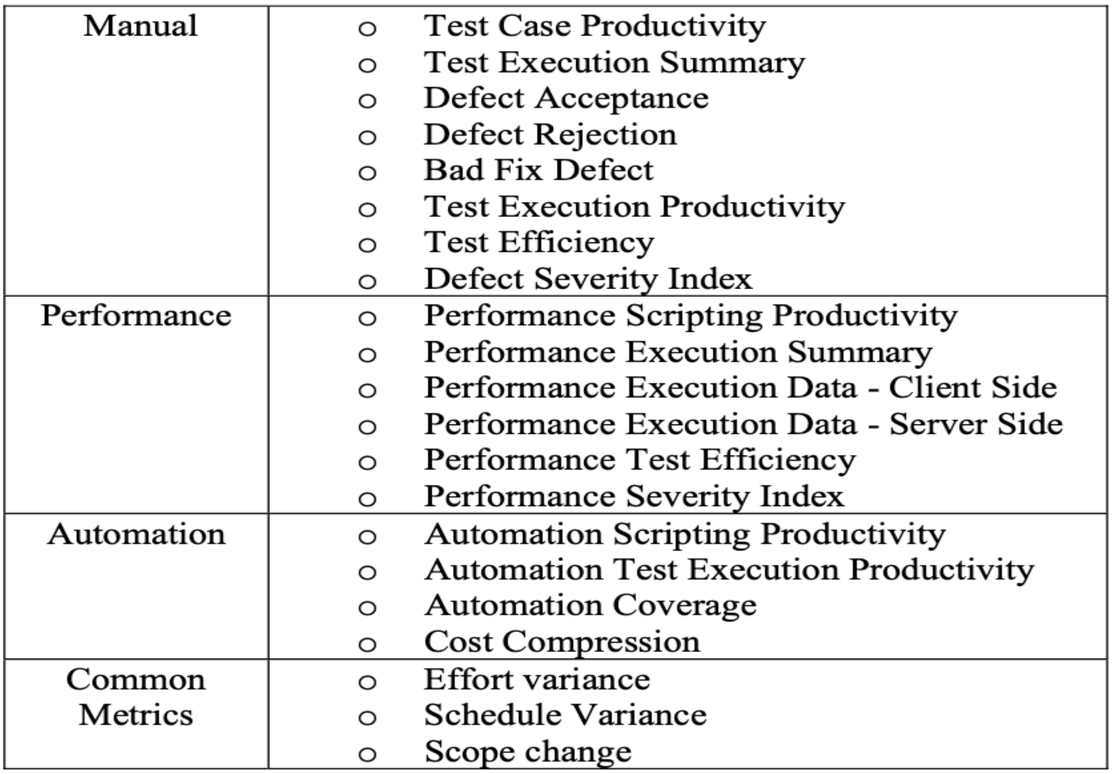
According to Munson (2004) evaluating test activities will give you great insight into the effectiveness of the test process. Different metrics trends can help in identifying when a process is going out of control and acts as a monitor for the process (Nirpal, 2011). Farooq et al. (2014) claim that effective implementation of metrics can help deliver the software on time and on budget. There are two categories of software metrics that are mainly concerned with testing activities namely test process metrics and test product metrics.

Test Process metrics are used to monitor testing phase progress. These metrics do not provide any information about the quality of testing but rather the effectiveness and quality of the testing process. Some examples of test process metrics include the number of test cases executed, total execution time to run the test cases, the average time it takes to execute a single test case, percentage of test cases that were passed or failed, etc.

Test product metrics on the other hand provide information about the testing status of a software product and are generated by test execution. Using these metrics helps in measuring the quality of the testing being performed which is useful information for releasing decisions. Some examples of the test product metrics are what is the average time interval between failures, the average number of failures experienced in time intervals.

Nirpal et al. (2011) categorized testing metrics based on the type of testing performed i.e., manual test metrics, automation test metrics, and performance test metrics. Table 7 presents some of the widely used test metrics in the software industry and categorizes them according to the type of testing that is being performed.

Table 7. Software testing metrics (Nirpal et al., 2011)



Manual	○ Test Case Productivity ○ Test Execution Summary ○ Defect Acceptance ○ Defect Rejection ○ Bad Fix Defect ○ Test Execution Productivity ○ Test Efficiency ○ Defect Severity Index
Performance	○ Performance Scripting Productivity ○ Performance Execution Summary ○ Performance Execution Data - Client Side ○ Performance Execution Data - Server Side ○ Performance Test Efficiency ○ Performance Severity Index
Automation	○ Automation Scripting Productivity ○ Automation Test Execution Productivity ○ Automation Coverage ○ Cost Compression
Common Metrics	○ Effort variance ○ Schedule Variance ○ Scope change

4.4 Process Improvement

According to Veenendaal (2013), process improvement techniques such as the Deming improvement cycle are as relevant to the testing process as any other process. The idea or philosophy behind the concept is to continuously look for areas of improvement in the process.

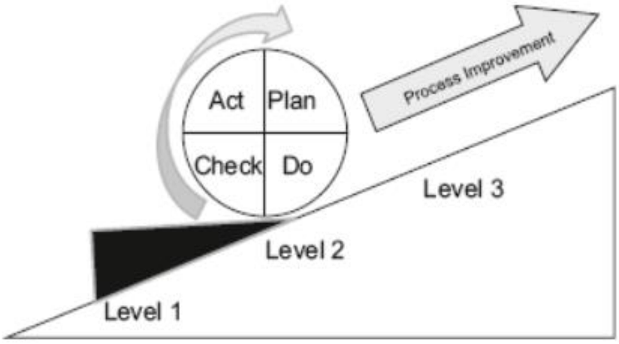


Figure 6. The Plan-Do-Check-Act Cycle (PDCA)

4.4.1 Plan-Do-Check-Act and IDEAL framework

Figure 6 above shows the Deming cycle or PDCA cycle as a four-step iterative process:

Plan (P): The cycle starts with the planning stage where management formulates initial improvement goals for the testing process after analyzing the current situation. These goals are then broken down into series of actions called “control points”. Before implementing the change, it is important to understand how the test process failed to meet the expectation.

Do (D): This stage is the implementation stage of the cycle where actions are taken and measured. It is important to involve people who are affected by the change for it to be effective.

Check (C): The next stage is to check or analyze the data collected to understand how the plan was implemented and if the objective of the improvement was met.

Act (A): Using the data collected during the check stage areas of improvement are identified and prioritized and move to the next PDCA cycle. It’s important to make sure the lesson learned during previous cycles are noted to avoid their recurrence.

The IDEAL Improvement Framework

McFeeley/SEI (1996) developed the IDEAL framework which is an extension of the Deming cycle specifically designed for the software process improvement named after the five phases of the framework. The IDEAL model provides a practical reference when implementing test process improvements (Veenendaal et al., 2013).

Figure 7 shows the five phases of the IDEAL framework which consists of fourteen activities in total:

Initiate: Initiate the process improvement by setting goals and establishing metrics.

Diagnose: Diagnose the current state of the process and desired state of the process and develop recommendations.

Establish: Prioritize improvements and prepare a plan for the actions that need to be taken.

Action: Create, test, refine and implement the solution.

Learn: Analyze the results and plan future actions.

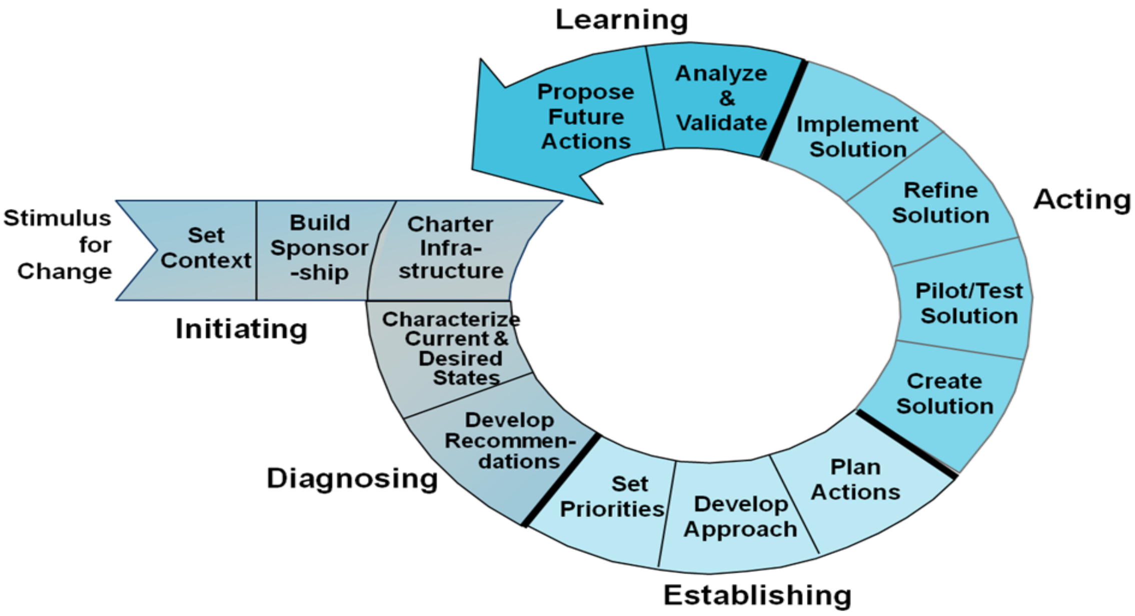


Figure 7. IDEAL Improvement Framework. Retrieved from https://www.plays-in-business.com/ideal-initiating-diagnosing-establishing-acting-learning/

4.4.2 Defining roles and responsibilities with RACI Matrix

As discussed in previous sub-sections there are not only opportunities but need for people to be engaged in any improvement initiative. Without clarity on who is going to be responsible, it is almost impossible for any improvement project to be successful. Role clarity can easily be compromised particularly within cross-functional teams (Wong et al. 2007).

Smith and Erwin (2005) introduced a simple technique called responsibility charting to resolve role ambiguities through a cross-functional collaborative effort often referred to as the RACI matrix. Responsibility charting helps managers from different organizational levels to handle process-related actions that must be accomplished. Responsibility charting ensures that accountability is in place and accountabilities are moved down to the most appropriate level (Smith et al. 2005). RACI matrix is especially useful in cross-functional processes (ITIL, 2011).

> *Trying to get work done without clearly establishing roles and responsibilities, is like trying to parallel park with one eye closed.*
>
> *Smith et al. 2005*

Responsible: The individual(s) who will do the task. There can be multiple people responsible for a task.

Accountable: The Person who is ultimately answerable for the activity. There should only be one person accountable for one action.

Consulted: Typically, subject matter experts to be consulted before final action.

Informed: The person who needs to be informed while activity is occurring or after it is complete but not required to be part of the process (Smith et al., 2005).

It is important to understand that responsibility does not end with just creating the chart, it must be an ongoing activity as responsibilities change over time.

4.4.3 Task prioritization with MoSCoW Technique

According to Sarah Hatton (2008), MoSCoW technique is one of the easiest and fastest prioritization methods available. Vestola (2010) claims that techniques might be better when prioritizing a medium to large number of items (Vestola, 2010: 3).

Agile Business Consortium describes MoSCow as a prioritization technique that can be applied to requirements, tasks, tests, etc. where letters stand for:

Must Have: No point in delivering without these.

Should Have: Painful to leave these but still viable.

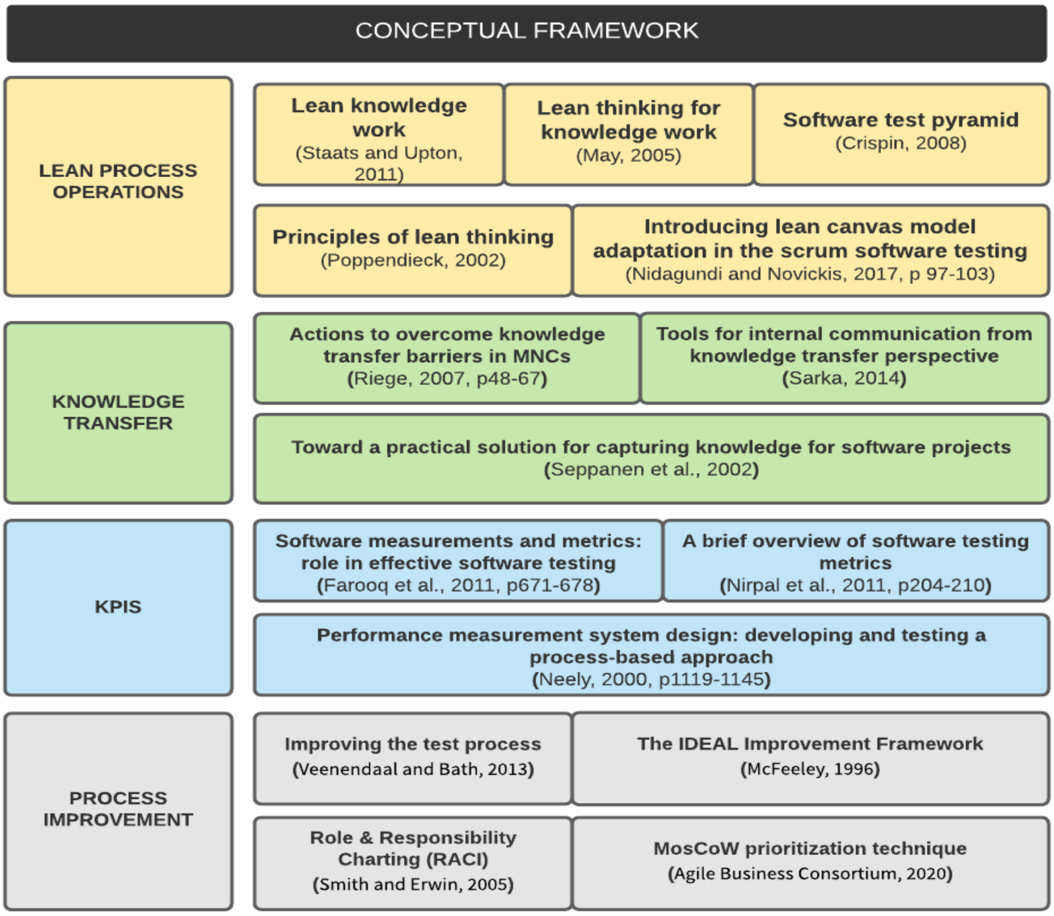
Could Have: Wanted but less important.

Won’t Have: Won’t be delivered this time (Agile Business Consortium, 2020).

4.5 Conceptual Framework of This Thesis

The Literature review provided several good ideas to tackle the business challenge and weaknesses revealed by the current state analysis in section 3 of the study. The key themes from the literature review are presented in figure 8 Conceptual framework for this study.

As shown in figure 8 the conceptual framework is divided into four categories that were identified through the current state analysis of the process. Process operation category includes instructions, best practices to utilize lean principles in knowledge-based organizations, and how test automation should be utilized in agile projects. The process improvement category includes two process improvement frameworks, responsibility charting, and prioritization techniques. The Key Performance Indicators (KPI) category includes types of metrics widely available in software testing and how KPIs should be selected and utilized. The last category is about understanding barriers in knowledge transfer and guidelines for overcoming those barriers.



CONCEPTUAL FRAMEWORK

LEAN PROCESS OPERATIONS
- Lean knowledge work (Staats and Upton, 2011)
- Lean thinking for knowledge work (May, 2005)
- Software test pyramid (Crispin, 2008)
- Principles of lean thinking (Poppendieck, 2002)
- Introducing lean canvas model adaptation in the scrum software testing (Nidagundi and Novickis, 2017, p 97-103)

KNOWLEDGE TRANSFER
- Actions to overcome knowledge transfer barriers in MNCs (Riege, 2007, p48-67)
- Tools for internal communication from knowledge transfer perspective (Sarka, 2014)
- Toward a practical solution for capturing knowledge for software projects (Seppanen et al., 2002)

KPIS
- Software measurements and metrics: role in effective software testing (Farooq et al., 2011, p671-678)
- A brief overview of software testing metrics (Nirpal et al., 2011, p204-210)
- Performance measurement system design: developing and testing a process-based approach (Neely, 2000, p1119-1145)

PROCESS IMPROVEMENT
- Improving the test process (Veenendaal and Bath, 2013)
- The IDEAL Improvement Framework (McFeeley, 1996)
- Role & Responsibility Charting (RACI) (Smith and Erwin, 2005)
- MosCoW prioritization technique (Agile Business Consortium, 2020)

Figure 8. Conceptual Framework

Section 5 utilizes the conceptual framework to generate initial recommendations to improve the testing process. The ideas found through conceptual framework themes and process strengths are used to tackle the weaknesses revealed by the current state analysis of the process with the involvement of the key stakeholders in the process.

5 Creating initial recommendations for the test process improvement

Section 5 combines the results of the current state analysis and the conceptual framework towards the building of initial recommendations using Data 2 for improving the test process in the case company. This section includes an overview of the stage, findings from the data collection, and a description of the creation process.

5.1 Overview of the Recommendations Creation Stage

The initial proposal was co-created with the help of internal stakeholders and an external expert in three themed interviews. All the interviews started with a general introduction of the business problem, objective, and outcome of this study. Then all the findings from the current state analysis and relevant ideas from the literature were presented to the participants. Also, Process flow, organization structure, and industry regulations were explained to the external expert to give a better understanding of the operating environment and limitations. All the discussions in the interviews were targeted around the categories of weaknesses identified during the current state analysis of the process.

Discussion during the interviews was then steered towards developing ideas of improvements in that category. Activities included brainstorming, arguments, and discussions and in the end, participants had the opportunity to address the process improvements as a whole.

The internal stakeholders were selected from the testing organization who have visibility of multiple projects in R&D and the choice of external expert was based on the vast experience in software testing and product management. Due to the COVID outbreak, all the interviews were conducted online. Interviews with internal stakeholders were conducted through Microsoft Teams and the ZOOM application was used in the interview with the external expert. All the commentary and discussions were captured in the field notes.

Finally, initial recommendations were compiled from the field notes and categorized again following the same logic as the current state analysis. Some of the recommendations were identified to help with issues in multiple categories but each item is placed in the most relevant category for clarity. The summary of recommendations is presented in the sub-section below.

5.2 Summary of the Initial Recommendations

The initial recommendations were created based on the current state analysis of the process and conceptual framework. Initial recommendations are also divided into the same categories as the current state analysis. Figure 9 shows the initial recommendations for the weaknesses identified in the process operations and ideas of the conceptual framework.

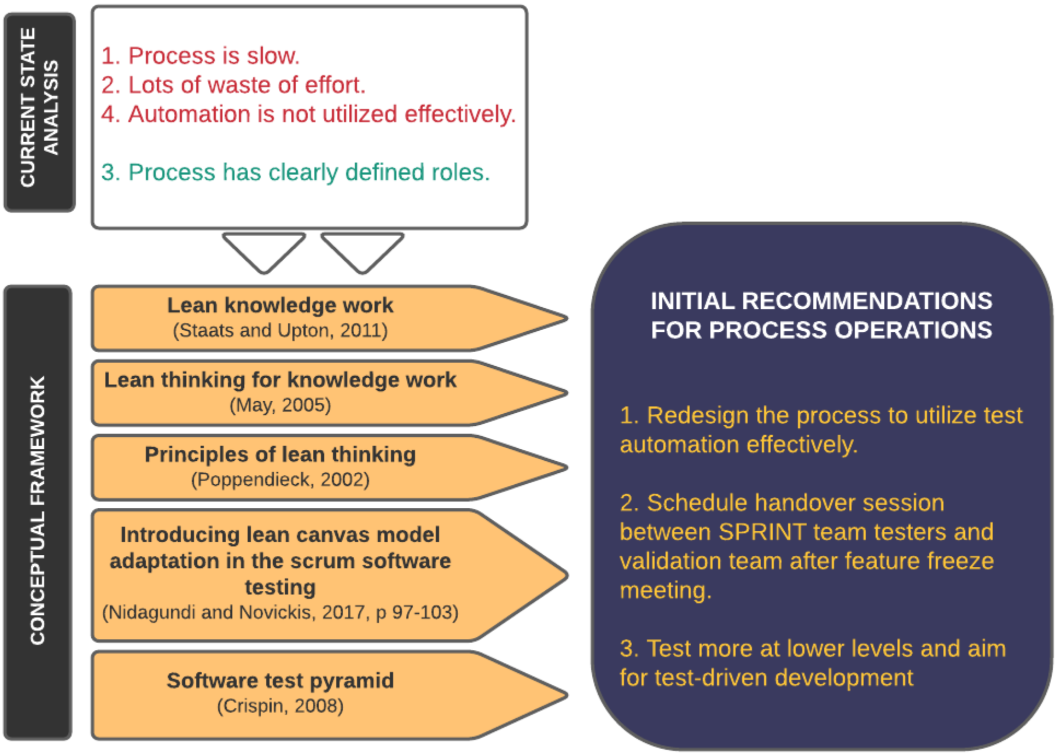

CURRENT STATE ANALYSIS
1. Process is slow.
2. Lots of waste of effort.
4. Automation is not utilized effectively.
3. Process has clearly defined roles.
CONCEPTUAL FRAMEWORK
Lean knowledge work
(Staats and Upton, 2011)
Lean thinking for knowledge work
(May, 2005)
Principles of lean thinking
(Poppendieck, 2002)
Introducing lean canvas model adaptation in the scrum software testing
(Nidagundi and Novickis, 2017, p 97-103)
Software test pyramid
(Crispin, 2008)
INITIAL RECOMMENDATIONS FOR PROCESS OPERATIONS
1. Redesign the process to utilize test automation effectively.
2. Schedule handover session between SPRINT team testers and validation team after feature freeze meeting.
3. Test more at lower levels and aim for test-driven development

Figure 9. Summary of Initial Recommendations for Process Operations

As shown in figure 9, three recommendations for the process operations are suggested. The first recommendation is to redesign the testing process to utilize test automation to its full extent so that ready-to-release software and test evidence are always available rather than waiting for the release activities to start and then start the evidence generation.

The second recommendation focuses on improving communication between SPRINT testers and validation testers to eliminate wastage of time during the validation phase and identify knowledge transfer needs if any.

The third and last recommendation for the process operation category concerns the test creation approach. Automation happens at different stages in the process and tests at earlier stages are faster to execute so focusing on adding more tests in earlier stages will considerably reduce the total execution time for these tests. Also moving towards test-driven development where tests are written first and then the actual development starts will reduce the SPRINT testing times.

In figure 10 initial recommendations for dealing with knowledge transfer issues are summarized. Figure 10 also includes the findings from the current state analysis and relevant ideas in the conceptual framework.

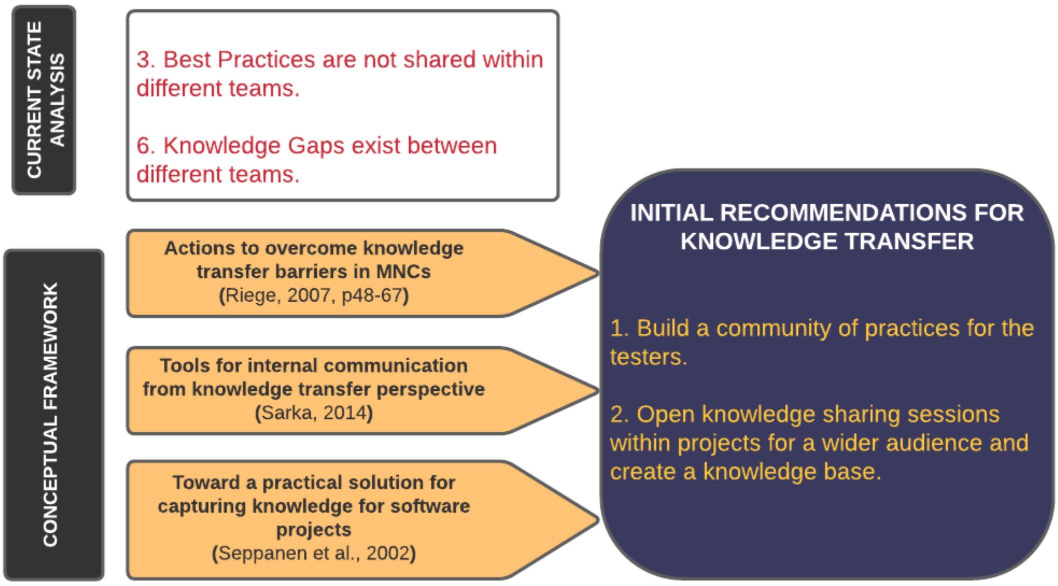

CURRENT STATE ANALYSIS
3. Best Practices are not shared within different teams.
6. Knowledge Gaps exist between different teams.
CONCEPTUAL FRAMEWORK
Actions to overcome knowledge transfer barriers in MNCs (Riege, 2007, p48-67)
Tools for internal communication from knowledge transfer perspective (Sarka, 2014)
Toward a practical solution for capturing knowledge for software projects (Seppanen et al., 2002)
INITIAL RECOMMENDATIONS FOR KNOWLEDGE TRANSFER
1. Build a community of practices for the testers.
2. Open knowledge sharing sessions within projects for a wider audience and create a knowledge base.

Figure 10. Summary of Initial Recommendations for Knowledge Transfer

As shown in figure 10 two recommendations are suggested for this category. The first recommendation to solve knowledge transfer issues among testers of different teams is to build a community of practices for test engineers within the company where different test engineers from different teams can get together and share ideas and exchange knowledge. The idea behind this is to provide a platform for testers where they can highlight the current challenges they are facing and get quick help from the members of the community.

The second recommendation is to open knowledge transfer session that happens within different projects to members of validation team and other testers working in different projects to get this knowledge out of specific projects and make this knowledge sharing as a part of the organization culture.

Figure 11. shows initial recommendations for the key performance indicators category with the identified weaknesses from the current state analysis and relevant themes from the conceptual framework.

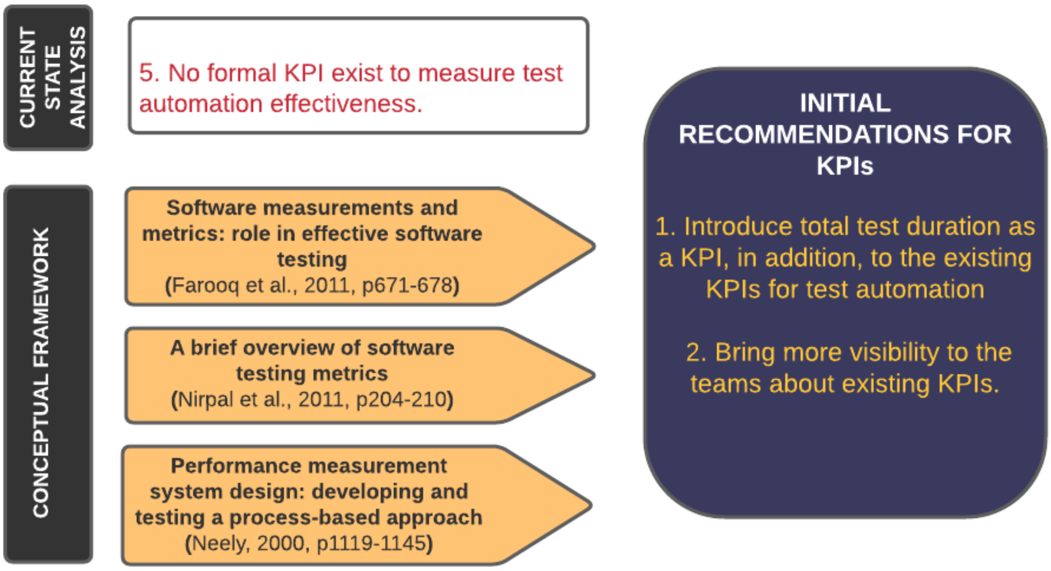

CURRENT STATE ANALYSIS
5. No formal KPI exist to measure test automation effectiveness.
CONCEPTUAL FRAMEWORK
Software measurements and metrics: role in effective software testing (Farooq et al., 2011, p671-678)
A brief overview of software testing metrics (Nirpal et al., 2011, p204-210)
Performance measurement system design: developing and testing a process-based approach (Neely, 2000, p1119-1145)
INITIAL RECOMMENDATIONS FOR KPIs
1. Introduce total test duration as a KPI, in addition, to the existing KPIs for test automation
2. Bring more visibility to the teams about existing KPIs.

Figure 11. Summary of Initial Recommendations for Key Performance Indicators

As shown in figure 11 total of two recommendations are suggested for this category. The first recommendation is to introduce a new key performance indicator to measure the time it takes to run these automated tests. A high number of tests does not guarantee high-quality tests. This KPI is suggested to address the slowness in the overall process and identify tests that needs to be optimized.

The second recommendation is to bring more visibility to the existing key performance indicators. There are key performance indicators that exist which are tracked by test managers and release leads, but visibility is poor to the test engineers in the teams who are ultimately responsible for those KPIs.

Initial recommendations for process improvement, current state analysis findings and themes of conceptual framework are presented in figure 12 below.

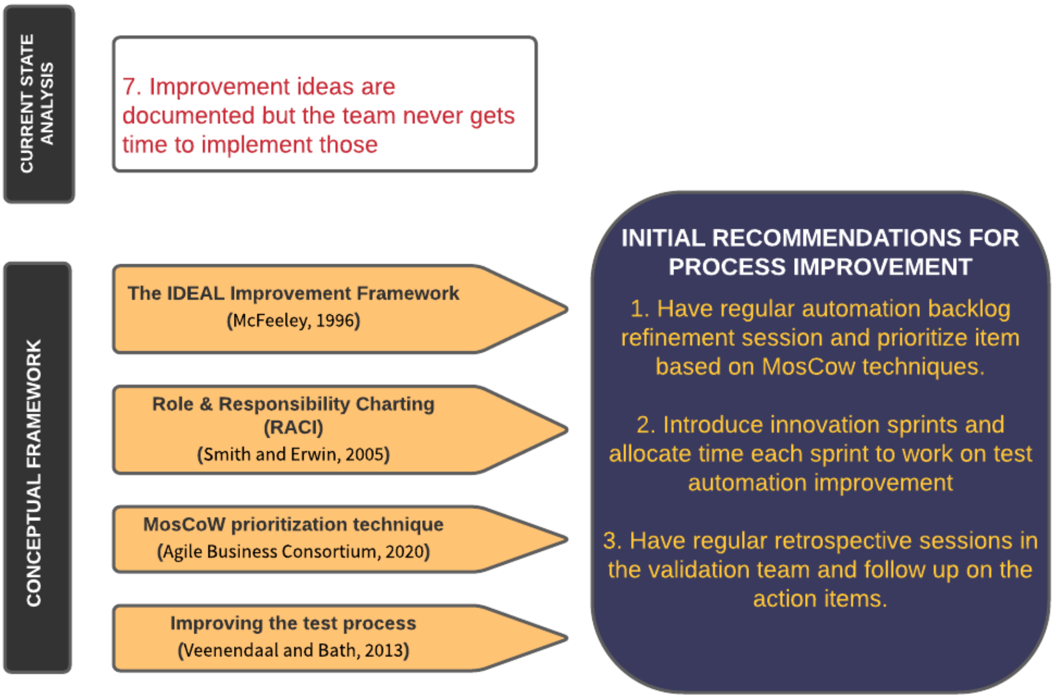

CURRENT STATE ANALYSIS
7. Improvement ideas are documented but the team never gets time to implement those
CONCEPTUAL FRAMEWORK
The IDEAL Improvement Framework (McFeeley, 1996)
Role & Responsibility Charting (RACI) (Smith and Erwin, 2005)
MosCoW prioritization technique (Agile Business Consortium, 2020)
Improving the test process (Veenendaal and Bath, 2013)
INITIAL RECOMMENDATIONS FOR PROCESS IMPROVEMENT
1. Have regular automation backlog refinement session and prioritize item based on MosCow techniques.
2. Introduce innovation sprints and allocate time each sprint to work on test automation improvement
3. Have regular retrospective sessions in the validation team and follow up on the action items.

Figure 12. Summary of Initial Recommendations for Process Improvement

In figure 12 three recommendations are suggested for the process improvement category. The first recommendation for test automation improvement is to have regular sessions within the projects to prioritize the improvement items using the MosCow techniques and push must-have items to the team’s backlog so that test engineers in the team can start working on those.

The second items suggest two approaches that can be taken to handle the increasing test automation improvement items. The first approach is to introduce what is called an innovation SPRINT before the next product planning where teams can work on these items before the new increment starts. The second approach can be to assign 10 to 20% of the time in each sprint to test engineers so that they can work on the must-have items as part of the SPRINT work.

The final recommendation in this category concerns the validation team. Retrospective meetings are an integral part of Agile development, and the basic idea is to continuously improve and identify issues at earlier stages. Currently, these sessions are not regular and follow-up on the action items is quite poor. It has been highlighted that there have been too many action items at times so MosCow should be used here and only must-have items should be taken as action items in the beginning.

The following sections describe the creation of the initial recommendations through internal and external stakeholder interviews in detail.

5.3 Process operations recommendations

During the interviews with internal stakeholders and external expert, process operations were discussed in two phases. The first phase discussions were focused on testing in SPRINT teams and during the second phase testing is validation phase was discussed. It was recognized that despite the higher test automation in certain components the full potential of the test automation is not being utilized and certain changes are required in the current development model and how the current test infrastructure is set up. One of the reasons for tests being unstable is because of ongoing development in the current environment and therefor a need for a separate test environment was identified. One of the interviewees commented on the need of a separate test environment as follows:

> *Development environment is unstable, A separate test environment would help with the stability of test results.*
>
> *(Data 2: Senior test automation engineer)*

The current state analysis highlighted the manual work that happens which includes test evidence generation and identification of components for releasing. One of the reasons for manual evidence generation was legacy tools that were used but recently that has changed in the case company and the teams have started working on newer tools and automating this part is now possible.

> *Test automation should be setup to mark test results in the validation phase with the new tools in place.*

(Data 2: Test Architect)

The testing process is not independent, and it must complement the overall software development strategy and the release strategy. The target releasing strategy was therefore discussed with the managers to understand how the testing process should be redesigned to align with the overall business strategy.

The process map was then modified based on the above inputs and discussed with another test architect. Some valid arguments were made during the interviews about the role of validation testing for components where test automation is high. The value is limited for these components and once the evidence generation is automated the validation testers would have the time that can be spent on performance and other non-functional forms of testing which will not only improve the overall quality of the products, but it will also help in reducing production bugs as well.

Validation team's scope should be more than just functional testing.

(Data 2: External expert)

Figure 13 shows the proposed updated process map for the testing process and the changes in the process map are highlighted with the red border. In the proposed set up a new environment is introduced called a test environment which is like the staging area in a warehouse where goods are temporarily stored waiting to be transported to another area.

In the new setup, process flow starts with increment planning, followed by SPRINT planning in the teams. Once the SPRINT starts testing preparation starts in parallel with the development rather than waiting for the developers to finish their work. Testers start writing the tests based on the requirements and get them peer-reviewed. Once the code is ready these tests are run as earlier, and issues found at this stage go to the team's backlog.

Once a new feature is fully developed and tested all the ready components will move automatically to the test environment. In the test environment, another set of tests called regression tests will be run regularly to make sure these new changes in the components are not breaking the existing functionality of the products and the products still work.

Therefore, ensuring that the test environment has ready to release components at all times. Having a separate test environment will also help with the issue of tests failing because of environmental issues and giving false negatives which will improve the overall confidence. All the defects that are found at this stage of the testing will go back to the team's backlog and will be fixed before the product is released to the end consumers resulting in a fewer number of production bugs that are found after the validation phase.

During the release phase all the ready components automatically move to the next environment for validation testing. In the validation phase in addition to functional testing, User acceptance testing and non-functional testing is performed resulting in better quality software released to market.

Then the development of the automation tests was discussed. It was pointed out that some of the projects are dealing with the problem of reverse test pyramid i.e., where the number of end-to-end and user interface tests are more than unit and integration tests that take significantly less time to execute and more stable. There has been some work going on in this area for quite some time in the test architect team, but individual efforts are needed to solve this issue. However, it was raised that while adding more tests at lower stages should be encouraged within the teams whenever it is possible, but it would require changes in the way the requirements are created and some significant time investments at the team level. Also, all the interviewees suggested that the aim should be test-driven development approach in which tests are developed first and then development starts which ensures faster feedback and rapid development.

> *There is work ongoing regarding this but individual effort should be done to move away from UI tests.*
>
> *(Data 2: Test Architect)*

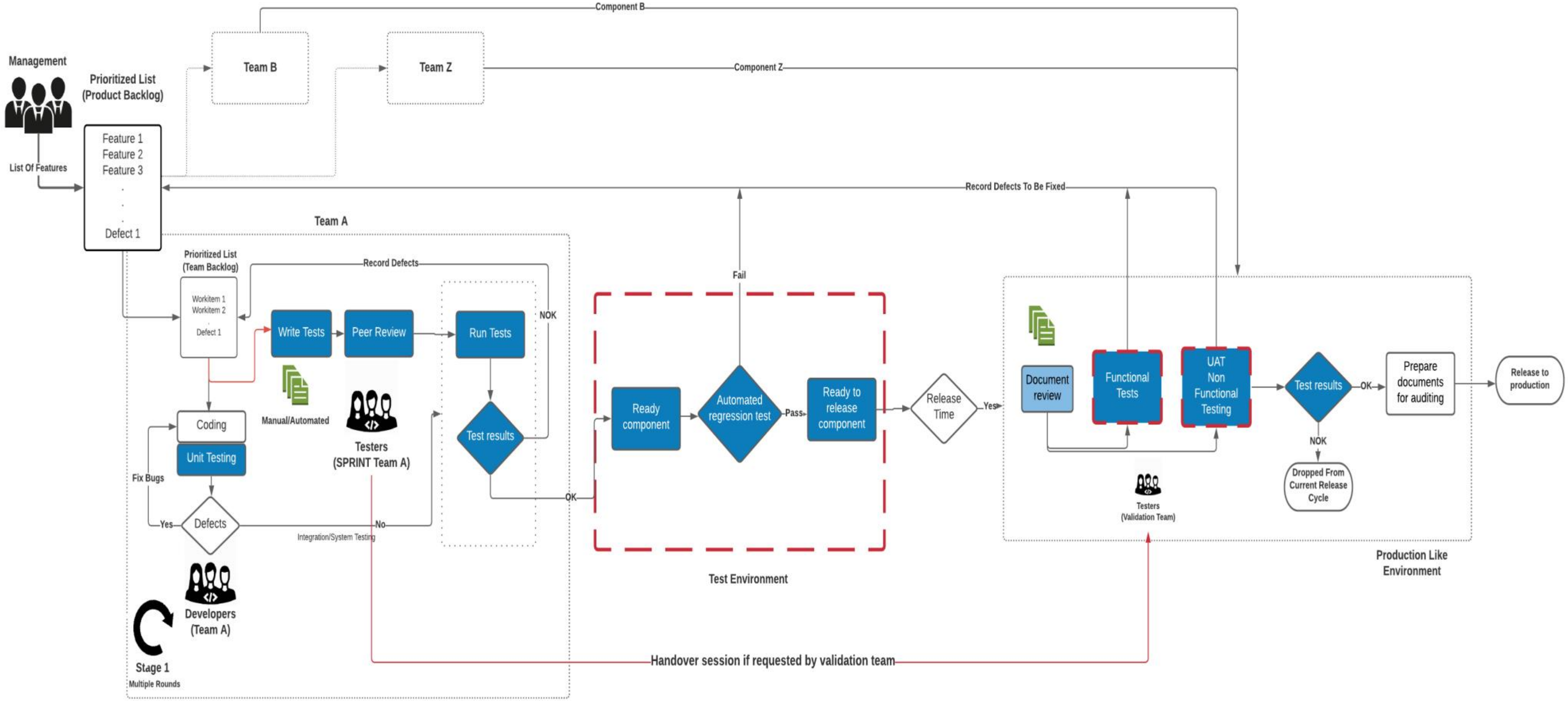


Figure 13. Updated Process Flow of the Testing Process

After the process map was drawn, issues raised by the validation testers were discussed. Validation testers have been requesting handover sessions from SPRINT team testers before the validation testing start. It was pointed out that the case company used to have handover sessions in the past where all the SPRINT team testers would do knowledge transfer to validation testers but those were stopped because those sessions were not organized properly, and it was just adding more overhead in the process. Therefore, it was recognized that on-demand sessions make much more sense rather than having all the teams providing handovers. One of the interviewees responded on the issue as:

> *It would be easier if validation team would review the content and assess the need of handover session.*
>
> *(Data 2: Release Lead)*

5.4 Knowledge Transfer Recommendations

In the current state analysis, it was highlighted that test automation coverage varies greatly within different projects and one of the reasons for that was the skill gaps that exist between test engineers of different teams. Also, the validation team has highlighted the need for regular knowledge transfer sessions, especially for test automation. To address this first step would be to capture the knowledge transfer needs of the teams.

The test managers would be responsible for capturing knowledge transfer needs in the SPRINT teams and the release lead would be responsible for capturing it in the validation team. Once the knowledge transfers needs are identified the next step would be to identify the internal experts within different teams and arrange knowledge transfer sessions for other R&D testers. It is also important to capture this knowledge and store this in a common repository that can be accessed by all the R&D testers. Also, the possibility of arranging some external trainings was discussed to speed up this process of bridging the knowledge gaps. It was mentioned that this has already started in few SPRINT teams, therefore, was not included as a recommendation.

During the discussions, it was also brought up that some of the SPRINT teams have the habit of regular knowledge transfer sessions but those are mostly limited to that team only. It was suggested that it would be a good idea to open those sessions for the wider audience and record those sessions and make them available to other testers.

> *We need to find ways to break silos in the teams and bring the knowledge within teams to other members of the organization.*
>
> *(Data 2: Test Architect)*

A suggestion was made by the external expert to reshuffle the team members in the SPRINT teams so that each project will have some experts helping with the daily tasks and bringing up the test automation coverage in other projects. The same suggestion was brought to the internal stakeholders, but team managers were not so keen on disrupting the teams because of prior commitments to already finalized released schedules.

While internal knowledge sharing sessions and external training are a good start to fill the gaps, it is important that knowledge sharing is part of the organization culture. It was recognized that test engineers will need continuous support from the experts to help them with the issues regularly till these gaps are bridged. To address this the external expert suggested the idea of starting a community of practices for the R&D testers which will provide all the testers a platform to share ideas, thoughts and get quick solutions to the problems they are facing. It will also help in building a knowledge base for the team. The case company already has a community of practices for the scrum masters of the teams, so the idea was supported by other interviewees as well. Also, scrum masters can help with getting started with the community.

> *Starting a community of practices would be very useful and act as a continuous support system. I would recommend one session every two weeks.*
>
> *(Data 2: External expert)*

5.5 Key Process Indicator Recommendations

The co-creation of recommendations for key performance indicators started by highlighting the need to measure the effectiveness and performance of the test automation. In addition, the visibility of the existing KPIs needs to be improved to have better accountability and achievability.

The existing KPIs to measure the effectiveness of the testing were then discussed. The case company has a few KPIs like the total number of defects that are found in production after a release is done and the test automation coverage i.e., the number of manual tests that are automated. During the interviews, it was highlighted that these are tracked by the R&D managers and agreed that there should be more visibility of these KPIs to the people who are responsible to achieve these. A recommendation was suggested to discuss these KPIs regularly in the test team meeting and bringing more visibility to the team members.

> *We do have some KPI's, but we need to make it more visible to the teams. These were discussed earlier in the test team meeting.*
>
> *(Data 2: Release lead)*

The next topic was KPIs that exist specifically for test automation. The existing test automation coverage KPI is useful to measure how many test cases have been automated but it does not tell anything about the quality of tests and the time it takes to execute those tests. Multiple interviewees suggested total test duration to be a useful KPI and suggested tracking it every release to measure how much slower or faster the test automation is getting. The test execution times should be tracked regularly, and slow areas should be identified, and improvement items should go to the test automation backlog, be prioritized, and make it to the SPRINT team's backlog.

Measuring code coverage was another KPI for test automation that was suggested and while interviewing the test architects it was found out there is work ongoing in this area already to make this visible as part of the existing test automation setup.

> *Test automation coverage is a very fluctuating KPI, we should also be measuring how much time we are taking to execute tests and how many bugs are we leaking into production.*
>
> *(Data 2: Release lead)*

> *We should be measuring test automation times all the time and how to get the execution time lower.*
>
> *(Data 2: Test Architect)*

5.6 Process Improvement Recommendations

One common issue that was highlighted in both SPRINT teams and the validation team during the current state analysis was that either team does not get time to implement those improvements or there is no action taken on the ideas suggested by the team. The interview with the release lead focused on the issues in the validation team and the test manager was interviewed to address issues in the SPRINT teams.

During the interview with the release lead, the discussion started with the sprint retrospective meeting which is an important ceremony in agile development. The whole idea of a retrospective meeting is to reflect on the team's ways of working and continuously improve the quality and effectiveness. The validation team does not have regular retrospective meetings in the first place which needs to be improved.

Then how these meetings are conducted was discussed in detail. The team often struggles with following up on action items because there are too many that come out of those meetings. It was then suggested to use the MosCow technique to prioritize the action items and follow up on only must-have action items. Also, regular retrospective sessions will make it easier to follow up as well. Release leads should be responsible to bring more visibility to the improvement ideas suggested by the validation team that is meant for the SPRINT teams and weekly test team meetings can be utilized to do that.

> *Follow-up on the action items is poor we always have more action items than we can follow-up on. These action items should get more visibility.*
>
> *(Data 2: Release Lead)*

The issue of SPRINT teams never getting time to work on test automation improvement items was then raised in the interviews with the test manager and the external expert. The test automation improvement items have increased over time, and it is important to work on these improvement items regularly for the process to remain effective. Therefore, there is a need for prioritizing these items. The test manager should be the owner of the list and regular refinement of items with the SPRINT test engineers is needed. The MosCow technique should be applied in the refinement meetings to prioritize the items on the list.

> *The test automation backlog should be maintained and prioritized on a regular basis.*
>
> *(Data 2: External expert)*

Since the items are now prioritized next step would be to allocate time to the testers in the SPRINT teams to start working on it. Some of these items will require dedicated developer hours to implement therefore support from the product owners was recognized as crucial.

Two suggestions came from the external expert on how to find time to work on these items. The first suggestion was to introduce innovation SPRINT that is after every increment few days will be allocated to innovation where different members of the SPRINT teams can work on the improvement items and test engineers can focus on implementing the automation improvement items, brainstorm on new ideas, etc.

The second suggestion was to allocate one or two days of SPRINT time to work on automation improvement items. The improvement items must be included in the product backlog so it was suggested that before every increment planning the test managers can discuss these items with the product owners and must-have items will make it to the product backlog. Once the items are in the product backlog different teams will pull those items in the team’s backlog and commit to implementing those.

> *Every sprint should have 20 percent of innovation capacity.*
>
> *(Data 2: External expert)*

The co-creation of initial recommendations was done successfully with active participation from key internal stakeholders and an external expert. Figure 14 presents the initial recommendations on the process map. Some improvements are general improvements, therefore, are listed separately. All the weaknesses identified in the current state analysis were addressed and a total of ten recommendations were created. Section 6 describes the validation of the co-created initial recommendations.

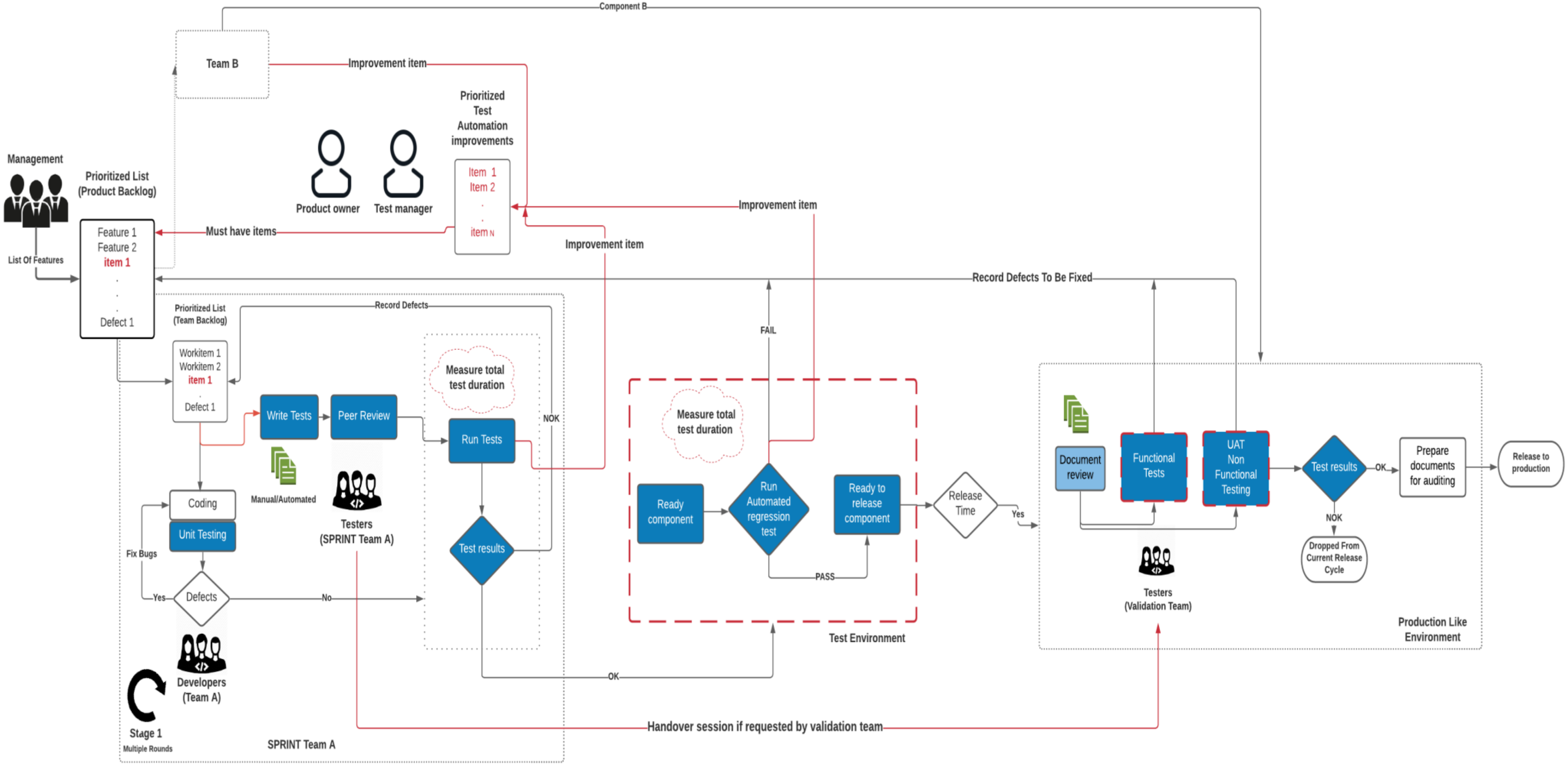


Figure 14. Initial Recommendations for the Testing Process

6 Validation of the proposed recommendations

Section 6 describes the validation of the initial recommendations created in the previous section. This section starts with an overview of the stage in general. Then the final recommendations are summarized. After the final recommendations, the feedback received from the management is described and finally, changes made to the initial recommendations are presented.

6.1 Overview of the Validation Stage

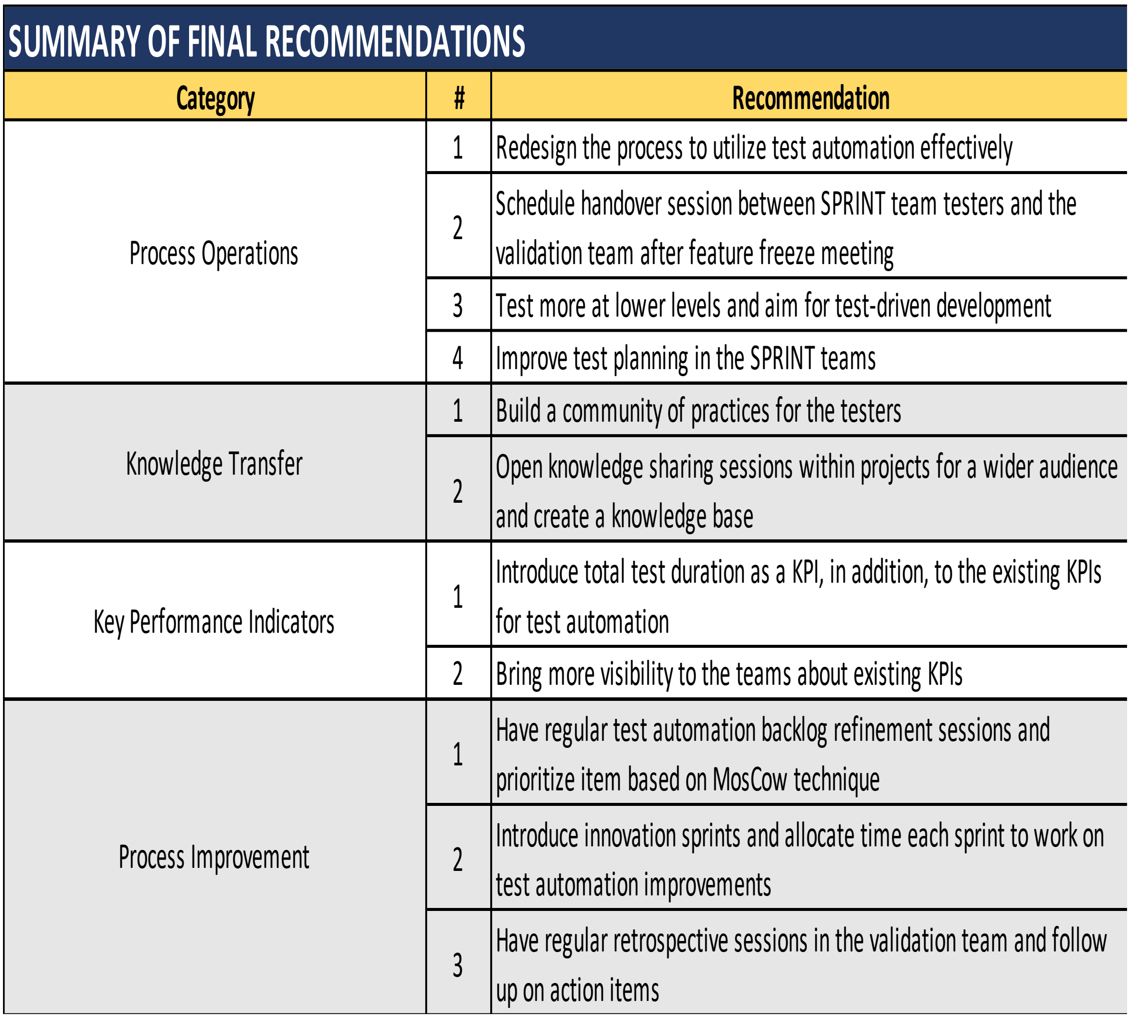
This section validates the initial recommendations created in section 5. The business need of reducing time to release to market and better utilization of test automation were the main drivers in defining the objective of the thesis. The initial recommendations were co-created with the key stakeholders of the process and an external expert focusing on key areas from the current state analysis. These areas included lean process operation, effective knowledge transfer, missing KPIs, and general improvement of the testing process.

The validation of the proposal was done by presenting the co-created recommendations to senior R&D managers of the case company to get their feedback and critique. The feedback was important to evaluate the effectiveness and feasibility of implementation from the business perspective.

The validation of the proposal was done in two online meetings arranged on Microsoft Teams application. The first meeting was arranged with the R&D director and the second meeting was with the VP of testing services. The meetings started with the general introduction of the study, followed by key focus areas identified during the current state analysis and themes of the conceptual framework. Then the initial recommendations for each category were presented and discussed generating DATA 3 for the study. The initial recommendations were then adjusted based on the feedback received during these meetings to generate final recommendations. The final recommendations for test process improvement are summarized into the same categories used throughout the study and presented in Table 8 below.

As shown in the table 8, only change in the recommendations was in the process operations category. Initial recommendations can be found in Section 5.2 and changes made to the initial recommendations are covered in Section 6.4.

Table 8. Summary of the Final Recommendations.



SUMMARY OF FINAL RECOMMENDATIONS

Category	#	Recommendation
Process Operations	1	Redesign the process to utilize test automation effectively
	2	Schedule handover session between SPRINT team testers and the validation team after feature freeze meeting
	3	Test more at lower levels and aim for test-driven development
	4	Improve test planning in the SPRINT teams
Knowledge Transfer	1	Build a community of practices for the testers
	2	Open knowledge sharing sessions within projects for a wider audience and create a knowledge base
Key Performance Indicators	1	Introduce total test duration as a KPI, in addition, to the existing KPIs for test automation
	2	Bring more visibility to the teams about existing KPIs
Process Improvement	1	Have regular test automation backlog refinement sessions and prioritize item based on MosCow technique
	2	Introduce innovation sprints and allocate time each sprint to work on test automation improvements
	3	Have regular retrospective sessions in the validation team and follow up on action items

6.2 Findings of validation stage, Data Collection 3

The initial recommendations received positive feedback from the senior management. The work done during the study was appreciated and seen as beneficial for the test process improvement.

> *These are really good findings, and practical improvement suggestions.*
>
> *(Data 3: R&D Director)*

The redesigned process got the most attention and was discussed in more detail. The introduction of the new stable test environment was recognized as a valid addition. The case company is moving to the cloud infrastructure and therefore setting up the test environment was now feasible with minimum investment. Also, a stable test environment was considered useful in saving a lot of development hours wasted trying to debug the issues due to an unstable environment. The role of the validation project has always been under discussion in the case company. While validation project does bring independence to the testing and make sure the regulatory requirements are fulfilled it was acknowledged that the value is quite limited for the projects with high test automation. The suggestion regarding increasing the scope of testing was seen as the next step in the overall quality journey and was seen as ideal.

> *I agree that the value is quite limited for the automation projects. The ideal setup would be where validation team is doing non-functional testing, exploratory testing which will improve quality of our products too.*
>
> *(Data 3: VP Testing Services)*

There was an argument made regarding the number of test cases that are executed during the validation phase. A valid point was raised that there is no need to execute so many regression test cases in every release especially in the future when the target is to release as often as possible because changes will not be huge. This suggestion was also aligned with the recommendation of testing more at lower stages in the process and therefore was taken as an additional recommendation.

> *There is no need to run so many tests in the validation stage, with proper test planning and carefully selected tests that should be run in validation phase we could speed up delivery even for projects where test automation is not that high.*
>
> *(Data 3: R&D Director)*

The recommendation of building a community of practices for testers was seen as a good starting point for bridging knowledge gaps in different teams. The idea of opening knowledge-sharing sessions for a wider audience was seen as something that should be encouraged.

The recommendation of introducing additional key performance indicators for test automation was considered important to keep a check on the quality of test automation. Since test automation plays a critical role in achieving delivery targets, measuring total test duration, and introducing it as a KPI was important going forward.

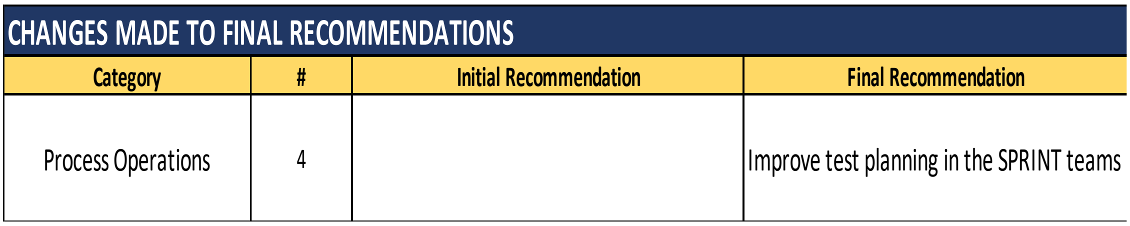
> *I would say we as an organization should be bold enough while setting targets for total execution times.*
>
> *(Data 3: R&D Director)*

The retrospective sessions were acknowledged to be important for overall process improvement and release leads are responsible for following up on action items. The idea of innovation sprint was already under discussion among senior management as well and will be introduced very soon in the case company. Prioritizing improvement items with test managers and product owners and pushing must-have items to product backlog was seen as logical. However, it was argued that improvement items not getting enough attention was not just a lack of time issue but also a cultural issue.

6.3 Changes Made to the initial Recommendations

One more recommendation was added based on received feedback from senior management in the process operation category.

Table 9. Changes to initial recommendations.



CHANGES MADE TO FINAL RECOMMENDATIONS			
Category	**#**	**Initial Recommendation**	**Final Recommendation**
Process Operations	4		Improve test planning in the SPRINT teams

As shown in Table 9, the recommendation for improving the test planning was made for the SPRINT teams. SPRINT team testers are responsible for creating test plans which include the tests that will be executed by the validation team during the validation phase; therefore, the suggestion was made only for the SPRINT team test engineers. This will reduce the number of tests that the validation team needs to run, and more focus can be given to other types of testing.

The next section presents an executive summary, recommendations for implementation, self-evaluation, and finally concludes the study.

7 Conclusions

The final section of the study includes the executive summary of the thesis followed by recommendations for implementation. Finally, self-evaluation of the study and the closing words are presented.

7.1 Executive Summary

The objective of the thesis was to generate recommendations to improve the software testing process in the case company. Software testing not only ensures the quality of products but also fulfills the regulatory requirements of the industry. The testing process plays a critical role in overall software development and releases to market. The case company has made a significant investment in test automation to address the demand for faster delivery, but the testing process has not evolved much. Therefore, the current setup is inefficient in utilizing the full potential of test automation. Improvements in the test process are needed to achieve the case company's strategic targets of faster delivery to the market.

The research approach used in the study was qualitative research. The study was done in four phases and data for the study was gathered through three data collection rounds. The first phase of the study was the current state analysis of the testing process. The data was gathered through stakeholder interviews and reviewing internal documents. Then existing literature was reviewed to ger relevant ideas resulting in the conceptual framework for the study. In the next phase recommendations were generated with the help of key stakeholders and an external expert. Finally, recommendations were validated with the senior managers.

The weaknesses identified in the current state analysis were slowness in the process, inefficient use of test automation, Knowledge transfer issues, missing KPIs, and challenges in implementing improvement ideas. These weaknesses were divided into four categories to enable more focused research on improvement ideas in the literature. The process operations category focused on making the process leaner. The Knowledge transfer category focused on knowledge gaps and sharing best practices. The Key performance indicator category included missing KPIs for test automation. The last category, process improvement targeted time-related issues in process improvement.

Three initial recommendations were suggested in the process operations category. Introducing a stable test environment and handover sessions between teams were suggested to eliminate wastage of time during later stages. Other recommendations were more testing at earlier stages and test-driven development. The process was redesigned to utilize test automation more effectively as shown in figure 13.

Two recommendations were suggested to handle knowledge transfer issues. The recommendations included starting a community of testers to enable a knowledge-sharing culture and provide support to less experienced team members. The second recommendation was to open knowledge-sharing sessions within projects to a wider audience and create a knowledge base that is available to everyone.

The key performance category had two recommendations. The first one was to introduce total execution time as a complimentary KPI for test automation coverage to monitor and control execution time and improve the quality of test automation. Bringing more visibility about existing KPIs to team members was also suggested.

Three initial recommendations were suggested to tackle the proper implementation of improvement ideas. Regular prioritization of improvement items using MoSCoW and follow-ups on action items suggested in retrospective sessions were recommended. To address the lack of time innovation sprint and allocating ten to twenty percent of SPRINT time on test automation improvement was recommended.

The initial recommendations were then validated by the senior R&D managers in two meetings. The overall response was positive, and recommendations were considered logical and practical. One more recommendation related to better test planning was suggested and added to the final list of recommendations.

The final recommendations provide a list of practical steps that can be taken to improve the testing process in the case company. The testing process is at the core of software development and these improvements will not only have a positive effect on the quality of software products but also help the case company achieve its strategic goals of releasing faster to market.

7.2 Next Steps and Recommendations toward Implementation

The study highlighted the weaknesses in the process and proposed a new process set up and new practices that will help in improving the testing process and reducing the overall time it takes to release new features.

The proposed new setup will change the ways of working in the teams and require time and infrastructure investment to implement. Therefore, the first step would be to build a detailed action plan. Setting up the infrastructure will require expertise from other members of the organization mainly software architects and release engineers. So, the next step would be to select people with this expertise and do a proof of concept in a medium -size project and later demo it to the entire R&D.

Additionally, new suggested practices such as prioritizing improvement items and allocating time each SPRINT for test automation improvement require allocation of few hours every week and can be promoted easily with the support from test managers and product owners.

7.3 Thesis Evaluation

The research objective of the thesis was to generate recommendations to improve the testing process that was set to address the initial business problem of faster delivery and improved quality of the test automation. The current state analysis targeted all the stages in the process and highlighted key weaknesses in the current process. The outcome of the thesis, generated recommendations directly target those weaknesses in the current testing process and provide practical solutions that will have a positive impact on the performance of the process. Also, the new proposed setup was suggested keeping the case company’s current investment in the new tools and infrastructure. Therefore, the objective of the thesis was met fully.

A diverse group of stakeholders was chosen including people outside the testing organization keeping in mind that the testing process is a supporting process, and it affects the overall software development process. A lot of the strengths and weaknesses were identified some of those were repeated by multiple stakeholders. However, the study does not guarantee that all the strengths and weaknesses were identified during the current state analysis. Validated recommendations to the identified weaknesses fulfill the objective of the study.

The clear categorization of the weaknesses helped in identifying specific themes in the existing literature that were relevant to the industry. The author of the thesis is a member of the testing team which proved to be very helpful during the interviews and promoted more open discussions. Interviewing the external expert proved to be very useful and brought fresh ideas and perspective to the discussions with internal stakeholders.

The case company was going through a lot of organizational changes during the time of the study which created a little bit of challenge to arrange these discussions and keep up with the study plans. Also, during the discussions, it was challenging to keep the focus on the process side of the testing because of the technical backgrounds of the participants and semi-structured interviews.

The thesis is evaluated on four criteria validity, reliability, logic, and relevance in the following sub-sections.

7.3.1 Validity and Reliability

According to Guba et al. (1985), validity is one of the important criteria to establish rigor and trustworthiness in qualitative studies. In qualitative research, internal validation helps in establishing credibility in the study and external validity answers the question of transferability of the results beyond the context of the study.

Creswell et al. (2000) state that there are multiple ways of establishing validity in qualitative projects such as triangulation, peer reviews, external audits, etc. Shenton (2004) also suggests the use of triangulation to promote confidence in the study and a way to establish trustworthiness. Triangulation can be established using diverse methods for data collection such as observations, individual interviews, etc., and by using a wide range of data informants (Shenton 2004: 63-66).

This study utilizes triangulation to establish internal validity. Multiple methods such as reviewing internal quality management system documents, regulatory requirements, and individual interviews were used to gather Data 1 for the current state analysis. Also, Data 2 was gathered from multiple rounds of interviews to create initial recommendations.

Data source triangulation was achieved by selecting participants from different teams involved in the process operations. In addition, an external expert was interviewed to bring a new perspective to the process. The selected senior managers interviewed during the validation stage belonged to different organizations in the case company.

Although the result of the study is not fully transferrable some elements can be utilized to improve the testing process in a different I.T organization. The proposed new setup gives a blueprint of what a software testing process can look like. Recommendations for knowledge transfer and handling of test automation improvement items can be utilized by a different I.T organization.

Shenton (2004) addresses reliability as the extent to which the results would be consistent if the research was repeated in the same context, with the same data collection methods and same participants. Research design and its implementation, data gathering methods, and reflective appraisal should be reported in detail to increase reliability (Shenton 2004: 71-72).

The reliability was ensured by clearly defining the research design of the study and data gathering methods in section 2. Detailed implementation of the research design is covered through current state analysis, literature review, recommendation creation, and followed by validation of the recommendations present in sections 3, 4, 5, and 6. Also, detailed self-evaluation is provided in section 7.

7.3.2 Logic and Relevance

The outcome of the thesis was to find a practical solution to a business problem that is relevant only to the case company therefore design research was selected. The study followed the logic of design research. The logic was ensured by creating a plan for the study in the beginning. The research design shows logical steps that were taken to complete the study starting with identifying the current state of the process, followed by identifying relevant improvement ideas from literature, developing a solution with the help of key stakeholders, and finally, validating the solution with the senior management of the case company.

The business problem of an inefficient test process and an urgent need for improvement was the main driver behind conducting this research. The outcome of the thesis focuses on improving the said process therefore the subject is relevant for the case company. Relevance in the study was ensured by involving key stakeholders in the process throughout the study. Also, the conceptual framework themes were selected to target the identified weaknesses.

The initial recommendations were evaluated and validated by the senior management of the case company based on their relevance. All the proposed recommendations were considered relevant during the validation phase and none of the recommendations was rejected by the senior management increasing the relevance of the proposal.

7.4 Closing Words

Software testing not only plays a critical role in ensuring the safety and quality of software products but also very expensive, yet it is hardly looked at from the process perspective in organizations. Processes must evolve and organizations should always look for areas of improvement and the testing process is not any different. The outcome of the thesis provides recommendations on how improvements can be made in the testing process of the case company. The next step would be the planned implementation of these recommendations which is outside the scope of the thesis.

References

Agile Business Consortium. (2020). Agile Business Consortium website. Chapter 10: MoSCoW Prioritisation. Available from: https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation (Accessed 9 March 2021).

Beizer, Boris (2012). Software Testing Techniques, second edition. Dreamtech Press

Cambridge Judge Business School (2020) . A quantitative study of the financial impacts of software failures on enterprise organizations. Available from: https://undo.io/the-cost-of-software-failures/ (Accessed 15 Dec 2020)

Cohn, Mike (2009). Succeeding with Agile.1st edition. Addison-Wesley Professional. Available from: https://learning.oreilly.com/library/view/succeeding-with-agile/9780321660534/ (Accessed 14 March 2021)

Creswell, John W. & miller, Dana L. (2000). Determining Validity in Qualitative Enquiry. Theory Into Practice. Vol. 39, pp 124-129

Creswell, John W. (2014). Research Design: *Qualitative, Quantitative and Mixed Methods Approaches.* 4th edition. Sage Publications

Crispin, Lisa & Gregory, Janet (2008). *Agile Testing: A Practical Guide for Testers and Agile Teams*. 1st edition. Addison-Wesley Professional. Available from: https://learning.oreilly.com/library/view/agile-testing-a/9780321616944/ (Accessed 14 March 2021)

Farooq, Sheikh Umar, Quadri, S.M.K & Ahmad Nesar (2011). Software Measurements and Metrics: Role in Effective Software Testing. International Journal of Engineering Science and Technology. Vol. 3, pp. 671-678

Ghobadi, Shahla & Mathiassen, Lars (2015). Perceived barriers to effective knowledge sharing in agile software teams. Information System Journal. Available from: https://www.researchgate.net/publication/269777015_Perceived_barriers_to_effective_knowledge_sharing_in_agile_software_teams (Accessed 17 Feb 2021)

Guba, Egon. G. & Lincoln, Yvonna S. (1985). Naturalistic inquiry. Sage Publications. Chapter 11, Establishing Trustworthiness, pp. 290

Hatton, Sarah (2008). Choosing The Right Prioritization Method. Conference Software Engineering.

Kananen, J. (2013). Design research (applied action research) as thesis research: A practical guide for thesis research. Jyväskylä: Jyväskylän ammattikorkeakoulu

May, Matthew (2005). *Lean Thinking for Knowledge Work. PP 33-40.* Available from https://tkmg.com/wp-content/files/Lean-Knowledge-Worker.pdf (Accessed 20 Feb 2021)

Munson, J. C. (2003). Software Engineering Measurement. CRC Press, Inc., Boca Raton, FL, USA.

Nidagundi, Padamaraj & Novickis Leonids (2016). Introducing Lean Canvas Model Adaptation in the Scrum Software Testing. Available from www.sciencedirect.com (Accessed 12 March 2021)

Neely, Mills, Platts, Richards, Gregory, Bourne & Kennerley (2000). Performance Measurement System Design: Developing and Testing Process Based Approach. International Journal of Operations and Production Management. Vol. 20, pp. 1119-1145

Nirpal, Premal B. And Kale, K.V. (2011). A Brief Overview of Software Testing Metrics. International Journal on Computer Science and Engineering. Vol. 3, pp. 204-210

Poppendieck, Mary (2002). Principles of Lean Thinking. Available from https://www.okyanusbilgiambari.com/bilgiambari/Yalin/Yalin.Fabrika/Leanthinking.Software.pdf (Accessed 17 Feb 2021)

Riege, Andreas (2007). Actions to overcome knowledge transfer barriers in MNCs. Journal of knowledge management. Vol 11, pp. 48-67. Available from: https://www.researchgate.net/publication/220363247_Actions_to_overcome_knowledge_transfer_barriers_in_MNCs (Accessed 24 Feb 2021)

Sarka, Hudcova (2014). Tools of Internal Communication from Knowledge Transfer Perspective. Journal of Competitiveness. Vol. 6, Issue 4, pp. 50-62. Available from: https://www.cjournal.cz/files/180.pdf (Accessed 22 March 2021)

Saunders, M., Lewis, P. & Thornhill, A. (2016). *Research methods for business students*. 7th ed. Harlow: Pearson Education.

Seppänen, Veikko, Mäntyniemi, Annukka & Komi-Sirviö, Seija (2002). Toward a Practical Solution for Capturing Knowledge for Software. pp. 60-62. Available from: https://www.researchgate.net/publication/3247857_Toward_a_practical_solution_for_capturing_knowledge_for_software_projects (Accessed 24 Feb 2021)

Shenton, A. (2004). Strategies for Ensuring Trustworthiness in Qualitative Research Projects. Education for Information, IOS Press, 22(2), PP. 63-75.

Smith, Michael L & Erwin, James (2005). Role and Responsibility Charting (RACI). Available from: https://pmicie.org/files/22/PM-Toolkit/85/racirweb31.pdf (Accessed 9 Mar 2021)

Staats, Bradely R. & Upton, David M. (2011). *Lean knowledge work*. Harvard Business Review, pp. 100-110

Veenendaal, Erik Van & Bath, Graham (2013). Improving the Test Process. 1st edition. Rocky Nook. Available from: https://learning.oreilly.com/library/view/improving-the-test/9781492001300/ (Accessed 14 March 2021)

Vestola, Mikko. (2010). *A Comparison of Nine Basic Techniques for Requirements Prioritization*. Helsinki University of Technology, Helsinki. Available from: http://www.mvnet.fi/publications/software_development_seminar.pdf (Accessed 9 Mar 2021)

Vocke, Ham (2018). The Practical test Pyramid. 26 Feb 2018. Available from: https://martinfowler.com/articles/practical-test-pyramid.html (Accessed 16 March 2021)

Wong, S.S., DeSantis, G. and Staudenmayer, N. (2007). The Relationship Between Task Interdependency and Role Stress. Journal of Management Studies. Vol. 44, pp. 284-303

Interview questions during the current state analysis

1. Is there a process description of the testing process?
2. What are the regulatory requirements that need to be followed?
3. How do we ensure the quality of the automation tests?
4. How are the test improvement items prioritized?
5. What kind of requests do you get from the validation team?
6. Why only a few projects have a high level of automation?
7. What kind of challenges do you face in SPRINT teams?
8. What kind of challenges do you face during validation testing?
9. What are the strengths of the testing process in your opinion?
10. What are the weaknesses of the testing process in your opinion?
11. Free word?