



CHALMERS
UNIVERSITY OF TECHNOLOGY



Evaluation of New Product Development Process and Organization

A case study of connected solutions company
Master's thesis in Quality and Operations Management

HYEJIN JEONG

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF SERVICE MANAGEMENT AND LOGISTICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se
Report No. E2023:115

REPORT NO. E2023:115

Evaluation of New Product Development Process and Organization

A case study of connected solutions company

HYEJIN JEONG

Department of Technology Management and Economics
Division of Service Management and Logistics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023

Evaluation of New Product Development Process and Organization
A case study of connected solutions company
HYEJIN JEONG

© HYEJIN JEONG, 2023.

Report no. E2023:115
Department of Technology Management and Economics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone + 46 (0)31-772 1000

Gothenburg, Sweden 2023

Evaluation of New Product Development Process and Organization

A case study of connected solutions company

HYEJIN JEONG

Department of Technology Management and Economics
Chalmers University of Technology

SUMMARY

Customer needs have been changed to experience products as early as possible with similar or even higher quality products than in the past. Therefore, to align with customer needs, firms' way of working has changed significantly to increase their velocity and flexibility when developing products. However, since the way of working is closely related to new product development (NPD) processes and organizations, merely changing the way of working cannot meet customer needs. Therefore, these three aspects should be considered when tailoring NPD processes.

The case company's way of working is continuously evolving to increase not only product quality but also efficiency by adopting a Scaled Agile Framework (SAFe) to meet customers' expectations through its NPD process. However, it is important to evaluate whether the current way of working, the NPD process, and the organization are closely managed together. However, the case company raises the need to evaluate the current way of working, the NPD process, and whether the organization is closely managed together, and to find potential improvements to meet customer needs. Therefore, this study was conducted according to the request of the case company.

Mixed research combining qualitative and quantitative approaches was adopted to evaluate the current situation of the case company. The primary source of the research is a qualitative approach that is in line with interviews and literature reviews, and the quantitative approach supports findings based on a qualitative approach. The purpose of conducting quantitative research, which is a survey and a self-completion questionnaire, was to investigate employees' perceptions. The self-completion questionnaire was aimed at evaluating the current SAFe maturity level in the case study. After analyzing the data, two benchmarks were conducted to gain ideas about how other companies are dealing with the challenges faced by the case company.

The study identified three categories of challenges: process-related, organizational-related, and knowledge management-related challenges. Furthermore, root cause analysis was conducted to determine where the challenges derived from the three categories of challenges, and the identified root causes are the inflexible NPD process, late risk management, decentralization, lack of customer involvement, and lack of knowledge management. Therefore, recommendations are based on aiming to decrease time-to-market and increase product quality to overcome the identified challenges, since time and quality are perceived as the most important factors when developing a new product at the case company.

Keywords: New Product Development (NPD) process, Scaled Agile Framework (SAFe), Agile process, Process tailoring, Agile testing, Capability Maturity Model Integration (CMMI), Centralization, Decentralization, Ambidexterity, Formalization, and Knowledge Management.

Acknowledgement

This master's thesis was conducted from January to June 2023 at the Department of Technology Management and Economics, Chalmers University of Technology. And this master's thesis study was performed at the case company in Gothenburg, Sweden.

First, I want to appreciate Andreas Hellström, my supervisor at Chalmers University of Technology, for providing valuable input and guidance for me to proceed with this report. In addition, by guiding me to be humble and objective in conducting the research study, I was able to analyze it based on the theoretical framework and avoid leaning toward personal preferences. Furthermore, I learned that I have to focus on a holistic view instead of a micro one when conducting a research study.

Next, I would appreciate the Product Portfolio & Planning team at the case company, which admitted me as a master's thesis student with a warm welcome. I would especially like to express my appreciation to Emin Mahjoub and David Lexell as my industrial supervisors, who gave me valuable feedback and support in conducting this thesis. When I applied for the master's thesis position, I was eager to conduct research on this topic not only for my interest but also to experience Swedish company culture, especially the case company as a core company in Sweden. I have achieved what I was eager to do, based on their support and favor.

In addition, I would like to thank my family and friends who advocate for me to continue and complete my master's program from a long distance in South Korea. I would especially like to thank Hwasub Lee, who always supported and encouraged me to continue my studies. Ddoljin, Gumjin, and Ultuk as well!

Finally, I want to retrospect my master's program in student life. It was a challenging journey to become familiar with study life after graduating from my bachelor's program in 2009. At the beginning of the study, I was convinced that "*I can do it.*". However, the COVID-19 pandemic made it difficult for me to adapt to different education systems, make friends, and explore the campus since almost all the processes were conducted online, and it turned into a question mark, "*Can I?*". After 2nd semester, I took a year off and took into consideration giving up my studies during that time. However, it has ended up as "*I did it!*" with a period mark. I have learned many things during my studies, especially about myself. Challenges and difficulties have become valuable experiences and knowledge. I will never forget double-loop learning, the trajectory model, Gupta's trade-off model, and diffusion and distribution. Humbly, I want to compliment myself that "*You did your best! You did great!*"

Hyejin Jeong
Gothenburg, Sweden
June 2023

Table of Contents

1.	Introduction	1
1.1	Background	1
1.2	Objectives and Research Questions	2
1.3	Delimitations	2
1.4	Specification of Research Issue.....	3
2.	Theoretical Framework.....	4
2.1	The New Product Development Process.....	4
2.1.1	Project Classification	5
2.1.2	Process Tailoring	6
2.2	Scaled Agile Framework.....	7
2.2.1	Dependency and Prioritization.....	8
2.2.2	Requirement Management and Testing	9
2.3	Alignment of CMMI and SAFe	10
2.4	Organizational Design.....	12
2.4.1	Organizational Structure	13
2.4.2	Formalization	13
2.4.3	Role Ambiguity and Conflict.....	14
2.5	Knowledge Management.....	15
3.	Methodology.....	17
3.1	Research Strategy	17
3.2	Research Design.....	17
3.3	Data Collection and Analysis.....	18
3.3.1	Literature Review.....	18
3.3.2	Organizational Data	18
3.3.3	Survey	18
3.3.4	Interview	18
3.3.5	Self-completion Questionnaire	20
3.3.6	Benchmarking	20
3.4	Research Quality	20
3.4.1	Quantitative Research	20
3.4.2	Qualitative Research	21
3.5	Ethics.....	22
4.	Empirical Findings	23
4.1	The Process-related Challenges	23
4.1.1	Iron Triangle and Definition of Quality.....	23
4.1.2	Perception of the New Product Development Process	25
4.1.3	Estimation and Prioritization	26
4.1.4	Requirement Management.....	26
4.1.5	Testing.....	27
4.1.6	Release	29
4.1.7	Operations	29
4.2	Organization-related Challenges.....	30
4.2.1	Using Informal Terminology	30
4.2.2	Holistic View	31

4.2.3	Roles and Responsibilities	32
4.3	Knowledge Management-related Challenges.....	34
4.4	Evaluation of SAFe Maturity	35
5.	Benchmarking.....	37
5.1	Company X	37
5.2	Company Y	38
6.	Discussion.....	40
6.1	Inflexible Process	40
6.2	Informal Resource Allocation	41
6.3	Lacking Testing Governance	42
6.4	Lacking Roles and Responsibilities	42
6.5	Lacking Knowledge Management	43
6.6	Unbalance between Development and Operations.....	44
6.7	Lacking Customer Involvement.....	45
7.	Recommendation	46
8.	Conclusion.....	47
8.1	Answers to the Research Questions	47
8.2	Limitation of the Research	48
8.3	Recommendation of Future Research	49
	References.....	i
	Appendix 1. Interview Guide.....	iv
	Appendix 2. The Method Coding Transcription.....	xii
	Appendix 3. Self-completion Questionnaire of SAFe Maturity	xiv

List of Tables

Table 1. Capability and Maturity Level (CMMI Product Team, 2010, p.23-29).....	11
Table 2. Interviewee list.....	19
Table 3. The definition of quality	24
Table 4. The knowledge of informal team names based on working experience	31
Table 5. The result of evaluating SAFe Maturity at the case company	36

List of Figures

Figure 1. The 5 th generation stage-gate process (Cooper, 2022)	5
Figure 2. Characteristics of the Staged and the Continuous Representations	11
Figure 3. The SECI model (Nonaka & Takeuchi, 1995).....	16
Figure 4. Process of conducting the research	17
Figure 5. The Iron Triangle.....	23

1. Introduction

Today, technology is evolving rapidly, and many new technologies have emerged to satisfy customer needs and expectations. At the same time, the market has become competitive, and customers want to get innovative new products promptly compared to in the past. Therefore, firms must find ways to expedite new product development (NPD) and increase creativity and flexibility to meet customers' needs, which can directly affect gaining a competitive advantage in the market.

NPD is regarded as a complex process because of its cumbersome characteristics such as many overlapping phases and elements (Ahmed et al., 2021). Despite the complexity of the NPD process, it has become a key factor in business success, because it is closely related to customer satisfaction. Therefore, firms must scrutinize and validate the current NPD process and tailor it to meet their customer needs.

Regarding the importance of the NPD process, changing the way it works from a waterfall method to an agile method has been emphasized to develop fast-launching new products. A waterfall model is considered a serial and traditional way of working when developing new products. However, there is a limitation in supporting the rapid development of NPD by adopting the waterfall model. In the 1990s, agile models such as sequential, planning-driven, and hardware-oriented product development were introduced to overcome the disadvantages of the waterfall model. The agile method is iterative, flexible, and fast compared to the waterfall method.

Despite the advantages of the agile way of working, integrating the existing NPD process and organizational structure becomes another challenge because organizations are used to developing products through the waterfall way of working, which is regarded as bureaucratic, with process gates and excess documentation (Paasivaara et al., 2018). Another challenge is to adopt an agile way of working in large-scale organizations, as it was initially developed for small organizations (Dikert et al., 2016). Therefore, the efficient management of many different functions by adopting the agile method is a core prerequisite for the achievement of NPD (Ahmed et al., 2021).

Given these statements, it can be seen that the NPD process, way of working, and organization are closely related, and these three aspects must be considered simultaneously to support customers' needs and time-to-market (TTM).

1.1 Background

The case company is a global organization within the case company group that leads to the development of its telematics-connected services and solutions. In addition, the case company aims to develop and deliver new services and solutions to TTM with high quality in response to the rapidly changing needs of the case company's business brands. In addition, the business of the case company is expanding the world to ensure

connectivity for products from all the case companies' business brands and to maximize the value of connected assets.

The case company developed new products through an NPD process that has defined its own NPD process consisting of four phases with three milestones. Over time, it has changed from a project-oriented to a product-centric process and has adopted the Scaled Agile Framework (SAFe), and development and operations (DevOps) principles. In the case company, there are three functional divisions involving NPD: DI30000, DI60000, and DI70000. The DI30000 is closely related to research and development (R&D) and long-term service development. DI60000 develops the back-end side and DI70000 develops the front-end side.

Even though all efforts to increase quality and performance are made by tailoring the NPD process, changing the way of working, and restructuring the organization, not all NPD are managed through the NPD process defined at the organizational level and the same tools for some reasons. In addition, the method of working at the case company differs because of the self-organizing teams. Therefore, there may be a potential risk of confusion and misalignment in the cross-functional divisions and teams when developing new products.

1.2 Objectives and Research Questions

The importance of the NPD process, way of working, and organization to meet customers' needs and TTM is presented in Chapter 1. With respect to the background of the case study company, the need to investigate and evaluate the current situation regarding these three aspects is described. Therefore, this study aims to examine and evaluate the current NPD process, way of working, and organization and to analyze the challenges currently confronting the case company from an academic perspective. By doing so, we find potential recommendations to minimize the gaps between the challenges and theoretical framework. Two research questions (RQs) were generated for this study:

- RQ1. What are the challenges faced when developing products through the NPD process, way of working, and organization?
- RQ2. What are the recommendations for overcoming these challenges, based on the theoretical framework?

1.3 Delimitations

As the scope of the research is relatively wide, considering the given time and limited resources, defining delimitations is necessary for the initial phase of the study. Two aspects were defined in terms of what was included. First, even though the targeted organizations are DI30000, DI60000, and DI70000, which develop new products and services, it is inevitable to scrutinize DI20000 and DI50000 because these two functional divisions support NPD as indirect cost functions of architecture, security,

and business planning. Second, because the DevOps framework includes a set of practices from plan, code, build, test, release, deploy, operate, and monitor included in SAFe (Wikipedia.n.d.), it is necessary to include an operational component. However, this study had three delimitations. First, customer perspectives were not investigated due to time limitations. Second, the change management, culture, and leadership aspects were not included in the scope of its wide range of scopes in order to cover topics in this study. Finally, evaluating an organization using the SAFe Maturity Model covers only evaluating practices and does not provide a solution for improving results.

1.4 Specification of Research Issue

Due to the NPD process, way of working, and organization continuously improving during the research study at the case company, data were frozen at the beginning of the research study to avoid misunderstanding and confusion. In addition, there is a likelihood that the empirical findings and recommendations might already be known issues and are in the middle of improvement. Nevertheless, the research findings and recommendations can be used as valid and reliable information for the case company to stand up to ongoing improvement directions and activities. This is because this study is grounded in objective theoretical frameworks that are mutually verified by multiple researchers and academic industries.

2. Theoretical Framework

In this chapter, theoretical frameworks relevant to this research are introduced to gain insights and prior knowledge for this study. This chapter is divided into five parts: NPD process, SAFe, Capability Maturity Model Integration (CMMI), organizational design, and knowledge management.

2.1 The New Product Development Process

The importance of the NPD process has been addressed by many researchers because it is directly relevant to TTM and customer delivery. Although the NPD process is important, it is regarded as cumbersome because different functions are engaged in the process to perform many activities by overlapping phases and elements (Ahmed et al., 2021). Understanding customer needs, targeting markets, and pricing during the NPD process is significant because it directly affects a firm's success or failure (Cooper & Sommer, 2020). Therefore, many researchers have tried to discuss how to manage the NPD process to support TTM, low cost, and higher quality of products by generating different frameworks.

One famous NPD process is the Funnel Model developed. The model supports the idea of integrating the generation and reviewing of alternatives and critical decisions through forums based on senior managers by collecting a wide range of inputs and gradually refining and selecting them to meet customers' needs.

It is addressed that the ideal funnel consists of three phases with two screens by decreasing the screen which can be seen as costs and by enlarging the opportunity to increase the idea generations at the beginning of the NPD process (Wheelwright & Clark, 1992). Regarding the screens, go/no-go evaluation is not conducted in the first screen gate because the aim of the screen is for reviewing completeness or readiness. Additionally, this screen is more about checking for their fitness with technology and product market strategies. In the second screen, senior management reviews products and makes go/no-go decisions that will become development projects. Enlarging the mouth of the funnel can be done in an iterative way by a variety of sources than R&D such as incentives or awards by motivating for engaging and generating ideas to increase the number of inputs (Wheelwright & Clark, 1992).

Another well-known NPD process is Stage-Gate model, developed by Cooper. The initial traditional stage-gate model consists of five stages, and the five stages are to make a decision to stop or continue the process, which is regarded as serial. Although the advantage of the Stage-Gate model is that it provides a holistic view of NPD for senior management to better manage and control, there is a downfall in the process such as process dependency (Cooper, 2016). For instance, if the current stage is delayed, it affects the next stage which means that it will be delayed. In other words, the accumulated delayed time from each stage will draw into late to market at the end of the process due to a "plan-driven model" from a macro perspective (Cooper, 2016).

Since TTM has become important due to the fact that customers expect to get new products as early as possible, these factors make firms to lead put high weight on small projects to accelerate NPD (Cooper, 2022). To support rapid changes in the market, the stage-gate process has also continuously evolved by reflecting market needs and introducing the 5th generation Stage-Gate Model in 2022. The background of the 5th generation process is to differentiate processes based on project classification in order to decrease the waste of time and effort but increase flexibility.

2.1.1 Project Classification

In conjunction with the previous regarding the NPD process, the need for project classification to tailor the NPD process has been emphasized by various researchers. Cooper (2022) highlights that “*Different NPD processes must be created and tailored to suit different types of projects: high risk versus low risk, and minor modifications versus true innovations (p.44)*”.

Additionally, Cooper (2022) guides three types of processes by classifying projects into Stage-Gate Full, Lite, and Xpress processes based on a larger, major, and higher risk (see Figure 1). All three versions of the stage-gate process begin with a discovery stage. The Stage-Gate Full process includes five stages: Scoping, Business case, Development, Testing, and Launch. The Lite process integrates Stage 1 with Stage 2 and Stage 3 with Stage 4. As a result of integration, the Lite process includes three stages: Scoping & Business case, Development & Testing, and Launch phases. The Xpress process is the most simplified process compared to the other two processes, and it entails only two stages: the Scoping & Business case and Execute.

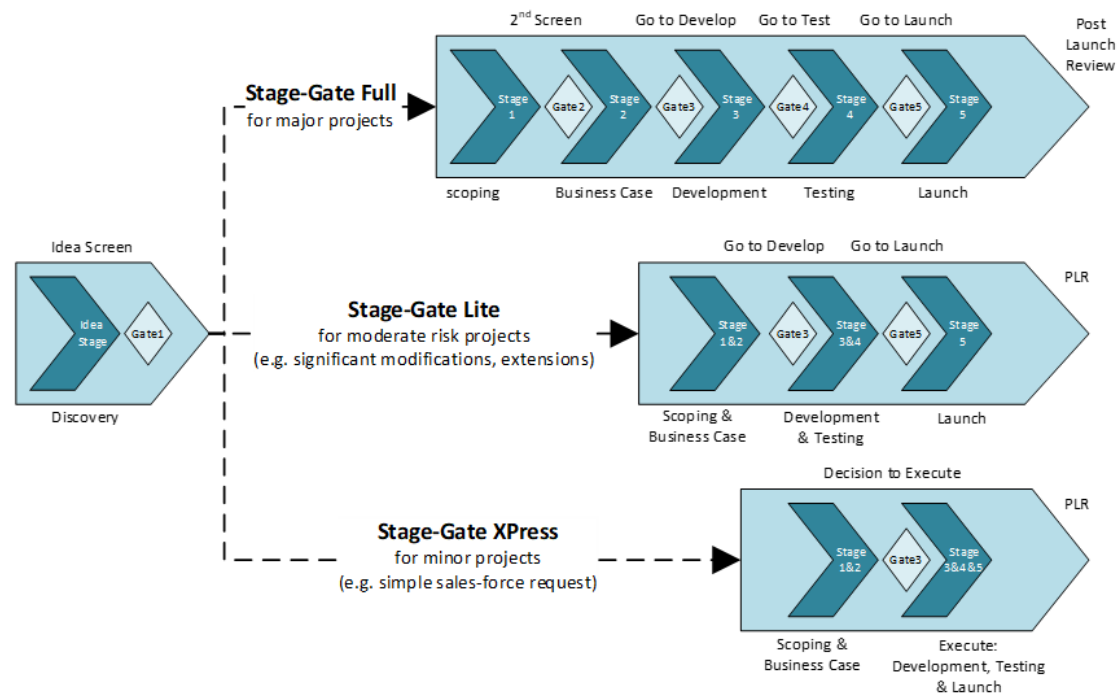


Figure 1. The 5th generation stage-gate process (Cooper, 2022)

An appropriate version of the process can be chosen depending on the type of project and risk assessment. For major projects, the Stage-Gate Full process is regarded as the original stage-gate process. The Stage-Gate Lite process can be used for moderate-risk projects with significant modifications, extensions, or improvements. The Stage-Gate Xpress process can be applied to minor-risk projects where minor modifications are required (Cooper, 2022).

Similar to Cooper (2022), Shenhar (2001) argues that “*One size does not fit all projects*” and that different types of projects can be managed and developed by considering the combination of technological uncertainty and system complexity. Technological uncertainty can be divided into low-, medium-, high-, and super-high-tech, and system complexity can be divided into assembly, system, and array scopes. Therefore, 12 different combinations of projects can be classified and fitted into the NPD process when tailoring (Shenhar, 2001).

On the other hand, De Meyer et al. (2002) suggest a project classification scheme based on the type of uncertainty in conjunction with project-management techniques. Uncertainty can be divided into four major types: variation, foreseen uncertainty, unforeseen uncertainty, and chaos. Each type of uncertainty must be identified and addressed differently in project management. Similarly, Akbar (2019) stated that software projects vary in terms of scope, nature, and complexity. Therefore, software NPD process tailoring is an inevitable practice that needs to be performed in all projects when required.

In summary, Cooper (2022) states that tailoring a process can be weighed on a risk basis; however, the specific criteria of risk are not sufficiently discussed in this article. Meanwhile, Shenhar (2001) and De Meyer et al. (2002) discuss project classification in detail; however, neither state how to tailor the process based on the defined project classification. Akbar (2019) addressed the importance of tailoring the software development process based on guidelines by providing a practical aspect. Given the statements, it can be argued that a project classification scheme should be chosen for process tailoring in terms of different types of schemes. However, one fundamental understanding that can be addressed by integrating the four articles is that the NPD process needs to be tailored differently based on project classification to improve quality and performance.

2.1.2 Process Tailoring

Process tailoring is a technique that adds, merges, and removes practices from a process based on requirements of companies or projects (Akbar, 2019). Because process tailoring affects the entire software project lifecycle, its importance has been addressed (Akbar, 2019). The most common driver of NPD process tailoring is a paradigm change from traditional approaches to agile methodologies (Akbar, 2019). This is because many companies have a tendency to tailor the process to fit the existing process

designed for the waterfall method into the last product development framework (Akbar, 2019).

When tailoring the process, sufficient understanding of the company regarding procedures, standards, operations, and characteristics and requirements of the software project, which can be regarded as a pre-requirement, is addressed by Akbar (2019). Without considering these factors, the existing development process will be re-tailored later (Akbar, 2019). However, many companies tend to perform process tailoring in an ad-hoc manner rather than a regular and well-defined standard approach, as standardized approaches with practical recommendations are not guided (Akbar, 2019).

With regard to the importance of process tailoring, Pedreia et al. (2007) highlight that taking four perspectives into consideration when tailoring process. The first aspect is the project triangle (cost, time, and quality), which can directly affect the quality and performance of the NPD process. The second aspect is that unnecessary activities in the process can lead to a loss of time and cost. The third aspect is improper process tailoring that can drag into the misalignment of either an organizational standard process or international standards, such as the International Organization for Standardization (ISO) or CMMI. The last aspect is taking employee satisfaction into account because employees who use a process do not want to waste time by performing unnecessary activities.

In contrast, Xu and Ramesh (2007) argue that it is essential to understand various environmental factors that can affect the application of the tailoring process. The various environmental factors were classified into four categories: project (size, budget, schedule, and complexity), team (size, experience, and cohesion), external stakeholders (number of stakeholders), and organization (policy, process maturity, technical support, and senior management commitment).

2.2 Scaled Agile Framework

The agile model embraces an iterative process with incremental ways of working, customer collaboration, high flexibility, productivity, and efficiency. In addition, it allows product development to change requirements, collaborate with customers, and cooperate with different functions that the waterfall does not support (agilemanifesto.n.d.). Since the agile model was initially designed for relatively small and single-team projects, the need for agile models available to large organizations has increased; therefore, various types of scaled large agile models have been developed.

SAFe is one of scaled large agile models that implements agile practices for product development at an enterprise scale. SAFe embraces the principles and practices of leanness and agility and helps achieve business agility (SAFe.n.d., 2023). SAFe includes four primary bodies of knowledge: agile development, lean product development, systems thinking, and DevOps. Furthermore, it introduces four core

values that make SAFe effective based on alignment, built-in quality, transparency, and program execution (SAFe.n.d.a., 2023).

Although large-scale agile methods have been developed for larger firms, one factor that defines the meaning of a large scale has become an arguable topic by many researchers because of the likelihood of inappropriate adoption. Dikert et al. (2016) defined the term "large-scale" as organizations employing fifty or more people or operating at least six teams, where six to seven people work together on average. Eckstein (2014) does not distinctly define the term "large-scale" distinctly, however, it can be inferred based on the definition of "small agile team, " which has no dedicated role or clear responsibility for the architect, developer, and tester. The definition of "large scale" can be arguable depending on perspective, although it is obvious that many more complex dependencies within product development organizations will be expected in large-scale agile organizations than in small agile organizations (Dikert et al., 2016).

2.2.1 Dependency and Prioritization

In Berntzen et al. (2021)'s article, dependency is defined as when the progress of one task is affected by the output of other tasks or the presence of a specific resource such as an artifact, a person, or relevant information. For instance, dependency between teams can be created when the output from one team's activity is used by another team as an input. Furthermore, the complexity of dependency increases linearly, depending on the size of the scale agile (Berntzen et al., 2021).

Dependencies on agile projects can be categorized into three types: knowledge, resources, and process dependencies (Berntzen et al., 2021). Knowledge dependency occurs when information is required for the progress of a project. For instance, requirements, expertise, history, and resource/task allocation can be considered information in knowledge dependency. Process dependency is a type of dependency easily observed in the linear development process. Process dependency occurs if a task cannot proceed until the previous task has been completed, where the task can be either an activity or business process. Resource dependency occurs when there is dependency between persons, places, or things required for a project to progress.

According to Berntzen et al. (2021), dependency, priority, and estimation are closely related to each other. It is explained that estimation is an important factor in determining priority because incorrect estimation affects priority. Then, these established priorities affect dependence on resources and processes. Thus, while the importance of dependency management and task estimation and priorities can also be important factors, many organizations face problems early on because of a lack of standardization that affects estimation issues (Screenivasan & Kothandaraman, 2019). To make the correct estimations that can affect prioritization and dependency, Screenivasan and Kothandaraman (2019) address that estimation can be managed by adopting CMMI and then achieving a high level of CMMI.

2.2.2 Requirement Management and Testing

Requirement management is a set of techniques for collecting, analyzing, and requiring breakdown, and verifying and validating requirements for the given product being developed. Successful requirements management ensures that not only completed deliverables meet customers' needs, but also well-refined requirements support to mitigate potential errors in the late phase of the NPD process (Baumgartner et al., 2022). In addition, requirement management provides several benefits, such as high product quality and traceability (Neto et al., 2017).

Another major benefit of requirement management is that it is tied to early risk management, which can be known as fuzzy front end (FFE) or front-end homework. The meaning of FFE is defined by Reid and de Brentani (2004), as all activities and ideas should be discussed prior to the first official meeting, since the late discussion of potential risks can drag into failed products. Similar to Reid and de Brentani (2004), Cooper (2019) argues for the importance of conducting front-end homework due to a higher likelihood of product failure and slowing down development. This is because missing integrated perspectives that are relevant to developing new products in the early phase of product development increase significant risks and costs to solve.

According to Baumgartner et al. (2021), potential defects can be identified in the early phase and eliminated in advance if testers cooperate with developers and review the requirements from the initial stage of product development. In addition, Tengstrand et al. (2021) stated that the alignment between requirements and test cases can lower the requirement risk, increase the coverage of requirements, and minimize verification costs.

Testing is a technique or strategy for an organization to improve its development quality and it is critical to establish a clear testing strategy at the organization/company level to increase the quality of products (Barraood et al., 2022). In general, test approaches such as test objects, methodologies, environments, and tools can be formalized and clearly defined in the testing strategy, and high-quality test specifications can be developed based on them and distributed to all teams in the organization (Barraood et al., 2022). Besides establishing the testing strategy to improve product quality, it is addressed the importance of having good-quality test cases and securing sufficient time to perform test activities because these can provide a high probability to detect hidden defects with minimum effort and cost (Barraood et al., 2022).

Despite the importance of the writing of test case documents, it has become an arguable topic due to a misconception of the agility of being free from documentation, (Dikert et al., 2016). Additionally, every developer tends to be free to decide what and how much to test, and even test cases are not written by them (Baumgartner et al., 2021). In addition, documentation and testing should begin from the early phase of development to secure sufficient time to perform testing because late testing leads to late defect detection and time constraints to fix them (Baumgartner et al., 2021).

However, there is a challenge between theory and reality, especially in large-scale agile projects, owing to complications of implementation and tight deadlines (Baumgartner et al., 2021). Therefore, the importance of an automated testing method is addressed in large-scale agile projects to maximize efficiency by minimizing the quantity of manual work (Baumgartner et al., 2021).

How to govern testers is another topic of debate. According to Baumgartner et al. (2021), the testing group needs to be separated for larger projects to supervise several projects independently. However, testers should work together with developers in a parallel manner to reduce the time span handover from developers to testers and to secure time to develop a test plan and specify test cases for testers (Baumgartner et al., 2021). In addition, it is recommended that a dedicated tester seats together with developers because it provides easy communication and coordination skills with developers in an agile way of working (Baumgartner et al., 2021).

The role of testers can be differentiated between that of generalists and specialists (Baumgartner et al., 2021). Generalists are fixed team members who can perform a wide range of test activities; however, they do not possess in-depth knowledge of specific areas. On the other hand, specialists have knowledge of either how to set up and adapt test frameworks or perform non-functional tests, such as performance, security, recovery, or usability, and they can act as supporters who work from the outside of teams. Baumgartner et al. (2021) argued that small projects can be tested by one or two generalists, which brings many benefits to the team. However, it is argued that both generalists and specialists should be engaged in larger projects.

2.3 Alignment of CMMI and SAFe

CMMI is a full set of best practices for process improvement that provides the product life cycle from conception to delivery and maintenance (CMMI Product Team, 2010). CMMI supports organizations in improving productivity, reducing defects, optimizing processes, and ensuring predictable operations (Sreenivasan & Kothandaraman, 2019).

There are two types of representations in CMMI model in version 1.3: Staged and Continuous (see Figure 2 and Table 1). In staged representation, a pre-defined set of process areas is used to define an improvement path for organizations, and the overall maturity for practice areas (PAs) is evaluated by pre-defined five maturity levels (MLs). Once all process areas reach a specific maturity level that has to be achieved, they can move forward to the next maturity level. This representation consists of five stages: Initial, Managed, Defined, Quantitatively Managed, and Optimizing.

On the other hand, continuous representation allows organizations to select a specific process area where they are targeting to evaluate, improve, and utilize pre-defined Capability Levels. This provides the basis for continuous improvement at each level of the corresponding process, enabling cross-organizational comparisons of specific

processes throughout the CMMI process. The continuous representation capability levels consist of four different levels: Incomplete, Performed, Managed, and Defined which are in line with levels 0 to 3, respectively.

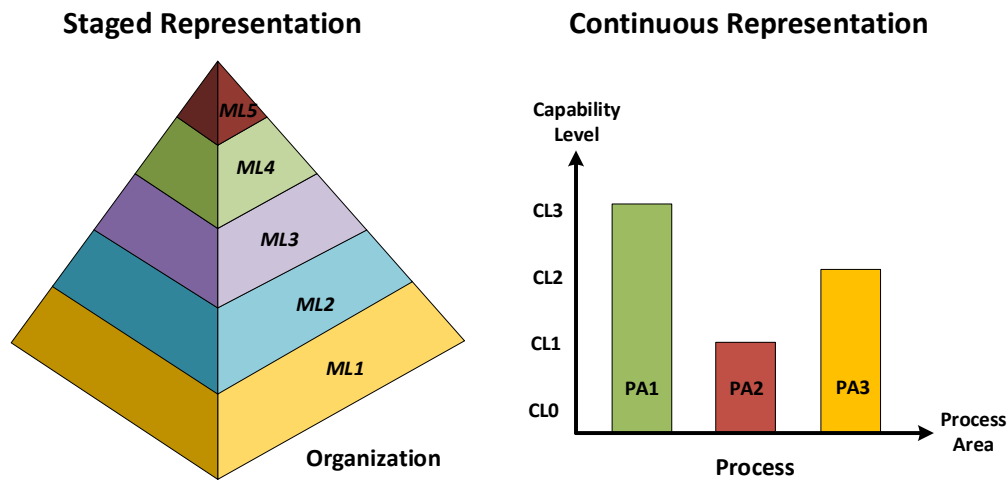


Figure 2. Characteristics of the Staged and the Continuous Representations

Table 1. Capability and Maturity Level (CMMI Product Team, 2010, p.23-29)

Level	Maturity Levels (ML)	Capability Levels (CL)
Lv. 0	-	Incomplete: A process that either is not performed or is partially performed.
Lv. 1	Initial: Processes are usually ad hoc and chaotic.	Performed: A process that accomplishes the needed work to produce work products.
Lv. 2	Managed: The projects have ensured that processes are planned and executed in accordance with policy.	Managed: A performed process that is planned and executed in accordance with policy.
Lv. 3	Defined: Processes are well characterized and understood, and are described in standards, procedures, tools, and methods.	Defined: A managed process that is a tailored from the organization's set of standard processes according to the organization's tailoring guidelines.
Lv. 4	Quantitatively Managed: The organization and projects establish quantitative objectives for quality and process performance and use them as criteria in managing projects.	-
Lv. 5	Optimizing: An organization continually improves its processes based on a quantitative understanding of its business objectives and performance needs.	-

Along with the evolution of CMMI, Agile practices are also adopted in various industries due to the fact that both of these are useful for software and system product development (Sreenivasan & Kothandaraman, 2019). Although there are several advantages of the agile method, there are certain limitations, especially when adopted in a large-scale organization due to a lack of structured guidelines and insufficient implementation strategies or methods (Turetken et al., 2017). This is because the agile method focuses on software development projects without considering other general software engineering issues such as scaling, training, organizational policy, and measurement into account (Ferdinansyah & Purwandari, 2021). To overcome these challenges, it is pervasive to combine CMMI and Agile models. According to Henriquez et al. (2021), 95% of organizations are applying agile practices, and at least 80% of CMMI practices are jointly adopt agile practices.

In the article by Turetken et al. (2017), approximately 40 maturity models for agile development were identified from a systematic review of the literature; however, none of them have referred to the practices, principles, and challenges of large-scale agile settings to measure large-scale agile (Turetken et al., 2017). Therefore, Turetken et al. (2017) introduced SAFe Maturity Model (SAFe MM) as a basis to assess the current maturity level of agile adoption. It is addressed that SAFe MM helps organizations not only define the roadmap for SAFe adoption, but also implement and improve SAFe practices in organizations (Turetken et al., 2017). In addition, SAFe provides guidance on “*how*”, on the other hand, CMMI provides “*what*” (Sreenivasan & Kothandaraman, 2019). Therefore, the two concepts do not conflict with each other, and even organizations can achieve synergies by aligning CMMI and SAFe environments (Henriquez et al., 2021).

CMMI version 2.0 was published in 2018 after version 1.3. The difference between version 1.3 and version 2.0 is that PAs are used in place of the process areas due to the fact that CMMI is a sort of collection of practices, not processes compared to CMMI version 1.3 (Wikipedia.n.d.a.). According to Henriquez et al. (2021), 35 literature reviews combining “Agile” and “CMMI version 2.0” are retrieved, however, these literatures do not provide specific guidelines on how to establish CMMI practices and Agile artifacts. To map version 1.3 and version 2.0 due to different structures, Henriquez et al. (2021) analyzed the two versions and mapped out version 2.0 and Agile artifacts. The study indicates that the majority of practices cover agile artifacts; however, seven PAs do not cover any agile artifacts (Henriquez et al., 2021).

2.4 Organizational Design

This chapter presents topics related to organizational design, which are organizational structure, formalization, role ambiguity and conflict, and ambidexterity.

2.4.1 Organizational Structure

There are many different elements to structure an organization, such as centralization, decentralization, formalization, standardization, and ambidexterity. Depending on the arrangement and configuration of these elements, two types of organizational structures can be defined: mechanistic and organic.

The organic structure is flexible and decentralized, and it is usually suitable for organizations that require high flexibility and adaptable structures (Daft, 2015). For instance, autonomy, free flow of information, less formalization and standardization, wide spans of control, innovative exploration, and cross-functional/hierarchical teams are examples of organic structures. Therefore, the organic structure encourages creativity and experimentation, while low consistency and reliability can be achieved.

In contrast to the organic structure, the mechanistic structure is centralized in a top-down mandate that has the locus authority to make decisions that affect the organization (Daft, 2015). In addition, the structure enables formalization, standardization, exploitation, and distribution. Therefore, the mechanistic structure can be beneficial in terms of operational efficiency, reliability, and minimization of variation; however, it may stifle creativity and experimentation.

Based on the concepts of the two structures, centralization and decentralization seem incompatible because of opposing concepts. However, the two concepts can be compatible and even achieve benefits. The concept is known as “ambidexterity”. Ambidexterity can be viewed as acting in both exploitation and exploration. Exploitation is a top-down mandate and more about efficiency, increasing productivity, control, and certainty. On the other hand, exploration can be regarded as a bottom-up approach, and it is about discovery, autonomy, innovation, and embracing variation (O’Reilly & Tushman, 2008).

Despite the advantages of ambidexterity structures, it is essential to achieve a trade-off between exploitation and exploration in order to attain capability and ability in dynamic environments. For instance, if organizations focus only on exploration without taking exploitation into account, it leads to investing in experimental innovation with a lack of development capability required for implementation and institutionalization (Ramesh et al., 2012). By contrast, concentrating only on exploitation without considering exploration drags organizations into an inability to catch up with innovations (Ramesh et al., 2012).

2.4.2 Formalization

Formalization has often been regarded as “red tape” due to the fact that it limits the ability to exercise authority or hinders effective performance, however, it has also been considered necessary to provide strategic direction and goal clarity (Hempel et al., 2012). Formalization provides a clear description of tasks and jobs, official and explicit responsibilities of divisions, and ways of working. In addition, it decreases ambiguity

in decentralization and increases opportunities to achieve goals. Therefore, leadership can be secured, and effective decision-making can eventually be achieved. In addition, formalization supports easier training for newcomers because it provides written documents to newcomers and the same opportunity to access the same information, independent of the network (Tolbert & Hall, 2019).

It is explained that formalization can be applied to two different organizational domains: organizational formalization and job formalization (Hempel et al., 2012). Organizational formalization explains the operational boundary between different teams and specifies the relationships within the organization. Even though decentralization can assist with autonomy in organizations, it only works when an organizational boundary is presented, and the team's direction is specified. Under these circumstances, formalization can provide consistent decision-making and aid in preventing authorization erosion (Hempel et al., 2012).

Job formalization provides clear roles within teams. Job formalization has two distinct aspects: job codification and job observation. Job codification refers to written rules, the degree of standardization of work, and a measure of how many rules define; on the other hand, job observation refers to whether the written rules are followed or not (Tolbert & Hall, 2009). A lack of codified jobs can facilitate the decentralization of power, which can be regarded as autonomy. In other words, less-formal job codification accelerates decentralization and autonomy (Hempel et al., 2012).

2.4.3 Role Ambiguity and Conflict

Organizational socialization is employees' interest in how to be addressed by an organization and how to be reflected in their interests, and it supports meeting the needs of both the organization and the employees (Judeh, 2011). When newcomers join a new organization, they try to adjust their behaviors to fit in the organization by acquiring new information and norms and clarifying roles and responsibilities to increase their emotional attachment to the organization (Judeh, 2011). However, if role ambiguity and conflict prevail in an organization, an unsuccessful organizational socialization case can exist.

Role ambiguity is defined as a lack of clarity in the roles that employees have to fulfill their expected duties, and it is considered that an individual is not instructed with a clear direction in the organization (Judeh, 2011). Role ambiguity causes individuals to be afraid of performing or taking responsibility for anything, and employees become demotivated or unproductive because of confusion (Judeh 2011). It is explained that role ambiguity can be derived from acquiring inadequate information or knowledge, such as inadequate training and poor communication (Judeh 2011).

Role conflict arises when there is disagreement regarding actions to be taken to achieve goals between at least two more parties, such as individuals, groups, or organizations (Tolbert & Hall, 2019). For instance, the combination of conflicts can vary, such as

person-person, person-group, person-organization, group-group, and group-organization. In the case of person-organizational or group-organizational conflict, the conflict between them tends to develop into organizational-organizational conflict, and it can be resolved or mitigated by third-party engagement (Tolbert & Hall, 2019). It is explained that role conflict of a low or moderate degree can be acceptable; however, an organization can become ineffective if the degree of role conflict exceeds an acceptable point (Judeh, 2011).

2.5 Knowledge Management

Modern society is flooded with a lot of information amid rapid technological developments such as artificial intelligence, big data, and analytics; however, it has become a task to make that much information meaningful from the gathered information (McMillan & Ihrig, 2015). McMillan and Ihrig (2015) describe the challenges of transforming big data into useful knowledge; however, it is also explained that many companies tend to satisfy with the fact they are just possessing big data and neglect the importance of information transformation which can largely affect companies' success.

To be successful with big data, McMillan and Ihrig (2015) address the importance of managing and codifying explicit knowledge from tacit knowledge. Tacit knowledge is a more intuitive understanding that can be difficult to integrate based on experience, whereas explicit knowledge, which can be seen as codified knowledge, makes it easier to communicate because it provides a common language and conceptual framework. Therefore, the codified knowledge in documents is more easily shared and diffused.

McMillan and Ihrig (2015) identified a generic engineering company's knowledge assets divided into (un)diffused and (un)structured. Undiffused knowledge tends to be inside the company, and it goes outside the company when knowledge becomes diffused. For instance, lead engineers' tacit knowledge is undiffused with unstructured knowledge. On the other hand, company product specifications published on a website can be viewed as diffused and structured knowledge. By doing so, readers can easily access the website and gain relevant knowledge regarding the product.

On the other hand, Nonaka and Takeuchi (1995) define a SECI model which consists of socialization, externalization, combination, and internalization phases based on tacit and explicit knowledge (see Figure 3). In the socialization phase, tacit knowledge can be transferred into tacit knowledge by providing workshops or networking activities that can be observed. In the externalization phase, tacit knowledge can be changed into explicit knowledge by codifying tacit knowledge documents. Once explicit knowledge is codified, it can be transferred into explicit knowledge, which is in line with the combination phase, and there is a likelihood that shares with different by combining a variety of explicit knowledge. In the internalization phase, explicit knowledge can enter tacit knowledge, and the phases can become routine. Nonaka and Takeuchi (1995) note

that knowledge management looks like a spiral model since knowledge management is not one-time and it is a rather continuous way to accumulate knowledge through these phases.

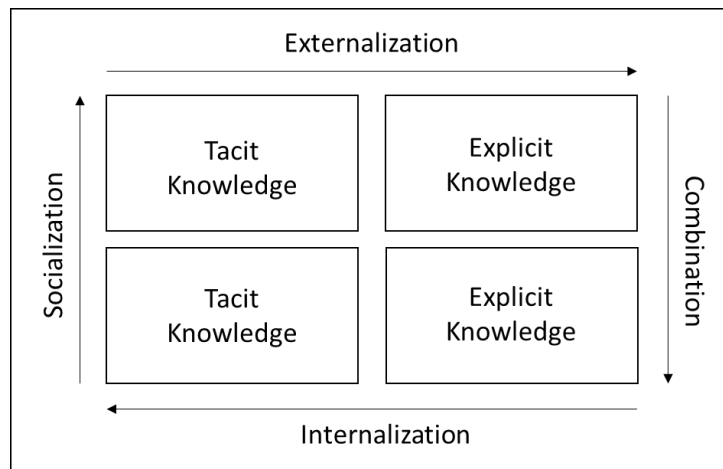


Figure 3. The SECI model (Nonaka & Takeuchi, 1995)

Given the arguments of the researchers, big data that are scattered from different areas can be managed by codifying documents, and the documents can be transformed into knowledge at the end. In addition, it can be distributed to individuals and organizations as tacit knowledge based on documents. In doing so, codified documents can help avoid confusion at various organizational levels (Nyfjord et al., 2014).

As mentioned in Subsection 2.4.3, it is explained that organizational socialization is important for organizations because it affects not only employees' satisfaction but also organizational success (Judeh, 2011). Codified documents are especially useful for newcomers because they have the same opportunity to access explicit knowledge and become tacit knowledge, which is in line with internalization in the SECI model compared to existing employees (Judeh, 2011). Based on this content, knowledge management is important in itself; however, the importance of knowledge management cannot be overemphasized because it can be confirmed that it affects the formation and growth of individuals and organizations for newcomers.

3. Methodology

The methodology chapter describes how this study was conducted. First, the research strategy and design are explained to provide an overview of this study. Data collection and analysis are then explained using specific qualitative and quantitative methods. Finally, the research quality and ethics are presented.

3.1 Research Strategy

This study adopted a mixed research strategy combining quantitative and qualitative approaches. According to Bell et al. (2019), mixed methods research provides an advantage in that quantitative research can fill in the gaps in a qualitative study in terms of time. To apply the benefit of the mixed approach to conducting this research, qualitative methods which are interviews and literature review were set as the primary input. Quantitative methods which are survey and self-completion questionnaire, were used as secondary data to support qualitative data.

3.2 Research Design

The study was designed by selecting a case study because the research was limited to a single company, which is the case company, and it focuses on analyzing a single case in a detailed and intensive way (Bell et al., 2019). A case study is known as one of the best studies in the business and management research approach, and it is frequently used in quantitative, qualitative, and mixed methods (Bell et al., 2019). Therefore, adopting a case study as a research design is suitable for this research study. A triple diamond process was adopted to proceed with the research study, which consisted of three activities in six phases, as presented in Figure 4.

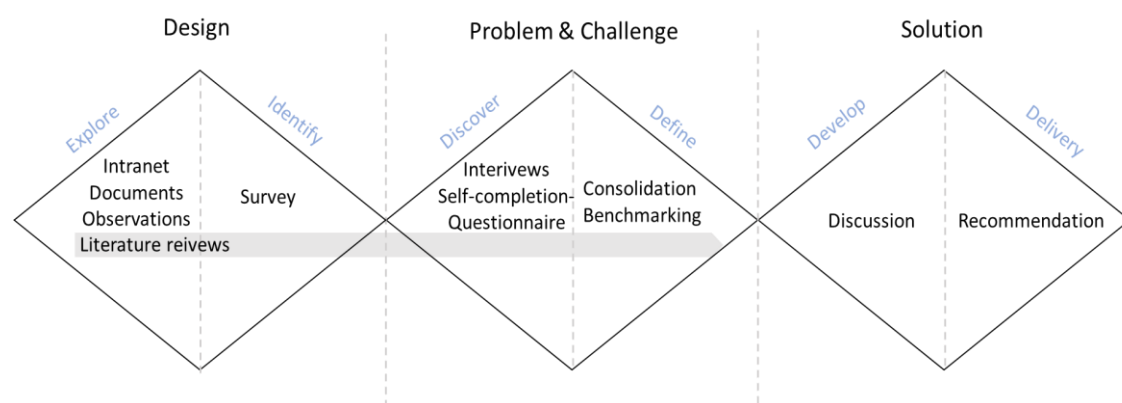


Figure 4. Process of conducting the research

The advantage of the triple diamond process is that it provides additional data discovery steps in addition to the problems and resolution steps provided by the double diamond model (Schleith & Tsar, 2022). The double diamond model identifies "what" in the problem and challenge phases, and the solution phase focuses on "how" (Schleith & Tsar, 2022). However, these two phases alone do not provide data related to end-user constraints and opportunities, making it difficult to understand user insights and

detailed problem steps; however, the triple diamond process compensates for this deficiency (Schleith & Tsar, 2022).

3.3 Data Collection and Analysis

In this sub-chapter, data collection and analysis are presented.

3.3.1 Literature Review

A literature review was performed using Google Scholar and the Chalmers Library. The literature review was conducted by scrutinizing different types of articles based on several keywords in an iterative way to not only gain prior knowledge, but also suggest recommendations based on gap analysis between academic theories and empirical findings. The topic of the literature review was divided into five parts: NPD process, SAFe, organizational design, CMMI and SAFe maturity model, and knowledge management.

3.3.2 Organizational Data

Organizational data were explored through the intranet, documentation, and observation of meetings and forums. As an observer who did not have information and knowledge regarding the case company, it provided ideas on how to formulate and design the research study and what to focus on.

3.3.3 Survey

The purpose of the survey is to identify the target scope of employees engaged in the NPD process. Furthermore, respondents' perceptions regarding the NPD process, the meaning of quality, and the project triangle were examined. The survey consisted of fifteen questions, including basic respondents' information such as working experience, job title, department, and so on. There were eight multiple-choice questions and seven subjective questions, and the survey was sent to the case company employees at the beginning of the study through Microsoft Forms.

In the survey, a cluster sampling method was adopted because it is beneficial for researchers to identify and define patterns based on data element analyses (Bell et al., 2019). The population comprised functional divisions, namely, DI30000, DI60000, and DI70000. The survey was sent out and received 72 responses out of 268 (27%). The survey responses were collected and analyzed anonymously, and the subjective answers were analyzed and categorized to increase an easier and better understanding of the results.

3.3.4 Interview

Interviews were performed after obtaining the basic information required for an effective interview from the survey. Interviews were conducted by adopting a semi-structured interview in which respondents answered based on pre-defined questions; however, it still allows asking questions that can explore other aspects, not pre-defined

questions. The benefits of the semi-structured format include flexibility and reliability (Bell et al., 2019).

Regarding the sampling method for the interview, a stratified sampling method was chosen because many interviewees were working in different roles, in different divisions, and engaged in different phases of the process. According to Bell et al. (2019), a stratified sampling method ensures that results can be seen from a distributed and different perspective regarding the topic. Therefore, this sampling method was considered suitable for this study.

An e-mail had been sent to a total of 27 interviewees from a different range of roles and divisions in order to propose interviews, and 18 respondents (66%) answered that became an interviewee (see Table 2). Due to the fact that interviewees are engaged in different phases of the development process and have different views and knowledge of it, 11 sets of interview questions were formulated for different roles to be able to proceed (see Appendix 1). The interviews were conducted via Microsoft Teams online and took approximately 50 minutes on average; however, they extended depending on the interviewees' willingness to answer more questions.

Table 2. Interviewee list

No.	Job Title	Division	Date	Minute
1	Product Manager	DI50000	2023-03-10	50
2	Quality Assurance	DI50000	2023-03-13	60
3	Project Manager	DI50000	2023-03-13	55
4	Business Analyst	DI60000	2023-03-13	55
5	Business Partner	DI50000	2023-03-15	45
6	Architect	DI20000	2023-03-15	40
7	Project Manager	DI50000	2023-03-17	60
8	Test Manager	DI60000	2023-03-17	60
9	Project Manager	DI50000	2023-03-20	70
10	Release Training Engineer	DI60000	2023-03-20	65
11	Release Training Engineer	DI60000	2023-03-22	55
12	Operations Manager	DI60000	2023-03-22	65
13	Business Partner	DI50000	2023-03-23	50
14	Release Training Engineer	DI70000	2023-03-24	45
15	Release Manager	DI50000	2023-03-24	50
16	Test Manager	DI70000	2023-03-24	65
17	Product Owner	DI60000	2023-03-27	60
18	Service Designer	DI30000	2023-04-03	100

At the beginning of the interviews, brief information such as background, purpose, and expected outcomes regarding the thesis study were informed to minimize deviation among interviewees by sharing the same understanding. In addition, the NPD process and relevant information were presented on a PowerPoint slide to avoid

misunderstanding of the target process. Each interview was recorded to analyze and categorize possible challenges based on code transcription.

After completing the interviews, the challenges addressed by each interviewee were first coded into Excel, and then all merged challenges were grouped into several categories. Then, every keyword based on Excel was copied into Mural, which is a collaborative workplace tool to analyze the relationship between challenges. Based on identifying and categorizing the challenges in Mural, the relationship between each category and the root causes of the challenges were identified (see Appendix 2).

3.3.5 Self-completion Questionnaire

A self-completion questionnaire was conducted to measure the level of SAFe maturity at the case company by adopting SAFe MM developed by Turetken et al. (2017). The self-completion questionnaire consisted of 5-Likert scales with 60 SAFe practices in an Excel file form (see Appendix 3).

The sampling of the self-completion questionnaire was chosen as the clustering sampling method, similar to that used in the previous survey, and it was sent out to 35 employees who were working as product owners and scrum masters. The response rate was 14% (5 out of 35), and the results were analyzed at the organizational level by calculating the mean value of the 5 Likert scaling per SAFe practice. As a result of processing answers, the result was ranked in 4 levels based on the interval range criteria (Sengel & Aktas, 2022): not achieved (1.00 - 1.75), partially achieved (1.75 – 2.50), largely achieved (2.51 – 3.25), and fully achieved (3.26 - 4.00).

3.3.6 Benchmarking

Two benchmarks were performed by two companies. The purpose of the benchmark is to investigate how other companies organize and manage the problems that the case company currently faces. Each interviewee was a project manager, and the other was an RTE, and each interview lasted about 60 minutes. The interview method was the same as that of the case company interview method; however, the interview questions were different (see Appendix 1). Interviews were not recorded because of the preferences of external companies and interviewees.

3.4 Research Quality

To increase the quality of the research study, quality criteria for both quantitative and qualitative research were considered.

3.4.1 Quantitative Research

According to Bell et al. (2019), the quality of quantitative research can be evaluated based on three criteria: validity, reliability, and replicability. Since SAFe MM adopted as the SAFe maturity model in this research study has already validated its quality in terms of validity and reliability by Turetken et al. (2017), additional validation activities

for SAFe MM were not taken into account when conducting this research. When it comes to replicability, the method of analyzing research results must be stated to yield the same results with different input data. However, because Turetken et al. (2017) did not explain how to analyze and evaluate results in a structured format, in this research, the interval range criteria introduced by Sengel and Aktas (2022) were adopted to support it.

3.4.2 Qualitative Research

Bell et al. (2019) described that the quality of qualitative research can be evaluated using four classic criteria: credibility, transferability, dependability, and confirmability.

Credibility ensures that research processes and findings are reliable, and that research is conducted in good practice (Bell et al., 2019). To increase the credibility of the study, a pilot interview was conducted with two employees prior to the official interview. In doing so, it was confirmed that the interview questions were written in an appropriate and understandable way after the feedback was revised. In addition, interview questions were sent to the interviewees in advance so that they could have sufficient time to understand the questions and prepare for answers. By doing so, interviewees had the time to read the interview questions and could prepare what they wanted to say. If unclear and possibly misleading content was discovered during or even after the interview, the interviewee was contacted to recheck and reconfirm.

Transferability refers to the degree to which research findings can be applied or replicated in other studies or not (Bell et al., 2019). In this study, the research subjects were the product development process adopting the agile methodology and the organization managing that process. While conducting benchmarking of the two global companies applying the agile methodology, it was found that there are many similarities from the process point of view because they also adopt well-known frameworks and tailor their process based on either academic theories or internal/external standards. Therefore, it is predicted that the results of this study can be transferred to other similar-sized companies (organizations) adopting an agile methodology for product development.

Dependability can be explained by the activities of the records of the research processes and findings. In this study, all the results of the interviews were documented and stored, and the processes and results were frequently shared with university and industrial supervisors. In the case of the company, a regular weekly sync meeting was held to obtain questions and answers. In the case of the university, a biweekly meeting was held to discuss the research strategy, design, methods, and findings. Through regular meetings from both sides, third-party perspectives, which can be seen as objective feedback, were included in the study. Furthermore, most of the research activities were conducted repetitively to identify similarities in the results and support the findings.

Confirmability refers to whether the value or experience of the researcher is reflected in conducting the research or leads the researcher to derive results in the desired direction. To increase confirmability, research has been conducted based on theoretical frameworks and academic theories rather than on the experiences, assumptions, or biases of the researcher. By doing this, we intended to secure objectivity rather than subjectivity in this study.

3.5 Ethics

Ethics are handled in four aspects based on a recommendation by Bell et al. (2019): avoidance of harm, informed consent, protection of privacy through confidentiality, and prevention of deception.

As a student at the Chalmers University of Technology, avoidance of harm was strictly followed. It is described that physical harm or stress to the participants should be avoided (Bell et al., 2019). Therefore, before starting an interview, it was informed that the interviewee has the right not to answer specific questions or stop the interview if he or she could feel uncomfortable or stressed. In addition, it was also known that the interviewers had the right to turn off their cameras during the interviews because the interviewees were recorded.

When proposing to interviewees through e-mail, sufficient information about the purpose of the interview, duration, method, and ethical aspects was written (Bell et al., 2019). Once the interviewees accepted the interview proposals, an interview invitation was sent. With regard to recording interviews, consent for recording interviews was confirmed by each interviewee and reconfirmed before starting the interviews.

A non-disclosure agreement with the case company was signed before starting the research study for the case company to protect privacy through confidentiality. All data collected during the research, including interview recordings, were stored only on the internal shared repository at the case company to avoid leakage of private or confidential information. To protect the privacy of interviewees, data collection and analysis were conducted anonymously. It avoids mentioning specific persons or business brands that can be used to infer who the interviewee is. It does not include off-record content and does not affect research results.

For deception prevention, questions were sent in advance to provide sufficient time to look through all questions before the interviews. Additionally, the interviewees were encouraged to sort out before moving to the next step if anything was unclear on both sides during the interviews.

4. Empirical Findings

In this chapter, the empirical findings evaluated on the NPD process and the case company organization are presented. The findings were categorized into process-related, organizational design-related, and knowledge-related challenges. In addition, the results of evaluating SAFe MM are also explained. This chapter is correlated with research question 1 which is “*What are the challenges faced when developing products through the NPD process, way of working, and organization?*”.

4.1 The Process-related Challenges

In this sub-chapter, challenges regarding the NPD process, called the NPD process in the case company, are explained. The main findings are related to employees' perception of the NPD process, estimation and prioritization of projects, requirement management, testing, release, and operations.

4.1.1 Iron Triangle and Definition of Quality

The perspective of the Iron Triangle was investigated through the survey in order to identify what are the most important factors among "time & quality", "cost & quality", and "time & cost" when developing products from the employees' and customers' sides. In the case of customers' perspectives, a survey question asked the employees what factors the customers considered the most important, since there is a limit to contacting the customer directly.

The result showed that 74% of the employees perceive "time & quality" are the most important factors, and "cost & quality" was recognized by 26% of them as significant factors, however, "time & cost" was chosen by none. Meanwhile, the result of the customer perspective showed that "time & quality", "cost & quality", and "time & cost" were answered by 41%, 37%, and 22%, respectively. Given the results, "time & quality" can be considered the most significant set of constraints when developing products for both employees and customers (see Figure 5).

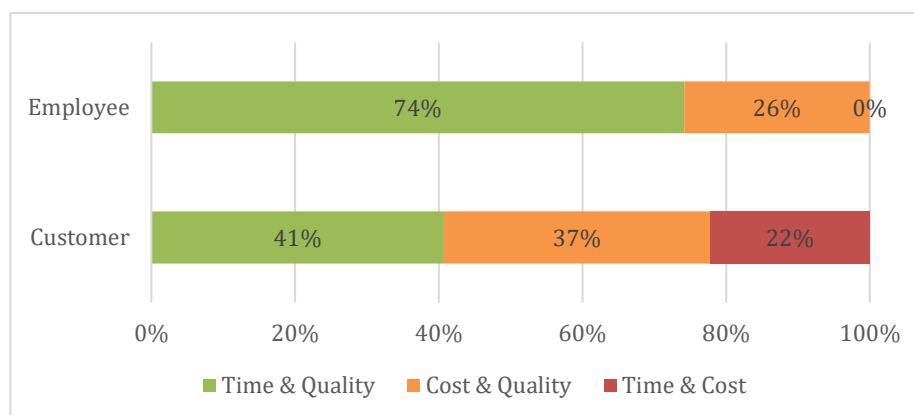


Figure 5. The Iron Triangle

Although the most important factors were identified “time & quality” and the importance of quality is emphasized during Program Increment (PI) planning sessions by top-management; however, this does not provide the clear meaning of quality. Since the definition of quality can vary based on individuals’ perspectives and the meaning can be broad, a subjective question on employees’ definitions of quality was asked. 26 (90%) out of 29 respondents described the definition of quality as the characteristics of software quality standard, ISO 25010¹ which are “maintainability²”, “reliability³”, “functionality⁴”, “usability⁵”, and “efficiency⁶” shall be valued when improving the quality of software, see Table 3 (Wikipedia.n.d.b.). Meanwhile, the remains of 3 (10%) described that quality is achieved by customers’ success needs.

Table 3. The definition of quality

Attribute	Characteristic	Number of Respondent	Sub-characteristics
ISO 25010	Maintainability	7*	Maintenance (3), robustness (3), scalability (2), testability (2), adaptability (1), measure (1), resilience (1)
	Reliability	7	Reliability (4), availability (2), fault tolerance (1),
	Functionality	6	Functionality (5), agreement (1)
	Usability	5	Less error/bug (3), incident (1), user friendly (1)
	Efficiency	1	Performance (1)
	Total	26	-
Customers' Success	Customers' success	3	-
	Total	3	-

* Total differs due to multiple answers by respondents

¹ Product quality was an international standard for the evaluation of software quality.

² A set of attributes that bear on the effort needed to make specified modifications.

³ A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.

⁴ A set of attributes that bear on the existence of a set of functions and their specified properties.

⁵ A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

⁶ A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

4.1.2 Perception of the New Product Development Process

To grasp the employees' perception of the NPD process, a question was asked regarding the NPD process as a rule that must be followed in all phases or a guideline that can be skipped in some phases/milestones. A total of 14 respondents answered, and 5 (35%) and 9 (65%) respondents described the process as a rule and a guideline, respectively.

Five respondents who answered with a rule described that this is because it can lead to side effects and poor software quality if the development process is not followed or skipped. On the other hand, nine respondents who expressed that the development process can be skipped or needs to be tailored explained that too many administrations can slow down the software development process. Some of them suggested that it might be beneficial to merge the screening and intake phases to streamline the process. Some were touched upon that, rather than focusing on a forum or planning held by the PI core team, it might be better to delegate the whole responsibility to product owners and let them lead projects by requests coming from service owners.

There was feedback that the development process should differ depending on product type. For instance, the process can be differentiated based on whether it is a completely new product or variant product that requires minor updating from the existing product. If it is a totally new product, all developments should be thoroughly analyzed and run through the development process. However, for additional development based on existing products, some of the phases in the process can be skipped or merged to expedite the process and increase efficiency. Similar to the previous feedback, unplanned development, such as change requests from customers or additional features/functions, can be done through a light version of the development process. There was an opinion that user experience should be considered in the development process, and customers should be involved.

Similar to the survey results, it was observed through the interviews that employees had different perceptions of the development process. Some interviewees insisted that the current development process is sufficiently well structured to support new product development and that it cannot be minimized further to maintain the quality of development. On the other hand, some interviewees shared opinions that the process hinders the acceleration of development, even though they have sufficient competency, budget, and time. This is because the process makes it difficult to follow the process and keep all things that have to be prepared for the process owing to the lack of a full picture of projects. Another interviewee expressed that it might contain redundant processes because the entire process is forced to be run step-by-step, even though the project comes from the customer and requirements are already written by them.

One notable thing of the interviews was that the majority of interviewees addressed the importance of focusing on operations rather than development through the NPD process. This is because many product quality issues arise after delivering products; however,

the case company focuses on delivering products on time, which can be seen by addressing the “time” aspect without considering the importance of the “quality” aspect.

4.1.3 Estimation and Prioritization

During the interviews, some interviewees raised questions regarding the inaccurate estimation of the project in the early phase of the development process. Because inaccurate estimation negatively affects cost efficiency and overall project management, including prioritization, the need for accurate estimation in time was highlighted. When it comes to prioritization, the importance of having more transparent prioritization was addressed by the interviewees because having clear prioritization for them affects delivering products. The reasons are that how a project is initiated and prioritized is not crystal clear, and prioritization seems to be subjective rather than objective.

With respect to project management based on prioritization, some interviewees complained that there was a case in which they had to push product delivery urgently because of unexpected priority changes. This can occur when a project is prioritized. For instance, once an existing project becomes a low priority owing to appearing of another higher priority of a new project, all resources have to be reallocated to meet new prioritization. In doing so, challenges that are difficult to solve might exist within the limited resources to deliver the existing project in the arranged time. Therefore, in this case, they tend to set up an aggressive plan to meet the product delivery schedule, which can lead to failure to follow the development process and eventually low quality of the product.

One interviewee stated that since each work package within one initiative project has different properties and should be dealt with based on different priorities, prioritization could be decided at the work package level instead of the initiative project. In addition, the importance of balance within different business areas for all resources is addressed to avoid bias toward specific business areas.

4.1.4 Requirement Management

In the case company, high-level requirements are broken down into lower-level requirements in the pre-study phase, and outcomes from the requirement refinement activity are reviewed and approved by Architecture Committee (AC). Therefore, it is important to analyze requirements in a proper way and document them well in order to not only obtain approval to proceed from AC but also to provide high-quality requirements to engineers. However, regarding the requirement approval process, conflicts and arguments were identified. One interviewee who engaged in the pre-study phase expressed that AC was administrative and said:

“When you go to architecture committee before the pre-development check (PDC), that is by itself a mount Everest. You have to book a session with each of them. You have to go and present. If you are not ready, then they go come back and you have to find another slot.”

Another a similar view in above:

“Pre-study those kinds of things, I don't know exactly what we should take away, but I think it's sometimes too much administrative.”

Some of them evaluated that requirement management at the case company has been well done in a structured way; however, the requirement refinement in large-scale and complex projects could be problematic. For instance, if only one or two teams are involved in requirement refinement, the requirements can be managed by the defined process without problems. However, it is described that if ten teams need to be involved, then a problem will be created owing to complex dependencies, and it is difficult to coordinate all necessary activities across ten teams.

4.1.5 Testing

The need for early involvement of testers in the pre-study phase was addressed by the majority of interviewees. Test case design, including testing resource arrangement and environment setup, needs to be performed based on requirements, and it is usually a time-consuming task. Therefore, the later the testing team is involved in the pre-study phase, the less time will be given to them, which could eventually result in a lack of time to validate all requirements. Sometimes, the testing team was informally involved in the pre-study phase at the request of the project manager. However, in most cases, they are involved in the backlog refinement phase after the pre-study phase. Although high-level requirements are mostly written by business analysts, one interviewee stated that, from a business analyst's perspective, it is difficult to analyze and develop requirements without feedback from testers with specific technical knowledge.

“The test managers should be in involved with the team of business analysts and architect in the very early phase. And test manager has the knowledge whether this project needs to be tested such as load test or performance test etc. Then the business analyst writes requirements.”

It was addressed by testers that if the end-to-end testing perspective was not considered in the pre-study phase, then they may not manage to conduct testing owing to limited time. One interviewee stated that the requirements are not written well enough to understand from the perspective of the tester:

“It is very often a lack of end-to-end requirements. So, for the requirements of how it actually should be working, I always say they are not documented enough, indeed.”

There was some concern about capturing the test object too late:

“It would be good if we can be involved a bit in the earlier phase. I mean it does not have to be involved very much, but just a bit so that we can see and define major test objects. Because it takes time to get them several months.”

One interviewee explained that end-to-end testing is a time-consuming task, and testing for one area requires at least three days. Therefore, it is sometimes difficult to cover whole areas where testers are responsible for validating in some cases if there are either time constraints or bottlenecks. For instance, too many validation requests occur in the last iteration. However, end-to-end testing is typically required at the end of the development. Then, if critical bugs are found during testing, another iteration is required to fix the bugs, validate them again, and drag them into delaying software delivery.

There is some feedback regarding test management, some of the interviewees evaluated testing management there is room for improvement due to the fact that formal documents such as test strategy, plan, test case, acceptance criteria, and test result are not managed at the organization level. One interviewee mentioned that writing test cases was optional for the case company. Therefore, some testers write test specifications and generate test results based on proper acceptance criteria; however, others simply use the requirements to check functionality. This is because of the absence of the perception that the end-to-end test is a sort of gate activity to move forward, and it seems relevant to the absence of a well-formulated test management strategy. This could lead to poor test coverage and eventually create the potential risk of not being able to detect defects in the early stages.

“I’ve been surprised of since I started here. We have a convenient way of testing. If you see it works, then you assume it works and stamp OK in the test result without further validation.”

In addition, questions were raised regarding end-to-end testing verification due to a lack of definition between product end-to-end testing and project end-to-end testing.

“... I don't think we are gatekeepers. It's more up to project to find out. If they don't think we are needed for some reason or service teams if they think that they will manage this without end-to-end tester.”

It was difficult to secure specific internal resources affecting project management because internal resource dependency was addressed by one-third of the interviewees. It is stated that the majority of resource dependency created in the case company comes from external, for instance, business areas that cannot be controlled internally. However, there is also dependency on internal resources that can be shared by many teams, such as end-to-end testers, user experience (UX), and solution architects, and it is important to scan resource availability before planning work packages to avoid delays in their deliveries. Securing resources, such as testers, UX, and solution architects, is performed by project managers. Depending on whether the resources are in teams, project managers must scan from all the different teams to plan projects. Sometimes, securing resources is a challenge for project managers because it is necessary to contact them informally rather than in a formal setting.

4.1.6 Release

In the case company, the software as a product is released only on certain dates according to the project plan established in the initial stage of development. One interviewee claimed that the case company intends to work in the SAFe system adopting agile methodology; however, in reality, the case company is working in a hybrid methodology between waterfall and agile, since the release plan is fixed early and not aligned with PI planning.

The major feedback on software release concerns delays issues for lower-priority projects and uncertainty. It was explained by some of the interviewees that there are often cases of postponing software release to fix bugs after release for a long time because lower priority projects are always pushed back by higher priority projects. For instance, there is a task that takes approximately 5 hours to fix a bug, test, and release; however, it takes one year to release official software even though the bug is considered a critical flaw owing to internal resource prioritization.

“Those are very important things. However, when they are out of the top 20, the budget disappeared, and the priority is going to be lower and lower. So, they have to wait until the domain teams are free to do.”

A series of activities, from release planning to continuous deployment, are controlled and led by development teams; however, exceptionally, projects for truck brands are led by one dedicated person, and buses are led by another dedicated person. This is because the three business brands operate customer portals with many similarities and share some functions with minor changes. Therefore, it would be beneficial to manage them with a few dedicated persons; however, it seems necessary to closely investigate side effects that can be driven by working based on the business in the function-based organization structure.

4.1.7 Operations

During the interviews, the majority of interviewees mentioned the need to improve operations because they believed that the case company is much more focusing on development. One interviewee stated that operations phase is a main part of DevOps in SAFe; however, the importance of operations is underestimated in the case company and is not often dealt with during the PI phase (development). This can be associated with the release issues that long lead time issues for software updates mentioned in the previous Subsection.

“On the Service Level Agreements (SLAs) unfortunately, it is not that too unusual that we fail to meet those KPIs that we had agreed to and there have been too many incidents. We have taken too long time to resolve all of these.”

Suggestions to mitigate identified problems in operations were proposed in an interview, in which the case company needs to have a monitoring process to detect problems

earlier than customers, as well as to slow down the pace of the development process and spend more time on operations and maintenance of the released product to increase the quality.

4.2 Organization-related Challenges

In this chapter, the challenges regarding the organization are presented. The main findings are regarding informal terms, the project triangle, definition of quality, holistic view, resource dependency, roles and responsibilities, leadership, and knowledge management.

4.2.1 Using Informal Terminology

During the PI planning session, it was found that using the informal team's name, which is not visible in the organizational structure chart, was pervasive. It was observed again during the interviews that many interviewees often referred to the informal names of teams. Here is the list of informal team names that are often referred to by many interviewees but do not have an official organization structure at the case company: *Delivery team, DevOps team, maintenance team, service team, framework team, business area team, PI core team, squad team, one truck team, and process team.*

Undoubtedly, using the same terminology in organizations supports communication more efficiently by eliminating ambiguity and increasing clarity. Therefore, a survey question investigating employees' understanding or perceptions of informal team names within the case company was added. The survey asked respondents to choose the team the interviewees thought to belong to among the listed informal teams, and the results were analyzed based on 61 responses.

From the results of the survey, 77% of respondents were able to map themselves to specific informal teams, meanwhile, the remaining 23% answered that they did not know which informal team they belonged to. This shows that there is no distinct pattern found in terms of demographic criteria such as organizational team, job title, country, division, and so on. However, it is correlated with work experience in the case company, such as how long has been working or how much knows about the history of organizational transition.

Given the results of the analysis, a proportional relationship between work experience in the case company and understanding of informal teams was identified. In other words, the longer the employee worked in the case company, the better their understanding of informal teams. For instance, 67% of respondents who had been working for less than three years answered they knew about informal teams; on the other hand, it increased to 100% for respondents who had over ten years of work experience at the case company (see Table 4). Given this result, it is reasonable to presume that employees who have joined or have less work experience at the case company may face challenges with senior employees, who often refer to informal teams.

Table 4. The knowledge of informal team names based on working experience

Attribute		0-3 year	3-5	5-10	Over 10	Total
Number of Respondent (Percentage)	Know	14 (67%)	13 (68%)	14 (93%)	6 (100%)	47 (77%)
	Do not know	7 (33%)	6 (32%)	1 (7%)	0 (0%)	14 (23%)
	Total	21 (34%)	19 (31%)	15 (25%)	6 (10%)	61 (100%)

There was other feedback regarding the use of informal terminology, particularly regarding testing. It is described that confusion and inefficiency can be increased owing to different definitions of the end-to-end test on different teams, which can negatively affect the quality of software. This is because the end-to-end test is defined by different perceptions (product vs. project) of the test objects, depending on the teams. In addition, the difference between verification and end-to-end testing is not clearly defined, which can cause confusion for employees.

4.2.2 Holistic View

It was a pervasive opinion that the majority of interviewees mentioned the difficulty of having a holistic view, which can be divided into two categories: a lack of end-user perspective and a lack of competency.

Lacking End-user Feedback

The importance of receiving feedback from end-users who actually use products was touched upon by some interviewees. Due to the long communication chain from the case company, business brands, and dealers to end users, there is a difficulty and potential risk that end users cannot obtain high-quality services what actually they expect.

Two approaches to collect feedback and capture end-users' expectations were suggested by some of the interviewees. One approach is to collect feedback from dealers who most closely communicate with end-user. Another is to conduct user research and interviews with end-users such as drivers and fleet managers to obtain different perspectives that the case company is missing now. User feedback helps function developers obtain a holistic view of the product they are developing. This is because user feedback can be used to improve software and incorporate it into the next release. Through constructive interaction with end-users, developers can expand their view of the product from a functional level to a holistic level.

“If the developers really understand what they have to develop, then they will do the right things and the quality will be better. but if they do not have the full picture of it, then they cannot...”

Lacking Competency

In conjunction with end-user feedback, some interviewees raised questions regarding employees' competency regarding development teams and product end-to-end testers. It was explained that development teams do not have sufficient competency to understand the total solutions, which are fairly complex and large environments. Another aspect of competency was touched upon regarding the deviation in maturity levels between the development teams. It was explained that since the maturity of each development team differs depending on how high the proportion of newcomers is on the team, it affects the team's ability to deal with the information for total solutions.

Additionally, one of the interviewees raised questions regarding the competency of the product's end-to-end testers. To validate the product at the vehicle level, the importance of having a holistic view was highlighted because of the complex dependencies across multiple functional domains. However, knowledge and experience of product end-to-end testers are limited within the product domains, rather than having an entire picture of projects. Therefore, if the scope of the development product is beyond the scope of the product's end-to-end tester, the limited knowledge of the end-to-end tester might affect product testing and verification.

4.2.3 Roles and Responsibilities

Various challenges regarding roles and responsibilities (R&R) were observed at the case company during the research study. In particular, role conflict and ambiguity at the project and product levels were analyzed as critical. In addition, a lack of ownership and misalignment between job roles and teams were also highlighted.

Role conflict is often observed between project and product end-to-end testing managers. In the case company, project end-to-end testing is aimed at performing cross-functional tests on real vehicles, and project test managers belong to verification and validation teams. Meanwhile, product end-to-end testers perform specific functions in both soft vehicles (virtual environment) and real vehicles, and product test managers belong to each function development team. According to one interviewee, different teams in different organizations often talk about different things regarding end-to-end testing.

it is stated that many testing activities overlap, creating confusion and conflict when end-to-end testing is needed because the test scope of end-to-end testing is not defined clearly between the project- and product-level testing. This conflict can result in decreasing work efficiency as well as quality because there are some possible risks that validation is done redundantly by two teams or that validation is missed due to the unclarity of which team is responsible. Additionally, a similar conflict was found between the UX team and UX developer in the functional teams.

“We have similar test cases where it gets confused for a lot of people regarding the reason why are teams doing same testing. But their intention is more on the project

level of how it must be delivered to the vehicle along with the delivery of the services.”

There are also opinions that role conflict can be developed into another challenge described in the previous "holistic view" and "resource dependency" Subsections. It is addressed that if there are too many overlapped works, it can be even more difficult for project managers to secure all necessary resources in the time, since unclear boundaries between them confuse project managers about whom to contact. There was a voice about the information asymmetry that hindered having the holistic view of the projects.

“There is a lot of collusion with the verification and validation team sometimes issues are reported directly to the verification and validation team, and we (functional testers) don't get information. There is a lot of confusion between teams, so that's challenges we are facing now.”

Role ambiguity arises when employees have insufficient or inconsistent information to perform their jobs adequately. From this perspective, some interviewees mentioned that role ambiguity exists between project managers and product owners (or product managers) and that there might be role conflict due to duplicated tasks between them.

“...It can be a competition on occasion neither party knows like it's not so much a competition as a lack of knowledge about each other's roles.”

Some interviewees raised questions regarding the lack of ownership and leadership. For instance, there are a number of processes within the case company; however, most of them have no assigned process owner. Since no one manages the process at the top level, it leads to the discretion of the individual or team when an issue or improvement is needed immediately.

There may not be an urgent problem in the short term because of a lack of owners. However, if this is prolonged and managed autonomously, the culture of working through the processes will become unstandardized. Despite the importance of potential issues, most employees may not be aware of this problem. This was confirmed through the interviews.

“No one cared about ownership of other tracks in the processes.”

Leadership issues have been raised, in addition to a lack of ownership. In particular, the situation at the time of the introduction of SAFe work was explained. Initially, there were roles centrally managing SAFe governance at the case company; however, the roles had gone, and it affected the way each department started to be different before maturing and establishing the culture as a new way of working. Therefore, there is a mix of old and SAFe workers. Although this is an important matter that needs to be

managed at the company level, it may eventually lead to a misalignment with the company's strategy because the person in charge is not clearly defined and managed.

Misalignments between the job roles and teams were also observed. One interviewee stated that the current job is inconsistent with the team to which the interviewee currently belongs. Therefore, finding where to belong in order to align between the job role and possible team is still unclear for the interviewee where to move, even though they have more than a year of meetings with HR and the line manager. In addition, it explained that finding the team where the interviewee was to move entirely depended on the interviewee's decision rather than being guided by the HR or the line manager.

“I have not decided exactly yet where to move. As I said, I didn't belong to any team so...It's not very well structured actually...”

Another interviewee stated that the current job is performed in the later phase of the development process; however, the current team that belongs is a team performed during the early phase of the process. Therefore, this raised the question of why the interviewee belonged to the current team. Based on these two examples, it seems necessary to clearly define how an organization is structured and deployed. In addition, it seems necessary to consider that the right person can be used in the right place to create synergy at the company level.

4.3 Knowledge Management-related Challenges

Most interviewees agreed on the difficulty of finding the right information and contact points at the case company. Interviewees who worked for at least three years said that they could find the correct people by putting in some effort for a while using their own networks. Even though the interviewees could find the right information and contact points, all interviewees agreed that it would be difficult to find the correct information for newcomers who have fewer networks.

“It could be hard to find things, and you do not always know if the information is accurate or if it is actually old. So, I think that is a problem.”

Furthermore, a lack of technological environment drags hampering knowledge sharing at the case company was observed. In the U.S. team under DI30000, the difficulty of accessing internal systems that employees' access in Sweden turns out to be due to limited VPNs. Therefore, they are reliant on the Swedish team to have the same knowledge and information. In addition, limited access to knowledge sharing makes them work in different ways. This may not be the main reason for adopting another development process and way of working. However, they could not share knowledge, so employees in the U.S. team created their own documents and evolved into their own different ways of working, such as working with startups on a partnership or crowdfunding basis to explore the market.

Knowledge management might not directly affect working in the case company in the short term. However, if employees have incorrect information and believe information is true until no one corrects it or has less opportunity to share knowledge and information in a formal way, it can hinder the alignment of strategy at the case company in the long term.

4.4 Evaluation of SAFe Maturity

To evaluate the maturity level of SAFe at the case company, a self-completion questionnaire with 60 questions for different practices was sent out, and answers received from five respondents were analyzed using the SAFe MM. As a result of the analysis, it turned out that there is a tendency that as the level increases, the number of practices evaluated as "not achieved" and "partially achieved" increased, while the number of practices evaluated as "fully achieved" and "largely achieved" decreased (see Table 5).

In summary, four practices—coding standards, collaborative teams, knowledge sharing, and self-organized teams—were assessed as mature. Regarding knowledge management, a deviation between the SAFe MM results and the empirical findings based on the interviews was observed. Since it was addressed by many interviewees that knowledge sharing at the case company can be difficult for newcomers who have less experience and networking, the result of knowledge sharing is ranked as fully achieved. By taking into account the results from the interviews and the SAFe MM, it can be presumed that the respondents of SAFe MM might have had long working experiences at the case company.

However, there are more immature practices classified as "not achieved" or "partially achieved" than mature ones. Based on the similarity and property of practices, it can be judged that practices that need improvement are areas related to customer involvement, risk management, planning, estimation, measuring, tracking, release, and testing. Therefore, the SAFe MM results correspond with the empirical findings, as similar challenges were identified in the interviews.

Attribute	Level 1	Level 2	Level 3	Level 4	Level 5
Fully Achieved (FA)	<ul style="list-style-type: none"> - Coding standards - Collaborative teams - Knowledge sharing 	-	<ul style="list-style-type: none"> - Self-organized teams 	-	-
Largely Achieved (LA)	<ul style="list-style-type: none"> - Acceptance testing - Empowered and motivated teams - Task volunteering 	<ul style="list-style-type: none"> - Automated testing - Evolutionary requirements - No big design up front - Product backlog - Requirements Discovery - Software configuration management - The define/build/test team - Tracking iteration progress 	<ul style="list-style-type: none"> - Continuous integration - DevOps - Unit tests 	<ul style="list-style-type: none"> - Intentional architecture - Managing highly distributed teams 	<ul style="list-style-type: none"> - Continuous SAFe capability improvement
Partially Achieved (PA)	<ul style="list-style-type: none"> - Collaborative planning - Reflect and tune process - User stories 	<ul style="list-style-type: none"> - Agile estimating and velocity - Continuous delivery - Customer contract reflective of evolutionary development - Release planning - Smaller, more frequent releases - Two-level planning and tracking 	<ul style="list-style-type: none"> - Agile Release Train - Continuous improvement (Refactoring) - Frequent face to face communication - Impact on customers and operations - Mastering the iteration - Plan features not tasks - PSI/Release - Regular reflection and adaptation - Roadmap - Vision, features 	<ul style="list-style-type: none"> - Adaptive planning - Continuous customer satisfaction feedback - Customer contact revolve around commitment of collaboration - Customer immediately accessible - Measuring business performance - Smaller & frequent releases (every 4-6 sprints) 	<ul style="list-style-type: none"> - Agile project estimation - Changing the organizations - Concurrent testing - Low process ceremony - Test driven development
Not Achieved (NA)	<ul style="list-style-type: none"> - Customer commitment to work with development team 	-	<ul style="list-style-type: none"> - Risk driven iterations 	<ul style="list-style-type: none"> - Client driven iterations 	<ul style="list-style-type: none"> - Face to face interaction between develops and users - Ideal agile physical setup

Table 5. The result of evaluating SAFe Maturity at the case company

5. Benchmarking

This chapter presents the results of case studies from the two companies based on interviews. The purpose of benchmarking is to determine how other companies manage and operate on the challenges faced by the case company.

5.1 Company X

Company X is a global IT company with over 25,000 employees. It develops and operates cloud computing, hosting, logistics, and digital transformation services with different types of products such as IaaS, PaaS, and SaaS, and supports Company X's group companies. Company X is driven by total quality management with enhanced security. The company possesses multiple international standards certificates and achieves CMMI Level 4, which is consistent with the quantitatively managed stage.

Due to its large scale of organization, the company is divided into several divisions: front-end and back-end. Additionally, some specific solutions, such as logistics, are set as one business division. There are several front-end divisions, and each division supports different groups of companies. In the case of back-end division, it operates not only infrastructure but also provides cloud services with multiple business area services. Company X adopts an ambidexterity structure, and the HR, finance, and quality departments exist under the CEO and each division. The product planning team plays a significant role in managing products. However, because of the quality-driven culture, the quality team under the CEO has ownership of the NPD process and its governance, and quality teams from each division guide and support the development of new products in their divisions.

Company X's NPD process consisted of five phases: product planning, product plan/analysis, development sprint (#1~N-1), final development sprint (#N), and launch/operations. In the product planning phase, cooperative product planning, development, and UX teams set a product concept. The product requirement document (PRD) is reviewed by SME functions (UX, architect, performance, security, open-source, legal/patent teams), quality, and development teams by having a meeting to set a holistic support plan. During the development phase, frequent support and guidance are provided by SME, as well as a testing team. In the final development sprint phase, the code is frozen, and the final testing is conducted by the SME and the testing team. Once the testing results are released, the product launch is approved.

According to the interview, the product requirements are led by the product planning team, and their role is to define acceptance criteria. Reusability is also checked in this phase if there are any cases of reusability from previous projects when PRD is written. A writing test case is responsible for the development team and managed as a mandatory activity. Based on the test case, product verification was led by verification teams, such as SMEs and testing teams, based on the guidelines and standards that they explained to development teams. However, the entire product development process is

led and facilitated by a quality team. Company X also adopts a large-scale agile way of working when developing new products; however, it was not clear which type of large-scale they work on. When it comes to the NPD process, the company's NPD process is inflexible. However, depending on the scope of product development, such as totally new products or extra development based on existing features, milestones, such as types they have to be verified by SMEs, can vary. Since product quality is significantly relevant to testing, it was addressed that verification should not be flexible and that verification is actually quite strictly proceeding. Furthermore, it was explained that verification should have proceeded by a third party to verify the product objectively.

When incidents or issues occur after delivering products, they are evaluated by the quality team based on each division that the incidents or issues occur from in the testing cases or outside of the testing cases. In addition, events or incidents were managed. In particular, when incidents occur and do not fulfill SLAs, the division will get a penalty. Operations are managed by conducting an internal audit by performing quality groups to increase the performance of their operations by adopting CMMI with plan/do/check/act (PDCA) on a yearly basis. To increase the collection of customer feedback, the customer satisfaction index (CSI) was calculated on a yearly basis. However, the target of CSI is its business brands, rather than its end-user.

5.2 Company Y

Company Y is a global automotive company with approximately 100,000 employees working worldwide. Company Y adopts SAFe some years ago, and the NPD process, culture, and organization have been changed to support time to market. Company Y had composed of divisions based on solutions that were responsible for quality, cost, and performance aspects. After reshaping organizations to fit SAFe way of working, they reunited some of the divisions and created new divisions based on hardware, software, propulsion, release, and innovation divisions. Each division consisted of multiple agile release trains (ARTs) where approximately 80 to 120 employees work in. ARTs are virtual organizations and self-organized structures that are responsible for developing, delivering, and operating solutions/services in the agile way.

Company Y operates different NPD processes based on three project classifications. The first project classification is for the first production car running on a new platform, which takes approximately four and five years to develop and launch in the market. For this project classification, all phases in the NPD process shall be run through. The second project classification is for the variant cars running on the existing platform, which take approximately four years to launch in the market. Compared to the first classification, the same phases exist in the development process but are streamlined to expedite the process. Therefore, some milestones in each phase are skipped. And the last project classification is for the new release of the existing vehicle, which takes about over 18 weeks. The development process for this project classification is

significantly minimized in comparison to the other two classifications to meet market and customer demands in a timely manner.

According to the interviewee, role ambiguity and conflict could have been mitigated since roles and responsibilities in SAFe teams are defined clearly at the company level when they introduce the agile framework, and there had been efforts to spread information within organizations. For instance, there are no known issues due to role ambiguity and conflict between the project managers and the product owners at Company Y. Nevertheless, they figured out some of the mandatory roles such as scrum masters in SAFe teams are vague, and they said a discussion is ongoing if the scrum master role is needed or not. Additionally, it was described that complexity and inefficiency can be created because there are three leaders which are the product manager, product owner, and scrum master in each SAFe team.

To increase product quality and meet customer needs especially focusing on end-user's feedback, they are securing information internally as well as externally. User representatives who have had long experience and deep knowledge in vehicle testing collect complaints from users in various ways such as diagnostic info collected at workshops, customer surveys and so on. Additionally, they also collect and purchase field issues from specific companies contracting to provide services. All collected issues and feedback are registered into their internal system for traceability. After acquiring feedback, Company Y analyzes the data and finds gaps with relevant function teams. The found information and solution will be adopted in the existing project through the simplest development process and used for planning new products if possible.

Regarding securing cost optimization and resource efficiency, Company Y is reformulating the way of working, NPD process, and organization. They have decided to remove SAFe as the central way of working and remove a connection to organizational structure; however, stay with the good part of agile were efficient. They are trying to simplify the framework by removing mandatory roles and ceremonies that can be regarded as inefficiency. Additionally, they will focus on more delivery rather than process. For instance, PI planning had been simplified from three days to one day of activity because they want to focus more on efficient and dependency synchronization to promote collaboration.

6. Discussion

In this chapter, the challenges identified in the empirical findings are analyzed and discussed in relation to theoretical frameworks and benchmarking.

6.1 Inflexible Process

The findings generally indicate that there are challenges related to process, organization, and knowledge management, and these challenges are directly related to “*time*” and “*quality*”. The meaning of time in developing products is the delivery of products at a planned time. On the other hand, the meaning of quality indicates that before and after delivering products are based on employees’ perspectives, as they define quality as *maintainability, reliability, functionality, usability, efficiency of products, and customers’ success*.

Because time and quality are chosen as the most important factors when developing products for both employees and customers, there is a tendency for the current NPD process to support both time and quality factors, even though they should be compatible. The NPD process is regarded as a “rule” that must follow the process because it affects the poor quality and code of products in the end. However, it is also considered as a “guideline” that can be skipped or tailored based on preference because the process hampers the expedited delivery time of products.

Cooper (2022) states that NPD processes must provide flexibility because all types of projects differ, and this argument is consistent with one of the benchmarking companies that has already developed its products through three different types of processes based on project classification. However, all types of projects in the case company have been developing through a single NPD process, even though there are two classified projects. Given this situation, the reason why the NPD process is regarded as hampering product development by some employees can be presumed to be correlated with the time aspect, since all projects are developed through one NPD process.

Pedreira et al. (2007) highlighted that providing high-quality products to customers in a timely manner can be achieved by tailoring the NPD process. Therefore, to increase development productivity by fitting the process to projects as well as employees who are considered the main users of the process, developing different processes for different environments can be considered. In addition to developing all projects using one NPD process, it is identified that one of the milestones can hinder the agile way of working and slows down software development. Some of them overall evaluated that the requirement management process at the case company has been well built in a structured way and has functioned well; however, they also agreed that sometimes it is administrative and bureaucratic. Therefore, to meet employees’ satisfaction, the NPD process needs to be tailored to streamline the unnecessary administrative parts, taking the project’s properties into consideration.

To increase the efficiency and productivity of software development, it is important to eliminate wasteful administrative management in the development process. However, quality is not a compromise. This means that the quality of the product should not be degraded as a result of process tailoring but rather improved in a way that increases. However, the definition of quality can be argued at the case company, since the meaning of quality is not clearly defined at the organizational level. As mentioned above, employees define the meaning of quality. Over 80% of employees defined that maintainability, reliability, functionality, and usability. Given the situation at the case company, the theoretical framework, and benchmarking, how to improve these meanings of quality as well as being compatible with time and quality factors should be considered when tailoring the NPD process.

6.2 Informal Resource Allocation

The majority of interviewees were aware of the importance of managing and breaking requirements in the pre-study phase by collaborating with relevant stakeholders. However, it turned out that it is difficult and insufficient to break down requirements by only referring to architect and security perspectives that there is a likelihood of missing *time* and *quality* aspects when developing products. This is because of missing testing and UX perspectives that are significantly related to later phases of the NPD process when managing projects owing to internal knowledge, processes, and resource dependencies.

It is difficult for business analysts to write the required documents without testing knowledge. From the testers' perspective, the need for early involvement in the process was addressed because the late defining of the test object affects delaying the project to prepare the testing environment, which is related to the time aspect. In addition, requirement documents are insufficient for testers because end-to-end testing perspectives are missing.

Cooper and Sommer (2020) address the importance of upfront homework when developing new products to gain the greatest profitability, since most of the cost and time is spent in the middle of projects. Similar to up-front homework, Baumgartner et al. (2021) argued the importance of early involvement in testing because it provides benefits such as cost, time, and code quality. Given these statements, it is understandable why the need to involve testers when breaking down and writing requirement documents in the early phase of the NPD process affects time and quality aspects.

Unlike architecture and security, which are formally set by top management, project managers contact testing and UX teams informally way in order to secure relevant resources. However, the difficulty of securing these resources, as it is necessary to scan from all the different teams, should be conducted by project managers. Berntzen et al. (2021) argued the importance of boundary-spanning roles that can act as a bridge across

teams within three different dependencies. One of the benchmarking companies manages a formal joint discussion session led by a quality team, which can be seen as a boundary-spanning role. However, this indicates a gap between the finding, benchmarking, and theory since project managers act as a boundary spanner at the case company not only spending extra time but also determining without having official decision-making from relevant stakeholders that might affect product quality in the end.

6.3 Lacking Testing Governance

Several researchers have focused on developing and managing test specifications in a structured manner. The importance of having good-quality test specifications was highlighted by Dikert et al. (2016), as its quality will have a positive correlation with the quality of products. When managing well-written test specifications, many resources can be saved by reusing test cases (Barraood et al., 2022), and it is possible to detect all defects in advance and secure good quality by expanding the test coverage (Baumgartner et al., 2021).

However, the findings indicate that testing governance is insufficient at the case company because the testing strategy, standards, planning, and writing of test cases are managed in an ad-hoc way rather than providing clear structures, even though testing is directly related to product quality, as addressed by Dikert et al. (2016). The need to focus on operations rather than development by employees might be relevant with a lack of testing governance, since there are frequent incidents, a long lead time to resolve them, and the inability to fulfill SLAs after delivering products. It can also be presumed that the culture at the case company should focus on product delivery, which can be seen as a time aspect underestimating the importance of testing.

In addition, the findings between the interviews and SAFe MM regarding testing should be discussed as to why the findings indicate some contradictions. As mentioned above, there is a tendency that testing governance in the case of a company is not well structured based on the interviews. However, the SAFe MM results indicate that acceptance, automation, and unit testing are largely achieved, which means that errors/issues/incidents after delivering products should not occur frequently because these tests support the detection of quality issues before launching products. Therefore, it also should be discussed the degree of the three types of testing is enough defined and covered testing coverages.

6.4 Lacking Roles and Responsibilities

Role ambiguity and conflict are examples of decentralization. Within a team that defines specific roles, the roles are performed well; however, cross-functional work becomes an issue because similar roles can exist in other teams, and it becomes challenging to collaborate with them. In the case company, it is indicated that project- and product-level teams become challenging. Additionally, it is mentioned that informal job titles and their responsibilities can result in decreasing collaboration and

productivity between teams as well as creating many redundant tasks and confusion about whom to contact. This challenge can be regarded as a drawback of a decentralized way of working, caused by the fact that job titles or responsibilities can be freely defined inside the team because of autonomy derived from decentralization.

Tolbert and Hall (2019) highlight that individual or group conflicts tend to expand into organizational conflicts. Job and organizational formalization can be a way to mitigate conflicts. In addition, defining and codifying roles based on job observation is essential to recognizing how many different types of roles perform their work properly. In doing so, consistent decision-making can increase leadership and prevent authorization erosion within boundaries (Hempel et al., 2012). Furthermore, the right people can be hired in the right places, which can affect the total labor costs at an organizational level.

6.5 Lacking Knowledge Management

It has been found that using informal terminologies is pervasive and can confuse various levels of organizational structures, as individuals may have different understandings and definitions of the same terminologies (Nyfjord et al., 2014). In addition, finding correct information and people generally relies on network-based rather than codified documents, which can lead to inefficient and ineffective communication (McMillan & Ihrig, 2015). Using informal terminology and network-based knowledge management might be sufficient for employees who are familiar with the company's culture and way of working. However, it can be difficult to adapt the method of sharing knowledge for newcomers who have fewer networks and are unfamiliar with the culture and the way of working.

There is a tendency to acquire new information and norms and clarify roles and responsibilities when newcomers join a new organization by adjusting their behavior to fit the organization (Judeh, 2011). Given this argument, the current knowledge management in the case company may yield confusion and hamper newcomers' adaptation, which can be seen as unsuccessful organizational socialization (Judeh, 2011). In addition, it affects deviated team maturity in the case company because there is a tendency to show a low maturity level if there are many newcomers in specific teams.

Another factor hindering knowledge sharing in the case company is the lack of a technical environment to access knowledge repositories such as Confluence and Jira. Limited knowledge assessment leads employees to rely on network-based methods and promotes a different way of working and processing. It is yet to be determined whether the use of informal terms, ways of working, and processes to promote decentralization or decentralization facilitate the use of different terms. However, it is clear that there is a correlation between them.

One of the benchmarking companies also utilized SharePoint, Confluence, and Jira as their main tools. On the other hand, official information and knowledge are managed through SharePoint to manage updated knowledge, which can be accessed by every employee. For the case company, it can be a discussion topic on how to provide the same opportunity to access official information and knowledge, such as the process and way of working, by providing a site map to see a holistic view of knowledge.

6.6 Unbalance between Development and Operations

This finding indicates that inaccurate estimation leads to non-transparent prioritization and unbalanced resource allocation. In addition, the results of SAFe MM related to estimation, prioritization, planning, and measurement also similar indications that are relevant with the empirical findings based on interviews. The thirteen practices achieved as partially.

- Collaborative planning (Lv1, PA)
- Release planning (Lv2, PA)
- Agile estimating and velocity (Lv2, PA)
- Smaller, more frequent releases (Lv2, PA)
- Two-level planning and tracking (Lv2, PA)
- Roadmap (Lv3, PA)
- PSI/Release (Lv3, PA)
- Agile Release Train (Lv3, PA)
- Plan features not tasks (Lv3, PA)
- Adaptive planning (Lv4, PA)
- Measuring business performance (Lv4, PA)
- Smaller & frequent release (every 4-6 sprints) (Lv4, PA)
- Agile project estimation (Lv5, PA)

For instance, the prioritization of product development still resolves issues and incidents in the operations phase because internal teams tend to focus on development within limited resources rather than operations. Therefore, inaccurate estimation affects the deviation of balancing between development and operations and leads to customer dissatisfaction due to the long lead time to resolve issues and poor quality.

According to Sreenivasan and Kothandaraman (2019), CMMI can ensure the predictability of operations and support accurate estimation by attaining higher CMMI levels with organizational standards. This theory is in line with one of the benchmarking companies that attained CMMI Level 4, which is quantitatively managed, and it turned out that the company can manage at a micro level of estimation. By referring to the findings, the theory and the benchmarking, adopting, and driving CMMI by top management in order to have accurate estimation needs to be discussed, which will affect the majority of prioritization, measurement, planning, and imbalance between development and operations.

6.7 Lacking Customer Involvement

A tendency to lack customer involvement owing to a long chain of communication from the case company to end users was observed, despite the importance of increasing customer collaboration when developing new products. Furthermore, the similarity was identified from the result of evaluating SAFe MM that most of the practices are relevant to customer aspects rank in a low achievement from level 1 to level 5. This opinion was supported by the result of conducting SAFe MM; the six customer-related practices were evaluated as having lower achievement:

- Customer commitment to work with development team (Lv1, NA)
- Customer contract reflective of evolutionary development (Lv2, PA)
- Client driven iterations (Lv4, NA)
- Customer immediately accessible (Lv4, PA)
- Customer contact revolve around commitment of collaboration (Lv4, PA)
- Face to face interaction between develops and users (Lv5, NA)

Lack of customer involvement makes development teams focus only on some functional parts instead of having a holistic view of the technology that customers need for developers. Cooper and Sommer (2020) argued that misunderstanding customer needs, targeting the wrong market, or incorrect pricing causes firms to fail, and their argument makes it understandable why customer involvement in the case company is important. One of the benchmarking companies also collected end-user feedback by contracting with a company. As mentioned in Subsection 4.2.2, user research can be discussed as a possible innovative way to obtain customer opinions, especially in collecting end-user feedback. By adopting “*ambidexterity*”, user research from the bottom level can be exploited by top management rather than by contracting other firms.

7. Recommendation

In this chapter, recommendations for the case company are presented based on the analysis of the discussion and benchmarking chapters. The recommendations are based on aiming to decrease TTM and increase product quality, since *time* and *quality* are perceived as the most important factors when developing a new product at the case company.

1. Flexible Process
 - 1.1. Adopting a flexible process based on project classification.
 - 1.2. Taking into consideration merging screening and intake.
 - 1.3. Building the quality aspect defined by employees into the NPD process.
2. Formal Joint Requirement Discussion
 - 2.1. Setting formal joint requirement discussion involving relevant functions.
3. Customer Involvement
 - 3.1. Increasing customer involvement.
 - 3.2. Taking user research into account exploiting by top management.
 - 3.3. Taking conducting CSI into consideration on a yearly basis.
4. Ambidexterity (Trade-off)
 - 4.1. Enhancing testing strategy and standards.
 - 4.2. Formalizing jobs by observing and codifying.
 - 4.3. Setting E2E testing boundaries between project and product testers.
 - 4.4. Decreasing administrative AC and providing in a supportive and practical way.
 - 4.5. Balancing between development and operations to improve product quality.
 - 4.6. Taking into consideration adopting CMMI to improve planning, estimation, measurement, and operations.
5. Knowledge Management
 - 5.1. Defining and using formal terms to avoid confusion.
 - 5.2. Codifying and updating knowledge for newcomers' socialization.
 - 5.3. Setting the same technical environment to have the same opportunity.

8. Conclusion

In this chapter, the conclusions of the research study and the answers to the research questions are summarized. The limitations of this research are then discussed. Finally, recommendations for future research are provided.

8.1 Answers to the Research Questions

The importance of NPD processes and organizations in meeting customer needs and TTM is presented in Chapter 1. Similarly, the need to investigate and evaluate the current situation in these three areas has been addressed in relation to the fact that the case company is improving continuously to meet customer needs and TTMs. Therefore, the purpose of this study was to examine the current NPD process, way of working, and organization, identify the challenges facing the case company, and analyze them from an academic perspective. Two research questions were created based on the analysis, and this chapter summarizes the answers to these questions.

RQ1. What are the challenges faced when developing products through the NPD process, way of working and organization?

The purpose of research question 1 was to investigate the current challenges in the case company developing through the NPD process and how organizations affect and are affected by each other. To achieve this goal, the identified challenges related to processes and organizations are presented in Chapter 4.

The lack of flexibility in NPD processes for process-related issues has confirmed the need for tailoring processes based on project classification. In addition, challenges have been identified for administrative AC management, dependency, informal resource allocation, inaccurate estimation, non-transparent prioritization, lack of test governance, inefficient release stages, and lack of operational phase focus. In terms of organization-related challenges, role conflict and ambiguity, lack of ownership, knowledge management, pervasive use of informal terminology culture, lack of a holistic perspective due to missing end-user feedback, and lack of competence have been identified.

While one of the main aspects of this study was to evaluate the NPD process, the need to focus on operations rather than development was addressed. This is because the case company pursues a development-oriented environment, and it makes teams focus less on operations. Therefore, an imbalance between development and operations leads to frequent errors and incidents after delivering products that can be recognized as product quality to customers. In addition, owing to a lack of trade-off between development and operations, it becomes pervasive, taking a long lead time to resolve defects and unfulfilling SLAs.

RQ2. What are recommendations in order to overcome the challenges based on theoretical framework?

The purpose of Study Question 2 is to provide recommendations based on a gap analysis between the theoretical framework and empirical results. Research questions are addressed in detail in Chapter 6, and improvements based on these are covered in Chapter 7. To secure a holistic view of the challenges, the challenges derived from were analyzed. The results show that the root causes of these challenges are inflexible processes, decentralization, lack of customer engagement, and lack of knowledge management. Recommendations are divided into five categories: flexible process adoption, formal joint requirements discussion setup, increased customer engagement, improved centralization, and improved knowledge management.

First, it is recommended that a flexible NPD process be adopted based on project classification. To do this, merging the initial stages of screening and intake and balancing development and operations is recommended. Second, relevant critical functions such as E2E testing, UX, and so on, are incorporated into AC for supporting a formal joint requirements discussion. This has not been discussed together through informal cooperation when defining requirements. In doing so, it supports the reflection of perspectives. It also recommends decreasing the formalization of AC by identifying unnecessary activities. Third, it increases customer engagement to gain a technical perspective and development team capabilities. It is recommended that end-user perspectives be constantly obtained through user research by obtaining feedback through user research. Fourth, most of the company's challenges come from decentralization, which can be seen as autonomy. Therefore, it is recommended that centralization is increased. This can be done by formulating organizations and individuals, setting standards and codifying, and providing boundaries to prevent abuse of authority outside the team. This does not mean that each team loses its autonomy, and that the team can still have the opportunity to explore innovation and make decisions within boundaries. In addition, it is recommended that CMMI be considered because of the lack of focus on operations. Adopting CMMI can improve the transparency and accuracy of release, estimation, and prioritization. Finally, knowledge management can be enhanced by defining and using formal terms to avoid confusion among teams. Organizing and updating knowledge for the socialization of newcomers is also recommended to prevent team deviations. In doing so, it provides the same opportunities to access knowledge by establishing the same technology environment.

8.2 Limitation of the Research

Three limitations were identified when conducting the study. First, when it comes to evaluating SAFe Maturity in the case company, the evaluation result is valid because the results show that there is a tendency to correspond with the challenges identified through interviews. However, owing to a few responses to the self-completion questionnaire, it is difficult to address whether the result can represent all teams possessing the same level. Second, because of the late identification of the two different

end-to-end tester roles, which are the project level and product level, some parts that are relevant to testing were analyzed, not taking into consideration the difference between them. Third, evenly distributed perspectives were missing during the interviews, particularly for employees participating in the later stages of the NPD process. This is because the participation rate of the interviewees in the final phase of the process was low. Finally, due to the root causes identified after completing the analysis of empirical findings, literature reviews related to the root causes could not be sufficiently covered.

8.3 Recommendation of Future Research

The recommendations of future research are folded into three subjects: tailoring CMMI to fit the case company, developing CSI, and evaluating organizational structure and span of control.

When it comes to tailoring CMMI to fit the case company, SAFe MM is still valid for evaluating the maturity level of SAFe at the case company because the evaluation result shows a tendency to correspond with the challenges found through the interviews. However, since CMMI v2.0 launched in 2018 and it entails an agile way of working compared to CMMI v1.3, tailoring CMMI v2.0 can provide an in-depth evaluation model rather than SAFe MM, which is used to measure in this study. This is because CMMI was developed by a prominent institute and is updated continuously. Furthermore, it provides domains, capabilities, and PAs that can be much more detailed than SAFe MM. Therefore, it is recommended to tailor CMMI v2.0 and evaluate the maturity level on a routine basis.

Developing CSI from both business brands and end-users' perspectives are recommended because the majority of interviewees touched upon the need for feedback from customers, especially from end-users. In doing so, what must be improved at the case company can be identified based on an analysis of the result. Furthermore, formalized regular evaluation CSI can be a valuable input for the case company to improve not only the NPD process but also process governance and organization.

With respect to evaluating the organizational structure and span of control at the case company, it is recommended to perform job observation and job codification. This is because the current organizational structure in the case study is a cross-functional organization that weighs decentralization. However, owing to decentralized organization, there might be a likelihood of job conflicts across teams, similar to the job conflict of the E2E testers. In addition, the number of employees varies depending on the team. Organizational cost and communication are related to the span of control; a suitable organizational structure and span of control are necessary for future studies.

References

- Akbar, R. (2019). Tailoring agile-based software development processes. *IEEE Access*, 7, 139852-139869.
- Barraood, S. O., Mohd, H., & Baharom, F. (2022). An initial investigation of the effect of quality factors on Agile test case quality through experts' review. *Cogent Engineering*, 9(1), 2082121.
- Baumgartner, M., Klonk, M., Pichler, H., Seidl, R., & Tanczos, S. (2021). *Agile Testing*. Springer International Publishing.
- Bell, E., Bryman, A., & Harley, B. (2019). *Business Research Methods*. Oxford: Oxford University Press.
- Berntzen, M., Stray, V., & Moe, N. B. (2021, June). Coordination strategies: managing inter-team coordination challenges in large-scale agile. In *Agile Processes in Software Engineering and Extreme Programming: 22nd International Conference on Agile Software Development, XP 2021, Virtual Event, June 14–18, 2021, Proceedings* (pp. 140-156). Cham: Springer International Publishing.
- CMMI Product Team. (2010). *CMMI for Development, Version 1.3* (CMU/SEI-2010-TR-033). Retrieved June 19, 2023, from the Software Engineering Institute, Carnegie Mellon University website:
<http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661>
- Cooper, R. G. (2016). Agile–Stage-Gate Hybrids: The Next Stage for Product Development Blending Agile and Stage-Gate methods can provide flexibility, speed, and improved communication in new-product development. *Research-Technology Management*, 59(1), 21-29.
- Cooper, R. G. (2019). The drivers of success in new-product development. *Industrial Marketing Management*, 76, 36-47.
- Cooper, R. G. (2022). The 5th Generation Stage-Gate Idea-to-Launch Process. *IEEE Engineering Management Review*.
- Cooper, R. G., & Sommer, A. F. (2020). New-product portfolio management with agile: challenges and solutions for manufacturers using agile development methods. *Research-Technology Management*, 63(1), 29-38.
- Daft, R. L. (2015). *Organization theory and design*. Cengage learning.

De Meyer, A., Loch, C. H., & Pich, M. T. (2002). Managing project uncertainty: from variation to chaos. *MIT Sloan Management Review*, 43(2), 60.

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87-108.

Eckstein, J. (2014). Architecture in large scale agile development. In *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers 15* (pp. 21-29). Springer International Publishing.

Ferdinansyah, A., & Purwandari, B. (2021, February). Challenges in combining agile development and CMMI: A systematic literature review. In *2021 10th International Conference on Software and Computer Applications* (pp. 63-69).

Henriquez, V., Moreno, A. M., Calvo-Manzano, J. A., & San Feliu, T. (2021). Agile-CMMI Alignment: Contributions and To-Dos for Organizations. *Computer*, 54(12), 38-49.

MacMillan, M. I., & Ihrig, M. (2015). Managing your mission—Critical knowledge. *Harvard Business Review*, 25-34.

Neto, F. G. D. O., Horkoff, J., Knauss, E., Kasauli, R., & Liebel, G. (2017, September). Challenges of aligning requirements engineering and system testing in large-scale agile: A multiple case study. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)* (pp. 315-322). IEEE.

Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company*. Oxford: Oxford University Press.

Nyfjord, J., Bathallath, S., & Kjellin, H. (2014). Conventions for coordinating large agile projects. In *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers 15* (pp. 58-72). Springer International Publishing.

Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2018). Large-scale agile transformation at Ericsson: a case study. *Empirical Software Engineering*, 23, 2550-2596.

Pedreira, O., Piattini, M., Luaces, MR, & Brisaboa, NR (2007). A systematic review of software process tailoring. *ACM SIGSOFT Software Engineering Notes*, 32 (3), 1-6.

Peng Xu & Balasubramaniam Ramesh (2007) Software Process Tailoring: An Empirical Investigation, *Journal of Management Information Systems*, 24:2, 293-328, DOI: <http://doi.org/10.2753/MIS0742-1222240211>

SAFe (n.d.), [Retrieve] 2023-03-06, <https://www.scaledagileframework.com/about/>

Schleith, J., & Tsar, D. (2022, October). Triple Diamond Design Process: Human-centered Design for Data-Driven Innovation. In *HCI International 2022-Late Breaking Papers. Design, User Experience and Interaction: 24th International Conference on Human-Computer Interaction, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings* (pp. 136-146). Cham: Springer International Publishing.

Sengel, E., & Aktaş, N. (2022). Blend Or Not To Blend? What Faculty Member Think About Blended Learning? *Journal of Turkish Science Education*, 19(3).

Shenhar, A. J. (2001). One size does not fit all projects: Exploring classical contingency domains. *Management science*, 47(3), 394-414.

Sreenivasan, S., & Kothandaraman, K. (2019). Improving processes by aligning capability maturity model integration and the Scaled Agile Framework®. *Global Business and Organizational Excellence*, 38(6), 42-51.

Tolbert, P. S., & Hall, R. H. (2009). *Organizations: Structures, processes and outcomes*. Tenth Edition, Pearson International Edition.

Turetken, O., Stojanov, I., & Trienekens, J. J. (2017). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and process*, 29(6), e1796.

Wikipedia (n.d.), [Retrieve] 2023-05-26, <https://en.wikipedia.org/wiki/DevOps>

Wikipedia (n.d.a.), [Retrieve] 2023-06-06, https://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

Wikipedia (n.d.b.), [Retrieve] 2023-06-06, https://en.wikipedia.org/wiki/ISO/IEC_9126

Wheelwright, S. C., & Clark, K. B. (1992). *Revolutionizing product development: quantum leaps in speed, efficiency, and quality*. Simon and Schuster.

Appendix 1. Interview Guide

Common Questions

Introduction

- Introduce myself as a master's student at Chalmers University.
- Explain the aim of the interview and approximate the interview duration (45m – 1h).
- Ask for permission to record the interview to transcribe it for coding analysis.
- The result will be analyzed anonymously.
- Inform there is a right to not answer and stop the interview.

Opening

- Can you describe your job title, and roles & responsibilities?
- How many years have you worked at the case company and the organization?
- How many projects/initiatives are you involving at the moment?

Process & Way of Working

- Can you describe how your works are related to the NPD process?
 - Which phases are you engaging in?
 - What kinds of activities, tasks, and outputs do you have to perform?
 - Who are your customers?
 - Who do you usually communicate with in the phases?
 - Can you see that your customers engage in the process continuously?
- Have you trained SAFe? How many times?

Perspective

- What makes you challenge when you perform quality assurance through the process?
- Do you believe that is there any invaluable activities or tasks in the process?
- Do you believe the NPD process makes to increase business value?
- Do you believe the NPD process supports building quality to meet customers' need?
- Do you believe the current organizations are sufficient to support your projects?
- Do you believe that company employees have a sufficient quality mindset? For example?
- If an organizational structure is formulated based on supporting business areas or brands, do you think it can be efficient and effective?
- What do you think about the case company's leadership, responsibility and accountability?

Closing

- Ask about the possibility of contact later if needed.
- Ask recommended next interviewee.

Business Partner

- Does your input affect prioritizing projects during the PI planning?
 - Which factors affects to prioritize projects?
- Do you have an experience that your inputs made projects reprioritization?
 - What was it? Can you give any examples?
 - Why has it had to be re-prioritized?
 - Which factors affects to re-prioritize projects?
- Challenges
 - What are the challenges when you work as a business partner?
 - What are the challenges when you work through the NPD process?
- How does your works affect the NPD process? Can you describe any examples?
 - What kind of activities do you have in order to increase tentative initiatives?
- From the customer's perspective, what do you see as the most important thing?
- Have you heard any complaints from customers? Can you describe any examples?
 - If you heard complaints after delivering the product, who would you contact in order to improve and counteract them?
 - Who has an accountability? How to deal with the complaints?
 - Due to cross-functional work as well as NPD

Business Portfolio

- Have you experienced the NPD process used in a different way, project by project? How can the process be used differently?
- Have you experienced a project suddenly informing you that the project is already developing?
 - When did you know that?
 - Why was the project not detected in the screening phase?
 - If a project is not detected by your team, what affects the case company level?
- Can you describe how and when projects are prioritized? Is there any strategy?
 - What are the most important criteria when prioritizing projects
 - E.g., budget, time, scope and quality
- Do you have experience re-prioritize projects?
 - Why has it had to be re-prioritized?
 - Can you see any impact on project teams due to reprioritization?
- Do you believe the NPD process makes to increase business value?
- Do you believe that products are build quality through the process to satisfy customers?

Continuous Delivery

- Can you describe how your roles are related to one- and runtime processes?

- To support continuous delivery, what are your activities during the PI phase?
 - Do you support continuous development/integration/deployment?
- Can you describe which processes belong to the runtime delivery process based on a subway map? Are you in charge of updating a subway map?
- Do you have ownership to manage documentation/code standards?
 - What types of documentation and code do you have and manage?
 - Where are the standards stored?
 - Are the standards common at the case company?
 - Can you describe how documentation and code standards proceeded?
 - When the standards revise? When were the final revision date?
 - When you revise them, how to distribute them at the case company?
 - Do you have a focal point (liaisons) when merging feedback from each team to improve it?
- What are the differences between one- and runtime deliveries of the way of working?
- Challenges
 - What are the challenges when connecting development and operations?
 - What are the difficult activities when you work as an operational specialist?
 - What are the challenges when you work with SAFe?
 - Which phase is the most difficult when you support continuous delivery?
- Developers work both maintenance and operations after the delivery of product. What can you see about your pain points?

DI30000

- Do you work with the NPD process?
 - If yes, can you describe how your work are related to the process?
 - E.g., backlog-refinement, PI
- Do you work with DIA (Discover, Incubate, Accelerate) process in the Innovation projects?
 - Can you describe the process a bit detail?
 - Who is the owner of the process?
- Communication
 - Can you describe which teams and what topics do you communicate with?
 - Can you access the Confluence and Jira? Can you access the SharePoint?
 - Have you ever seen the subway process map?
 - Do you communicate with other business unit such as Consumer Services or Technology?
- Does Innovation Lab projects prioritize?
 - Do you have any experience of project (re)prioritization or customer requirement changes affecting developing products?

- Challenges
 - What are the challenges when you work at Innovation Lab and in USA?
 - What are the most difficult activities when you work at Innovation Lab?
 - What can you see as challenges when you design services?
 - What are the challenges faced when managing development and operations as product owners at the team level?
 - What are the challenges when you work with your WoW?
- If the case company asks the Innovation Lab to change the way it works, then what can be challenges?
 - Which aspects should be considered?

Product Owner

- Do you work a product owner role as a full-time job?
- Have you trained SAFe? How many times?
- Communication
 - Can you describe what topics you discuss with scrum masters and developers?
 - Can you describe what topics you discuss with RTEs?
 - Can you describe what topics you discuss with testing teams?
 - Can you describe what topics you discuss with project managers?
- Do you have any experience that dependency-affected developing products?
- Do you have any experience of project (re)prioritization or customer requirement changes affecting developing products?
- How do testing results affect the product development?
 - Does your team conduct product testing as well?
 - Does your team perform automation testing?
- Story
 - What kinds of information do you write on the storyboard?
 - Can you see the degree of progress when you check the storyboard?
 - Does Confluence enough to use as a system of development management tool?
- Challenges
 - What are the challenges when you work as a product owner?
 - What are the most difficult activities when you work as a product owner?
 - What can you see as challenges when developers work?
 - What are the challenges faced when managing development and operations as product owners at the team level?
 - What are the challenges when you work with SAFe?
- Do you believe SAFe way of working at the case company is mature and has enough governance?
- Do you believe the way of working at the case company is fully SAFe or mixed with a waterfall?

Project Manager

- Is it easy to arrange supporting by architecture, testing, and UX functions?
 - When do you contact those functions?
 - Do you contact them by yourself or do any governance exist in which those functions should be engaged in specific phases?
 - Do you think that testing process and outcome is transparent?
- Which phase is the most difficult when you manage projects?
- Does it differ follow the process based on project types?
 - If yes, why do you think that deviation happens?
 - Who made a decision that projects can skip the process?
- What are the challenges do you have when you work though the NPD process?
- What are the challenges of working with SAFe?
 - What kinds of dependency can you see and affects delivering your project?
 - Do you think that development teams are working in an autonomous way?
- What are the challenges when you work as a project manager?
- If customers want to complain after delivering the project, who are the complaints? E.g., BP? Maintenance team?

Quality Manager

- Can you describe how your project quality is assured and proceed through the NPD process from the start to the end phase, as well as gates and milestones?
 - Do you engage in all phases or specific phase?
 - What kinds of quality activities do you have to perform?
 - Do you control anything KPIs with respect to project quality assurance aspect?
- Do you have any experience that your R&R duplicate someone at the case company?
- Do you have any experience that you have seen process tailoring case?
 - Can you describe examples that you saw it?
 - Which phases & aspects made them tailor the process?
- Do you believe process tailoring affect outcome of project quality?
- Do you engage in backlog refinement and PI phase?
- What kinds of challenges can you encounter during the PI phase with regard to quality?
- Do you engage in any activities for test or release?
- Do you believe project (re)prioritization affect managing project quality? How?

Release Manager

- Can you describe how to set a release plan based on this type of information?

- Can you describe the upcoming release plan?
 - What are the challenges?
 - What do you discuss with your stakeholders in order to solve the challenges?
 - What are the important aspects from your side of product area level?
 - How many times releases are generally conducted in a yearly basis?
- Do you work the release manager role as a full-time job?
 - When did release manager role generate?
 - Are you a dedicated release manager?
 - Why specific business areas have dedicated release manager?
 - Do you be involved in every release if projects/initiatives are relevant to your business brands?
 - Do you have any experience that your works were conflicted with someone?
- Do you have an experience with a failed release?
 - If you have, can you describe the situation and what made it fail?
 - What was the next step after the failed release?
- Do you have any experience that dependency affected your release management?
- How do testing results affect the release phase?
 - Do you have your own E2E testing manager?
 - Once release is conducted and deployed, who takes over maintenance?
- Challenges
 - What are the challenges when you work as a release manager?
 - What are the most difficult activities when you work as a RM?
 - What are the challenges when connecting development and operations?
- To be released in an efficient and effective way, what kind of support and information do you need before and after release?

Release Training Engineer

- Do you work the RTE role as a full-time job?
- Have you trained SAFe? How many times?
- Do you have an experience with a failed release?
 - If you have, can you describe the situation and what made it fail?
 - What was the next step after the failed release?
 - What levels of release were achieved? For example, story, epic, capability, and WP?
- Communication
 - Can you describe what topics you discuss with product owners and scrum masters?
 - Can you describe what topics you discuss with RTEs and the release manager?
 - Can you describe what topics you discuss with the testing teams?
- Influence

- Do you have any experience that dependency affected delivering your project?
- Does project (re)prioritization affect release management?
- How do testing results affect the release phase?
- Challenges
 - What are the challenges when you work as an RTE?
 - What are the most difficult activities when you work as an RTE?
 - What are the challenges when connecting development and operations?
 - What are the challenges when you work with SAFe?

Solution Architect

- What is the difference between enterprise architect and solution architect?
- Can you describe how your work are related to the NPD process?
 - Which phases are you engaging in?
 - What kinds of activities do you support?
 - What kinds of documents do you request?
 - Who do you usually communicate with in the phases?
 - Do you engage in before every PI for every project?
 - What kinds of challenges can you see when you support?
 - Are both functional and architectural requirements managed?
- How does your work affect the NPD process? Can you describe any examples?
- Can you describe about AC approval?
- Can you define a work package, capability, and epic? What is the difference between the two?
- Do you have a strategy with regard to requirement break down?
- How often do you communicate in a formal and informal way?
- What kinds of challenges do you have when you work through the NPD process?
- Do you believe that any activities or tasks should be enhanced or weakened to meet customers' expectations?
- How does working with the SAFE model of project development affect architecture work
- Does your input influence prioritizing projects during PI planning?
 - Can you describe how and when projects are prioritized? Is there any strategy?

Testing Manager

- Can you describe the E2E testing process?
- Can you describe which phase do you engage in the NPD process?
 - Do you engage in E2E test by request from project manager or formally assigned?
 - What do you discuss with project teams?
 - How long does it take time to conduct E2E test?

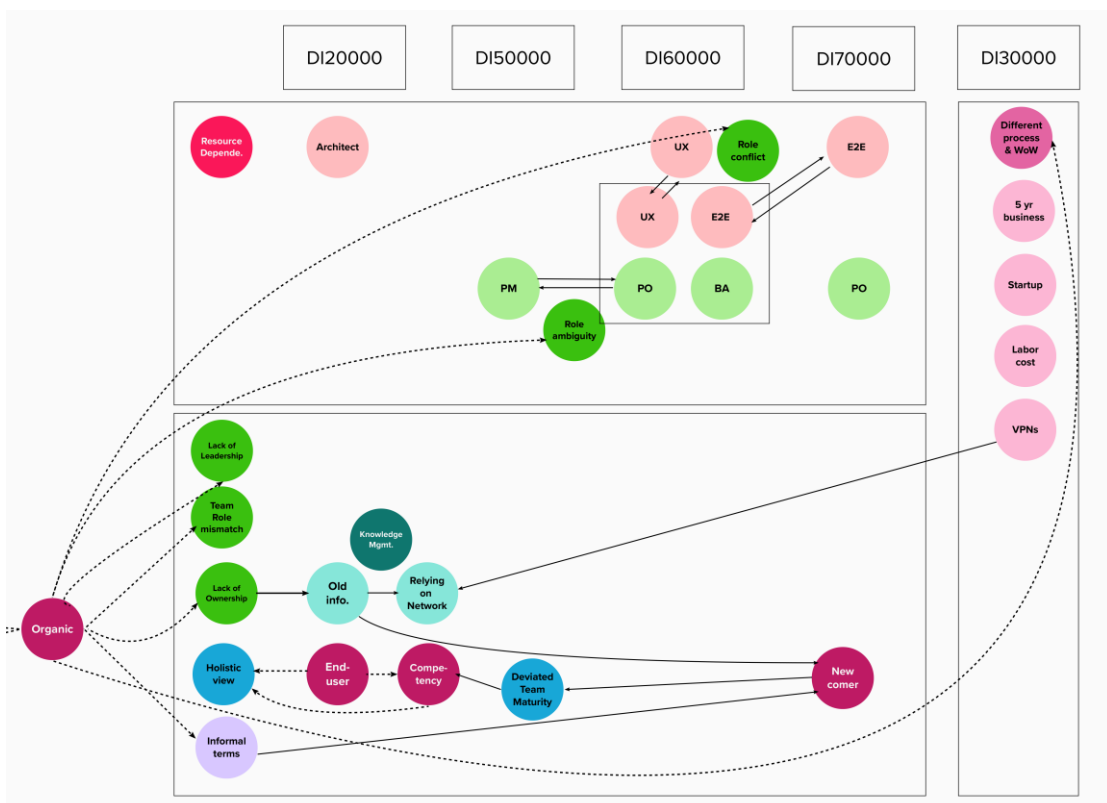
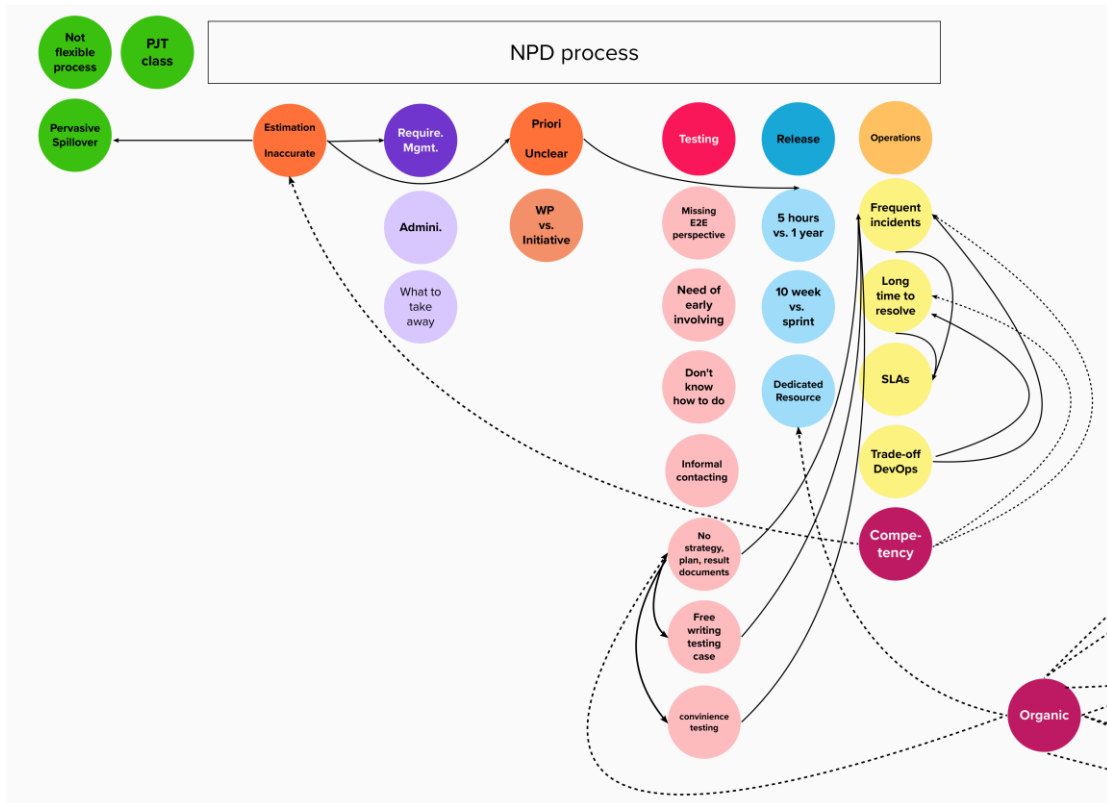
- Do you help and guide development teams for unit/component/system test levels, as well as those of development teams?
- Which phase do you generate test cases?
 - How long does it take to generate test cases?
 - How long does it take to conduct E2E test?
 - Do you conduct testing alone sequentially, with multiple testers sequentially, or with multiple testers in a parallel way?
- Do you work with an agile testing way?
- How do you frequently communicate with RTE? Can you describe it?
- Is E2E testing conducted based on project prioritization? What does prioritization affect your team?
- Do you think that continuous development and continuous deploy are well performed?
- What kind of documents do you generate and provide project teams?
- Do you have any specific acceptance criteria? If the E2E test did not meet the acceptance criteria, what did you do?
- Do all projects belong to your team are tested by you?
- What is the difference between your role and that of the E2E test manager in Consumer Service/Technology?
- Which teams conduct the other tests? For example, the load, performance, integration, and frontend/UI
- What is difference between you and verification and validation team?
 - Does your activity is overlapped with V&V team?
 - What is your E2E testing scope?

Benchmarking

- Process
 - Can you describe how does your company's NPD process look like?
 - Are there formal and informal gates in the process?
 - Do you believe that the process is flexible or not?
 - Does it differ follow the process based on project types?
 - How is your company's project classification?
- Can you describe your company's testing governance?
- Can you describe how your company organizational structure looks like?
- SAFe & DevOps
 - Does your company adopt a SAFe way of working or other methods?
 - Can you describe what development and operations look like?
- What kinds of evaluation models does your company adopt? E.g., CMMI
- How does your company collect end-users' feedback?
- How does your company organizational structure look like?
 - Is it functional based or matrix organization?
- Have you been experienced your role was conflict with someone before?

Appendix 2. The Method Coding Transcription

[Mural]



[Excel]

Category	Sub-category	Issue	Root Cause	Recommendation	
Process	Process tailoring	Improper two phases	NA	Merging Screening & Intake	
	Project classification	Inflexible process	NA	Flexible process	
Process Governance	Requirement Mgmt.	Bureaucratic AC	Centralization	Decrease formalization	
	Resource allocation	Missing E2E perspective	Dependency	Formalization	
		Informal resource securing	Dependency	Formalization	
	Estimation	Inaccurate estimation	Governance/Competency	CMMI	
	Prioritization	Unclear prioritization	Governance/Competency	CMMI	
	Spillover	Pervasive spillover culture	Governance/Competency	CMMI	
	Testing		No codified strategy/plan/result	Governance	Formalization
			Not mandatory writing test case	Governance	Formalization
			Convenience testing	Governance	Formalization
	Release		5 hours --> 1 year	Governance/Competency	CMMI
			Between waterfall and agile	Decentralization	CMMI
	Operations		Frequent incidents	Governance/Competency	CMMI
			Long time to resolve issues	Governance/Competency	CMMI
			Non-fulfillment of SLAs	Governance/Competency	CMMI
			Lack of focusing on operations	Governance	CMMI
Organization	R&R	Project vs. Product E2E tester	Governance	Formalization	
		PM vs. PO	Governance	Formalization	
		Mismatching between role & team	Decentralization	Job observation	
		Lack of ownership	Decentralization	Job codification	
		Lack of leadership	Decentralization	Formalization	
	Knowledge Mgmt.		Old information	Decentralization	Formalization
			Lack of codified information	Decentralization	Formalization
			Relying on network	Decentralization	Formalization
	Misalignment	Different WoW & Process	Decentralization	Formalization	
	Informal terms		Informal team names	Decentralization	Formalization
			Deviated definition of terms	Decentralization	Formalization
	Holistic view		Lack of end-user perspective	Customer	CMMI
			Deviated team maturity	Newcomers	CMMI

Appendix 3. Self-completion Questionnaire of SAFe Maturity

Maturity Level	Practice	Description of Practice
Level 1	Reflect and tune process	Holding retrospectives at regular intervals of the development process. The objective of this practice is to overcome process challenges that have been faced thus far. [1]
	Collaborative Planning	Collaborative Planning encourages all stakeholders to come together during the planning phase. The collaborative efforts increase visibility, loyalty, and acceptance and buy in from all stakeholders. [1]
	Empowered and motivated teams	Supervisors must empower and equip the development teams with the authority to make decisions on their own. This authority helps motivate the team members to solve problems and tasks on their own. [1]
	Collaborative Teams	Team members must communicate and cooperate with each other and other teams. [1]
	Coding Standards	Collaboration through code by creating a common language (coding standards) among developers. Coding standard increase the comprehension and ease of sharing code among team members. [1]
	Knowledge Sharing	Knowledge sharing tools enhance collaboration by helping to document and maintain the information and knowledge exchange among team members. Knowledge sharing tools can be electronics (e.g. wiki, blogs) or simple whiteboards and walls. [1]
	Task volunteering	During the planning meeting the developers should volunteer for tasks, rather than tasks being assigned to them by managers. This practice increases motivation, job satisfaction and quality of performance from team members. If there are no volunteers, the team should take collective responsibility to complete the task. [1]
	User stories	An agile practice where software requirements are formulated as one or two sentences in the every-day language of the user. Users stories are a quick way of handling customer requirements with minimal amount of documentation. [1]
	Acceptance testing	Acceptance tests are functional tests that verify that the system implements the story as intended. To avoid large volume of manual tests they are automated wherever possible. [2]
	Customer commitment to work with development team.	The customer has to commit to working with the development team in order to establish collaboration with development team. [1]
Level 2	Evolutionary requirements	Requirements should be evolving iteratively rather than being fully developed in one major specification effort. The requirements in agile tend to evolve and change based on customer feedback and thus this agile practice is important. [1]
	Smaller, more frequent releases	Agile teams should deliver smaller and more frequent releases to customers and end users. This provides increased responsiveness and reduces risks to the enterprise. This practice does not specify how often an organization should schedule a release which is the purpose of a similar practice at Level 4. [3]
	Requirements Discovery	Agile practice which is used to better understand what needs to be build and why. There are variety of software requirements techniques which are recommended and can be used by teams e.g, Spikes, Use-Case modeling, User Experience Mock-ups, etc. [2]
	Continuous delivery	Delivering software in small iteration at regular intervals. These regular intervals ensure that the team has to divide the product for each iteration. At this level the duration of the iteration is irrelevant but it is later addressed at level 4. [1]
	Two-level planning and tracking	This agile practice suggests development of iteration planning and release planning. Iterations plans are with short term focus and are estimated with great precision and confidence. Releases have long term focus and are more coarse-grained, less comprehensive and less precise. [2]
	Agile Estimating and Velocity	Agile estimating is a practice for estimating the workload to be delivered and is critical to the productivity and reliability. Estimating is done using story points. The number of stories points a team delivers per iteration defines the team velocity. This velocity estimate can then be used for estimating schedule and estimating cost. [3]
	Release Planning	Release planning is the seminal, cadence-based synchronization point of the Agile Release train (ART). It is a routine, program-scale, face-to-face event which typically includes all members of the program. It takes place over a two-day period at each PSI boundary. [4]

Maturity Level	Practice	Description of Practice
	The define/build/test team	A cross functional group of individuals that has the ability and authority to Define (elaborate and prioritize requirements and design the solution), Build (write the code) and Test (run tests cases) – all in a short iteration time-box. The define/build/test team is self-organize, empowered and self-managing and thus depends on practices from lower level. [4]
	Software configuration management	SCM tools help control the different versions of the software being developed. [1]
	Tracking iteration progress	Agile practice which is concerned with the team having the means by which they can measure the progress of the development effort within each iteration. This concept does not dictate a particular method to fulfill this tracking but its latter defined at level 4 (Daily progress tracking meetings). [1]
	No big design up front	Agile practice which ensures that the product is being developed using an evolutionary approach. BDUF is where a “big” design is created before coding and testing takes place and is typical for waterfall development process. In agile design occurs throughout the development process. [1]
	Product Backlog	The product backlog is a repository for all the upcoming work which is anticipated to be delivered. The product backlog can be utilized at different levels of granularity, e.g. team, program and portfolio backlog. [2]
	Customer contract reflective of evolutionary development	The customer understands the evolutionary nature of software development and the contract reflects this evolutionary approach. This practices prevents the contract to define the dates when milestones should be completed but its rather reflecting of the evolutionary approach. [1]
Level 3	Regular reflection and adaptation	Teams adopt regular reflection and adaptation at each iteration and release levels. These reflections are accompanied by quantitative assessment which measure iteration and release metrics. [3]
	Risk driven Iterations	Risk driven iterations help tackling risk elements as early as possible. Mitigating these risks early ensures that the project team does not spend a considerable amount of time building a system that they cannot complete. By catching these issues, the development becomes more effective. [1]
	Planning features not tasks	Customer expresses their needs in terms of features, so when the feature changes the impact on the related tasks is minimized. The planning should be done in terms of features in order to prepare the development process for Client Driven Iterations at Level 4. [1]
	Roadmap	Agile practice used to establish alignment across all the teams involved in the ART while also providing predictability to the deliverables over an established timeline horizon. [4]
	Mastering the iteration	The base construct of agile and iterative development is the iteration- the ability of a team to create working, tested software in a short time-boxed interval. The aim of this agile practice is effective production of an increment of working software at each iteration. [3] In essence, this practice validates of the effectiveness of the practices established on lower levels.
	Software Kanban Systems	SAFe suggests the development and implementation of Kanban systems for business and architectural portfolio epics. The Kanban system describes four queues that an epic passes through on the way to implementation: Funnel, Backlog, Analysis, and Implementation. [4] The Kanban Systems are used for visualizing workflow, limiting the work in progress, measuring and managing flow, making process policies explicit and using models to recognize improvements. [2]
	PSI/Release	PSI is a development time box (super sprint) that uses cadence and synchronization to facilitate planning, provide for aggregation of newsworthy value, and provide a quantum unit of thinking for portfolio level consideration and road mapping. The goal of PSI/Release is continuous integration and system validation and reducing the risk of deferred integration. [4]
	Self-organized teams	Team is empowered to make decisions without waiting for approval from management. Teams are cross-functional, roles and responsibilities of team members are not or loosely defined and the whole team is responsible for delivery of working software. The importance of self-organizing teams is highlighted in the Agile Manifesto which states that the best architectures, requirements and designs emerge from self-organizing teams. [1]

Maturity Level	Practice	Description of Practice
	Frequent face to face communication	For establishing efficient and effective development process frequent communication among team members is necessary. [1]
	Continuous integration	Continuous integration is an agile which encourages members of a team to integrate their work frequently. It is preferred that each integration is verified by an automated build tool in order to detect any integration errors as quickly as possible. [1]
	Continuous improvement (Refactoring)	Refactoring is an essential practice to be adopted at level 3 because of the evolutionary development process assumed at Level 2. Refactoring involves rewriting the code to improve its structure while preserving the behavior. In general refactoring focuses on removing code duplication. [1]
	Unit tests	Unit tests are code procedures used to validate that the individual units of source code are working properly. A unit of source code is the smallest testable part of an application. It provides a strict, written contract that the code must satisfy. It is recommended that unit tests are automated. [1]
	Agile Release Train	ART is a long-lived team of agile teams that serves the program-level value delivery mechanism in SAFe. Release trains are organized around the enterprise Value Streams and ART aligns teams to common mission, schedule and cadence which helps implement continuous product development flow. [4]
	DevOps	An agile practice that is used to ensure a faster flow of value to the user by tighter integration of development and operations. This is accomplished by integration personnel from operations into the agile teams or by continuously maintaining deployment readiness. [4]
	Vision, features	Describes the stakeholder's view of the solution to be developed in terms of stakeholder's needs and proposed features. It captures the essence of the envisaged solution in the form of high-level features, non-functional requirements and design constraints, and provides an overview of the system to be developed. [4]
	30% of level 2 and level 3 people	Cockburn people levels are directly related to the amount of experience the developer has. Cockburn identified three levels of understanding when they approach new material and has argued that a person level of understanding is directly linked to his experience in the domain. At Level 3 the team needs developers who can handle new unexpected problems and that is why Cockburn Level 2 and Level 3 people are needed. [1]
Level 4	Client driven iterations	The client is in control and can request and prioritize features per iteration. The client steers the project, iteration by iteration, requesting the features that are of highest business value to them. [1]
	Continuous customer satisfaction feedback	Continuous customer feedback is crucial to ensure the customer is satisfied with what is being developed. If customer feedback is only sought at the end of the project, then there is a significant risk that what has been developed is not what the customer needed. [1]
	Lean requirements at scale	For larger organizations, it becomes increasingly more difficult to make the whole team working toward a common purpose. There is a need to create a scalable requirements pattern consisting of vision, roadmap and just-in-time elaboration as a method that brings the benefits of agility to larger scale teams. [3]
	Smaller and more frequent releases	This encourages organizations to keep small timeframe for the releases and limit them to 8 weeks. Usually building a release will include multiple iterations. Having shorter releases helps the development process to embrace change. [1]
	Adaptive planning	Adaptive planning delays developing the iteration details until immediately before the following iteration, and therefore incorporates all the feedback obtained including what is learned during previous iteration. Adaptive planning helps organizations embrace change because the focus shifts from following a plan to adaptive planning based on latest feedback. [1]
	Intentional architecture	Intentional architecture is a set of purposeful, planned architectural initiatives to enhance solution design, performance and usability and provides guidance for inter-team design and implementation synchronization. [4] Architecture that has been planned to some extent, has been built incrementally, has emerged over the course of prior iterations or has evolved to adequately support the needs of the customers. [3]
	Managing highly distributed teams	Large corporates are distributed. They should be managed with proper communication and the necessary networking and tooling architecture. [3]
	Daily progress tracking meetings	This is a more specific version of tracking iteration progress practice which was introduced at Level 2. This practice emphasizes that the team should be informed on a daily bases regarding the status of the iteration. [1]

Maturity Level	Practice	Description of Practice
	CRACK Customer immediately accessible	The customer is Collaborative, Representative, Authorized, Committed and Knowledgeable. At this level the concern is not with the location of the customer but rather how responsive they are. [1]
	Customer contact revolve around commitment of collaboration	The customer agrees to contract the degree and amount of collaboration and not the requirements and features. This is one of the ultimate factors that enables organizations to embrace change. [1]
Level 5	Low process ceremony	Lower process ceremony allows for being responsive to changes. The changes don't have to be approved by a least 3 levels of management. Process ceremony is also the level of paperwork involved in the process. [1]
	Agile project estimation	Under the "Planning and Deliver Software Frequently" agile principles most practices are related to planning. Agile estimation is important since plans are only as good as the estimates they are based on. [1] SAFe uses several estimation techniques ex. Cost of Delay (CoD), WSJF, Story points for estimating. [2]
	Measuring business performance	Implementing a flexible, automate and meaningful BSC for the enterprise that measures performance in terms of efficiency, value delivery, quality and agility. The organization also measures team and ART performance. [3], [4]
	Ideal agile physical setup	Ideal agile physical setup helps establish the right environment for the agile software development to thrive in. The key in this setup is that the team is co-located and knowledge can be shared among team members instantly. [3]
	Changing the organizations	Organizations must make substantive changes to achieve full benefits of agile at enterprise level. These changes can mean changes in engineering practices, managing the impact on operations, reorganizations, etc. The executive sponsor has a critical role and his dedication and commitment to the process, as well as leadership by example, is the key factor in unlicking great benefits. [3]
	Test driven development	Software development technique that involves repeatedly first writing a test case and then implementing only the code necessary to pass the test.
	no or minimal number of Cockburn Level 1B or -1	Cockburn Level 1B and Level -1 developers have the least experience and are not able or willing to collaborate which can hamper the transition to agility. This is why their presence is discouraged at Level 5. [1]
	Concurrent testing	The practice of incorporating test-development, test automation, and test execution practices within the course of the iteration. Concurrent testing involves several Agile Testing strategies such as: Unit Testing, Acceptance Testing, Component Testing, System, Performance and Reliability testing. This practices is listed at Level 5 since all the tests should be automated, run frequently and all these efforts take long time to master. [3]
	Face to face interaction between develops and users	It is ideal to have not just the developers collocated but also have the customers and users in the same rooms. This ensures almost instant feedback and incredible communication. [1]
	Impact on customers and operations	Measuring the impact on sales, operations and customer due to the changes in the development model. [3]

[Main Source] Turetken, O., Stojanov, I., & Trienekens, J. J. (2017). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and process*, 29(6), e1796.

[1] Sidky, A.S. (2007) "A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework. PhD Thesis, Virginia Polytechnic Institute."

[2] Leffingwell, D. (2011) *Agile software requirements: lean requirements practice for teams, programs, and the enterprise*, Addison-Wesley Professional.

[3] Leffingwell, D. (2007) *Scaling Software Agility: Best Practices for Large Enterprises*, Addison-Wesley Professional.

[4] "Scaled Agile Framework (SAFe). <http://www.scaledagileframework.com> [5 May 2015]."

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF SERVICE MANAGEMENT AND LOGISTICS
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY