CHALMERS

The quest for quality when working scaled agile

A case study of the challenges and the best practices related to quality within software development

Master's Thesis in the Master's Programme Supply Chain Management

VIKTORIA ANDRÉASSON
SOPHIA JACONELLI

Department of Technology Management and Economics
Division of Science, Technology and Society
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019
Report No. E2018:123

MASTER’S THESIS E2018:123

The quest for quality when working scaled agile

A case study of the challenges and the best practices related to quality within software development

VIKTORIA ANDRÉASSON
SOPHIA JACONELLI

Supervisor Chalmers University of Technology: Erik Bohlin
Supervisors Knowit Insight: Teresa Thorsson

Department of Technology Management and Economics
Division of Science, Technology and Society

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

The quest for quality when working scaled agile

A case study of the challenges and the best practices related to quality within software development

MATER’S THESIS E2018:123
Department of Technology Management and Economics
Division of Science, Technology and Society
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone: + 46 (0)31-772 1000

Chalmers Reproservice
Gothenburg, Sweden 2019

Acknowledgement

This master thesis was our final step to completing the master program of Supply Chain Management at Chalmers University of Technology. The thesis was designed in collaboration with Knowit and constitutes 30 credits at a master level within Industrial Management Engineering. The concept of agile development was new for both of us. Therefore, the desire to conduct a research within the area of agile development was based on own interests to learn more about the concept. The journey has been interesting, educative and challenging and we will take all the valuable experiences from this autumn with us to the upcoming work life. We are also hoping that this research will contribute to inspiration for future research.

We would like to thank our supervisor Erik Bohlin at Chalmers University of Technology, for giving us the opportunity to write our master thesis at the Department of Technology Management and Economics. Furthermore, we would like to thank all interviewees that we had the opportunity to interview. Also, we would like to thank the supervisor Teresa Thorsson with colleagues at Knowit, especially Jenny Gorner who were very engaged in our master thesis. Lastly, we would like to thank each other for a good collaboration and an enjoyable semester together.

Sophia Jaconelli

Viktoria Andréasson

Sophia Jaconelli

Gothenburg, Sweden 2019

Abstract

The increased demand for faster deliveries, more customer-specific products, improved performance and increased productivity force organisations to change their way of working. Organisations must become more flexible to meet society's market demands. The adoptions of agile methods and principles have increased in popularity within organisations over the past years. The agile way of working is characterised by promoting self-organising teams, close customer collaboration, less documentation and reduced time to market.

The traditional quality assurance practices that have in the past proven to be efficient might not fit into the agile way of working since these occur late in the project process. In the agile way of working quality should be an inherent part of the agile practices and be involved earlier in the process. However, many organisations have today challenges to ensure quality when they are scaled agile through the whole organisation. This master thesis investigates the area of quality assurance when working agile on an upscaled level. Further, the thesis investigates which challenges organisations are facing that are related to quality when working scaled agile. The most important agile practices to ensure quality will also be examined in the report. A qualitative research method was conducted in order to collect data and answer the research questions. The empirical findings were conducted theough 17 semi-structured interviews with different people from different organisation.

The result of the research showed 12 main challenges related to quality when working agile on an upscaled level. In the report, the challenges that were found have been divided into: organisational related challenges, process related challenges and technology related challenges. The organisational related challenges reflect the challenges when transforming an organisation into the agile way of working. The process related challenges express that there are, for example, difficulties to break down traditional quality tools and standards, regulations and legal requirements into shorter sprints. Lastly, the technical related challenges describe that there is a lack of both agile metrics and test automatisation. Moreover, the research shows that there is a great variance of agile activities that are perceived to be important to secure and enhance the quality. The findings show that continuous integrations, program increment planning, retrospective, close customer collaboration, test automatisation and working with cross-functional teams that are responsible for the quality are the best activities in order to assure quality. Most of the organisations also argue that the quality have been increased after the agile way of working has been implemented throughout the organisation. In conclusion, recommendations of areas for further research that could be of interest to investigate are presented.

Keywords: Agile, Scaled Agile Development, Agile Practices, Agile Software Development, Quality, Quality Assurance, Software Quality.

Terminology

The words in this list below are marked (*) at the first presence in the report.

Backlog- Is a list of all the task and activities that need to be complete in the project through close customer collaboration. The list is changing and reprioritising constantly in order to adapt to changed customer requirements.

Definition of done (DoD)- It is a list of software requirements that must be met before a product increment consider done. It is created by the teams with respect to the customer requirements. DoD ensures that the whole team knows exactly what is expected from them.

Incremental development- This means that every new added functionalities of a product is build upon the previous version. In other words, the final product should not be delivered all at once. An increment is the sum of all the items completed during one sprint including the increments from previous sprints.

Iterative development- It is a fixed time period in which a predetermined number of tasks should be completed. The duration of each iterative period might vary from project to project.

T-shaped competence- All agile teams need to build upon T-shaped competence, which refers to deep knowledge within a specific area but at the same time include all necessary skills to accomplish a task.

User story- A user story describes the requirements of the system based on customer requests. The description only contains enough information for the developers to estimate the time and effort to implement it.

Disposition

Below is a reading guide to the eighth chapter of the report in order to facilitate an understanding of the structure and the content of the report.

Chapter 1
The first chapter of the report describes the background behind the issue to be investigated, the purpose, the research questions and the limitations of the report.

Chapter 2
The second chapter presents the theory of the literature review, which consists of traditional project management, agile way of working and quality assurance.

Chapter 3
Chapter three describes how the master thesis was conducted, including research strategy, research process, data collection and analysis of data. Moreover, a discussion regarding the quality and ethics of the research is presented.

Chapter 4
This chapter presents the empirical findings based on the interviews. The difficulties and the best practices that the interviewees experience associated with quality when working agile are also described.

Chapter 5
The analysis of the report is presented in chapter five. The analysis combines the empirical finding compared to the literature framework in order to answer the research questions.

Chapter 6
Chapter six discusses the findings of the report based on the analysis. First the challenges are discussed followed by the best practices to ensure quality when working agile.

Chapter 7
In this chapter, the conclusion of the research questions is presented. For research question one, recommendations on how to improve the identified challenges are also presented.

Chapter 8
The last chapter makes suggestions for further research based on the report that the researcher believes is of interest to investigate further.

Table of contents

1. Introduction

This chapter describes the background behind the issue to be investigated, the purpose, the research questions and limitation of the report.

1.1 Background

The increased demand for faster deliveries, more customer-specific products, improved performance and productivity force organisations to change their way of working (Bardhan & Krishnan, 2007). Organisations must become more flexible to society's increased market demands. Furthermore, today’s products contain more software and software development than before. Therefore, it has been increasingly important for organisations to deliver products and services fast to the market, before their competitors, and at the same time ensure high quality. Traditionally, quality assurance methods occur late in a project process. It is difficult to estimate in forehand how the final system or product will be visualised. Besides, it required a very detailed specification of requirements from the customer right from the beginning. Therefore, it is a need for process flexibility because of the increasing project complexity.

Agil means mobile and the essence of the agile methods is to make the development more flexible and easier to operate (Cohen, Lindvall & Costa, 2004). The agile methodology was a response to traditional system development methods. The traditional methods were considered bureaucratic, hard controlled and demanded extensive documentation. Instead, developers wanted to create self-organising teams, close collaboration with the customer and more flexibility. The adoption of agile methods and principles has increasingly gained popularity within organisation over the past years. The agile way of working is characterised by promoting self-organising teams, customer collaboration and less documentation (Ahmed, et al., 2010). Furthermore, built-in quality is a daily activity to embrace quick responses to change. When transferring from traditional project management into an agile way of working, Joseph et al. (2015) claim that “*quality assurance departments have been quality gatekeepers rather than actively engaged in the ongoing development and delivery of quality software*”. Quality should be an inherent part of the agile principles and an agile team should use the agile methods in order to ensure the quality.

However, surveys have indicated some challenges due to quality assurance when changing from traditional management (Bhasin, 2012). According to the author, organisations strive to achieve built-in quality assurance systems but organisations have not been able to secure the quality. Today, the agile approach is using face-to-face communication rather than traditional documentation, which therefore hand over the quality responsibility to the teams. By transferring the responsibility to each team, quality assurance could vary. Moreover, the surveys have shown that transformation on the project level is fairly simple. However, organisations face their next challenge of scaling up the agile throughout the whole organisation (Vaidya, 2014). Therefore, this master thesis will investigate the challenges that organisation facing due to quality when working scaled agile.

1.2 Purpose and Research Questions

The purpose of this master thesis is to investigate how different organisations are working with quality when they work scaled agile. The research will analyse which challenges there are when organisations have become scaled agile, which quality activities that are performed and which activities that improve quality. Furthermore, the report will present theoretically anchored recommendations to the problem areas identified. Finally, the outcome of the report is expected to provide a basis for further investigations within this the area.

Based on the background and the purpose, following research questions have been selected:

- ***RQ1***: *What are the challenges due to quality for organisations that work scaled agile?*
- ***RQ2***: *Which solutions could be implemented to overcome these challenges?*
- ***RQ3***: *What are the most important agile practices in order to ensure quality?*

1.3 Delimitations

The thesis will be focusing on organisations with departments that are working scaled agile, in other words, where there are more than one agile team. Also, only organisations that are working with some kind of software development in their end-product will be analysed. The interviews were held with quality managers, developer managers or similar professionals from the quality department at respective organisation. Most of the interviews will be with only one or few representatives from the organisation. This can indicate that the collected data from this research will reflect individual or bias views of the subject. Furthermore, no possible solutions to the identified challenges that the report will investigate will be implemented.

2. Theory

This chapter present the theory of the literature research. In more specific the theory that categorise and describes traditional project management, agile way of working and software development quality. Each category consists of sub-categories, which are derived to create a broader base of knowledge in order to support the investigation.

2.1 Traditional project management

A traditional project is often divided into three different parameters: time, cost and result, the so-called project triangle as visualised in figure 2.1 (Gustavsson, 2014).

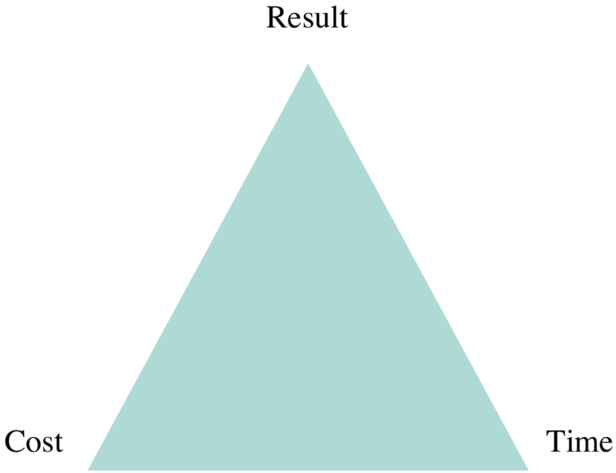


Figure 2.1: The project triangle within traditional project management (Gustavsson, 2014).

In traditional project management, the focus has been on time and cost, which often leads to neglecting the result (Gustavsson, 2014). However, projects still have a tendency to be delayed and exceed the intended budget. When a project is short of time, new staffs are usually involved to solve the problem. Existing staff must then educate the new staffs, which creates a bad circle and even more delay and costs. Additionally, traditional project management has difficult to manage the higher demands and becoming more adaptive, as the great willingness of projects lies in documentation and strict hierarchical organisation that makes it difficult to be flexible.

2.1.1 Waterfall method

The waterfall method is the oldest and the most executed software development method among the traditional project management (Huo et al., 2004). The method was introduced in 1970 and the name of the method comes from the fact that it is a sequential falling method, similar to a waterfall, where a process is divided into different steps. Each step of the process must be completed before the next step can begin, even though these sometimes overlap in practice. The method consists of five falling steps and once a step is completed, it is not possible to return to the previous one (Sommerville, 2007). As figure 2.2 visualise, the first step is to develop a requirement analysis, followed by design, implementation and testing.

Before the product can be released, a maintenance analysis has to be performed to ensure that the product meet all requirements documented from the previous step.

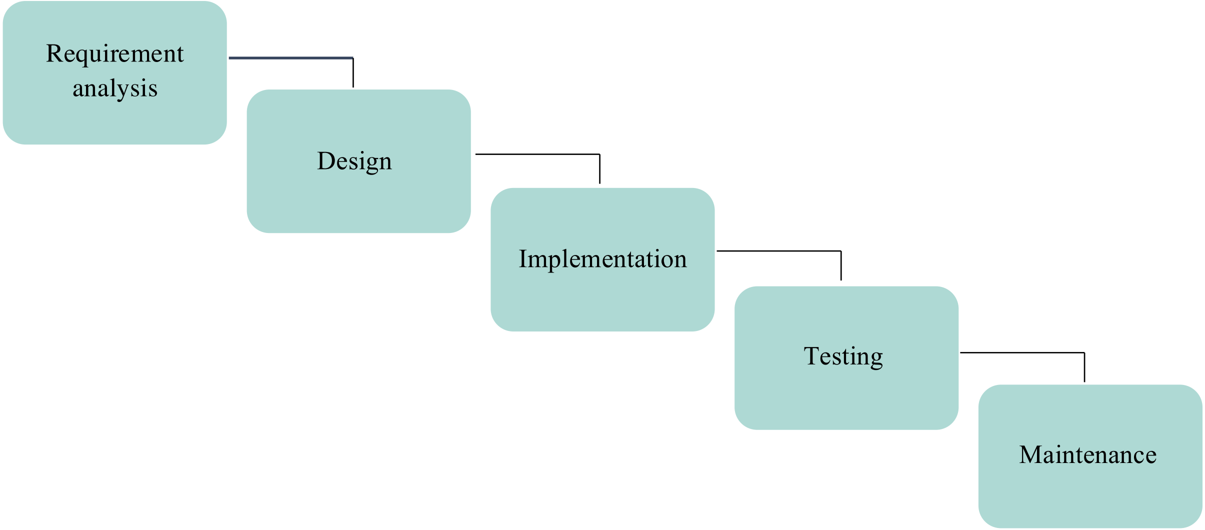


Figure 2.2: The waterfall method (Sommerville, 2007).

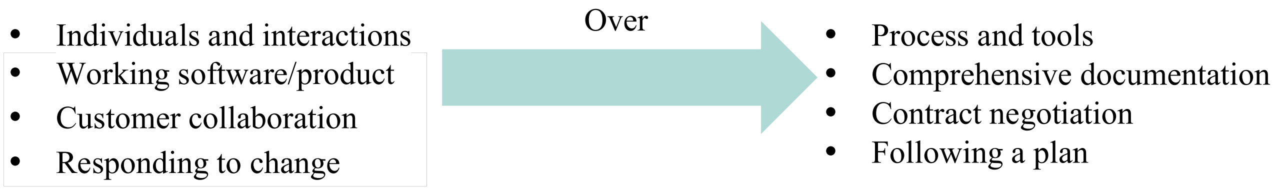
The waterfall method is suitable when requirements are very clear (ibid). Much detailed specification is needed to design, build and install requirements. However, the method relies on assumption of the future that are hard to know in advance. An issue often arises during the process and the users rarely know exactly what they want at the beginning of a project. Software is often a unique development process and rarely standardise, which is wanted within the waterfall method. The technology, the market and the customer need changes faster than the waterfall method usually can handle. Therefore, the technology, the market and the customer request a new framework to avoid standardised way of working. Especially, within product development when an entirely new product or system are being released.

2.2 The agile development

The agile methodology is a collection of values and principles, which comprises the ability to increase flexibility and quickly adapt to changes (Cooke, 2012). The agile development has emerged over the past two decades and derives from the software development industry. There was a need for higher process flexibility as a result of the increased project complexity (Wysocki, 2011). The agile methodology was created in order to provide a structure to develop and sustain complex software systems in an environment of rapidly changing requirements to enables customised solutions.

2.2.1 The Agile Manifesto

The term agile was first established in 2001 when leading people from the software development industry gathered and created a common platform named the Agile Manifesto (Beck et al. 2001). In the Agile Manifesto there are four core values that are presented:



- Individuals and interactions
- Working software/product
- Customer collaboration
- Responding to change

Over

- Process and tools
- Comprehensive documentation
- Contract negotiation
- Following a plan

The core values emphasise that even though the previously core values contributes with business value, there are more value in the new once (ibid.). The purpose behind the Agile Manifesto was to create a framework to increase the ability to develop the required products and at the same time minimise waste in terms of extra work regarding, for example, longe planning phases. The Agile Manifesto could be seen as a reaction against the more traditional methods, there a comprehensive plan first has to be created before the actual work can begin. In the software industry, the systems or the IT-product that are developed are usually unique and customised. With the traditional methods, it was extremely difficult to estimate in forehand how the final system or product would be visualised. Besides, it required a very detailed requirement specification from the customer from the beginning. The agile approach, on the other hand, can be used when the solution is not clearly defined and specified.

2.2.2 Agile methods

According to Gustavsson (2011), to be agile means to constantly renew and evaluate the project to find new opportunities for improvements and ways of working in order to succeed in a rapidly changing environment. The core in the agile way of working is to adopt iterative* and incremental* development to increase the flexibility to deliver products and services quicker to the market. The work should be team based where the responsibility is distributed to the teams. This in turn requires capable individuals across different disciplines and a non-hierarchical management. Furthermore, the team should have a close collaboration with the customer to create business value.

Agile methods refer to the agile philosophy and describe practices and principles that follow the agile manifesto (Shore & Warden, 2007). There several different agile methods to select from when transferring from traditional project management (Cohen, Lindvall & Costa, 2004). However, they are all formed from the same values and principles and which method that is most suitable for an organisation depends on the characteristics of the organisation. The agile methods Scrum, Extreme Programming, Kanban and Lean Software Development will be described further in the text below.

Scrum

Scrum is one of the most commonly used agile methods (Zelkowitz, 2004). The aim of the method is to handle complex process development in an unpredictable environment. Furthermore, the method is focusing on the relative effectiveness in order to make improvements within the organisation based on previous experiences and information that the organisation actually knows. The Scrum method is divided into three iterations as follows:

- *Pre-sprint planning*- In this initial sprint the scrum team is created, and it is determined which types of tools and competences that is needed (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). After the team is created, a product backlog* is defined consisting of existing requirements. These requirements are divided into tasks that will be completed during the next sprint. The time to implement each task is estimated in order to prioritise the tasks. Prioritising is an important step since the product backlog is constantly updated with new requirements that need to be prioritised. The pre-sprint planning also includes a higher level of "abstraction", it is therefore important for the team members to identify and have a common perception of the sprint goal (Cohen et al., 2003).

- *Sprint*- In this phase, the team members choose one task to develop during a number of sprints (Abrahamsson et al., 2002). One sprint is a time period between one week and up to one month. Each sprint includes a collection of requirements, analysis, design, development and delivery of the task. Scrum meetings are performed continuously during the sprints. The meetings are short and are usually held every morning, so called daily scrums. The purpose with these meetings is to assure that the project developes in the right direction. The meetings also facilitate communication between the team members and stakeholders to ensure that no problem arises (Cohen et al., 2003).

- *Post-sprint meeting*- The third and final sprint, the post-sprint meeting, begins when the system is ready for delivery (Abrahamsson et al., 2002). No new features can now be added, instead, the implementation of the system starts. In addition, the post-sprint meeting enables the team member to analyse the progress of the implemented project (Schwaber & Sutherland, 2013). The evaluation of the previously sprint constitutes the foundation for the next sprint.

A Scrum team must be cross-functional and consist of people with different competences (Cohen, Lindvall & Costa, 2004). The team model within Scrum is designed in order to optimise the flexibility, creativity and the productivity. As follow, the different roles are described:

- *Scrum master (SM)*- The SM is responsible to ensure that the work is in line with the scrum practices, values and rules (Abrahamsson et al., 2002). However, the SM is not a traditional project manager, instead the role comprises to support and coaching the scrum team.

- *Product owner (PO)*- The PO is responsible to maximise the product value and the work of the team (Schwaber & Sutherland, 2013). The PO is ultimately responsible for leading the project and to update and manage the product backlog.

- *Scrum team*- The scrum team is a self-organised team, without no formal leader and titles (Abrahamsson et al., 2002). In order to be efficient, the scrum team should be small and not exceed more than nine team members. If the team includes more than nine members, it is recommended to split the team to maintain productivity. Additionally, the scrum teams are cross-functional and contains people with different competences (Schwaber & Sutherland, 2013).

- *Customer*- The customer sets the requirements for the final product and provides information to the product backlog (Abrahamsson et al., 2002).

- *Management*- The management team create goals, specify requirements and makes final decisions (ibid).

Ceremonies within Scrum

The ceremonies or activities of the Scrum framework have a time limit with a purpose to create regularity (Schwaber & Sutherland, 2013). In figure 2.3 some examples of the ceremonies within Scrum are visualised.

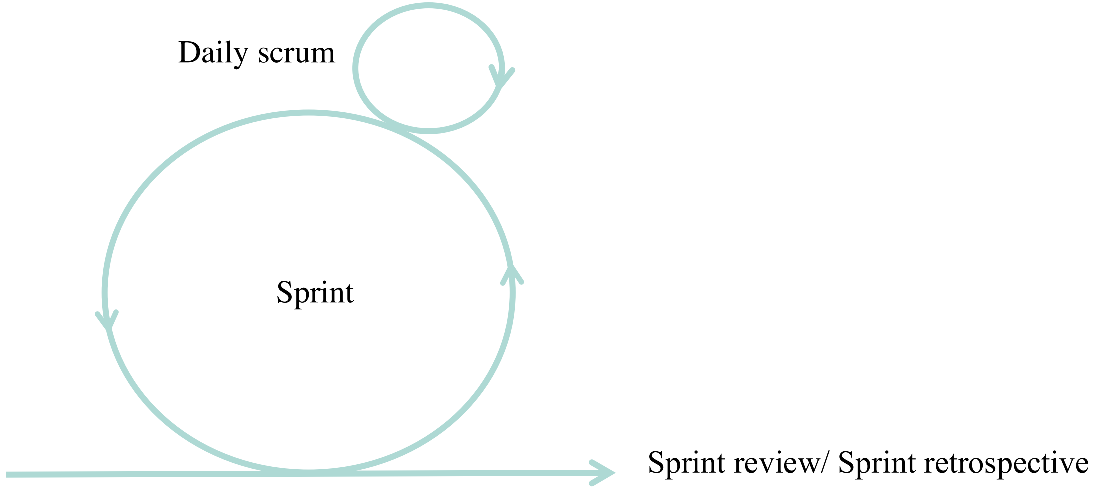


Figure 2.3: Cermonies within Scrum (Schwaber & Sutherland, 2013).

- *Daily scrum*- The activity is performed on a daily basis and is time-limited to 15 minutes (ibid). During the meeting the teams evaluate their work, inspect the progress against the sprint goal and decides what is needed to be done until next daily scrum.

- *Scrum-of-scrum (SoS)*- Scrum-of-scrum is a coordination meeting between different teams in large-scale organisations (Paasivaara, 2012). The meeting can be held every day or a few times every week depending on the need of the project. It is only one member from each team that is chosen to participate during the meeting. Relevant discussion of what each team have done since last time, what they will do until next time and if issues have arised are discussed during these meetings. The same as for daily scrums, it is recommended that the SoS meetings last only for 15 minutes in order to be efficient.

- *Sprint Review/Sprint Demo*- A sprint review is performed in the end of each sprint in order to review the increment and adjust the product backlog (Schwaber & Sutherland, 2013). This activity is a collaboration between the scrum team and the stakeholders to evaluate the performed work and examine what is needed to be done in the next sprint.

- *Sprint Retrospective*- This activity is an opportunity for the scrum team to review their performed work and create a plan for improvements to the next sprint (ibid). The retrospective is executed after the sprint review and before the next sprint planning. During a retrospective the scrum team tries to improve the product quality for the next sprint by adjust the definition of done* in an appropriate way.

Extreme programming (XP)

XP emerged as a response to the problems and complexity of the traditional project management (Abrahamsson et al., 2002). This agile method is not as common anymore but many activities from XP have been adapted into other agile methods. This method is focusing on the programming parts of the development process, in contrast to Scrum, which focusing on planning and organising the project. The activities within XP are performed usually for two weeks iterations. At the end of the iterations, the team delivers software functionalities. Within XP, the team should be cross-functional and consist of team member with different knowledge.

According Beck and Gamma (2000), XP could be summarised into four key values: communication, simplicity, feedback and courage. These values are described in 12 different principles:

1. *The planning game*- An interaction between the customer and the developer in order to develop the functionalities that are required in the system.
2. *Small releases*- In order to be flexible and adaptive to changes, new releases of the system occur often, from a daily up to a monthly basis.
3. *Metaphor*- One or several metaphors create a foundation of the system, which are constructed by the parties involved within the project.
4. S*imple design*- The developers should use a simple design as possible to prevent duplication of work.
5. *Tests*- The developers first have to write a test code that must the accepted before initiate.
6. *Refactoring*- The developers should continuously evaluate if the existing code can be improved in order to maniate a simple design.

7. *Pair programming*- The code is written by two people in order to improve the quality and reduce defects. One person is responsible for implementing the function, while the other one focusing on test improvements.
8. *Collective ownership*- All developers have a collective responsibility for the system and are allowed to change the code if it leads to improvements.
9. *Continuous Integration*- After a new function is approved, it should be integrated as soons as possible into the system.
10. *40-hour weeks*- A work week should not exceed more than 40 hours and every iteration should be planned so no over time is required.
11. *On-site customer*- The customer should be a part of the development team and be available at all time if issues arise.
12. *Open workspace*- The team should work together in one room, with the developers setting in the middle.

Kanban

The Kanban method aims to design, manage and improve the workflow (Hammarberg & Sundén, 2014). The method origins from Toyota's production philosophy and is from the beginning a material supply strategy. The word kanban means "visualise card" and therefore, the work should be visualised in order to find constraints and improve the existing processes. Organisations that have applied kanban are using kanban boards in order to visualise the work and these boards could be either be physical or virtual.

It could be stated that kanban in software development is more a set of tools rather than a method itself (ibid). The concept of kanban is to let the need, not the capacity, control the process. Kanban could be combined with all the other agile methods and is build upon four principles:

- Begin where you are
- Visualise your work
- Constraint the amount of work
- Improve the workflow

When using Kanban for software development the focus is mainly on flow efficiency (Hultgren & Lyhammar, 2018). The purpose of kanban method within the agile development is to minimise multitasking for the teams in order to make the software development process more efficient. This could be done by decide a limited number of task and project within the development process, for example, how many projects that at the same time could be analysed, coded or tested. The team cannot begin with a new task before there is an ”empty spot”. From this way of working, a higher level of collaboration between the teams is achieved due to that teams have to help each other to get their work finished before starting off with new tasks or projects.

Lean Software Development (LSD)

The LSD is orientated from the Japanese car manufacturing industry in 1980's. This agile method is inspired from the lean production philosophy and are build upon seven principles (DSDM Consortium, 2008):

1. *Eliminate waste*- Everything that not create value to the end-product should be eliminated.
2. *Amplify learning*- Focus should be on learning and improvements.
3. *Decide as late as possible*- Make decision late as possible to be flexible to changes.
4. *Deliver as fast as possible*- Short business cycles are desire in order to be efficient.
5. *Empower the team*- Individual competence and learning is in focus.
6. *Build integrity in*- The system should have a coherent design and fulfill its purpose. It should be a balance between the customer and system view.
7. *See the whole*- View the system as a whole and not as different parts.

Within LSD, it is important to eliminate waste since it does not add value to the end-product, such as overtime and waiting (Liker & Meier, 2006). There are often delays in different part of a project, delays that will affect the customer. It is therefore recommended for the development teams to work in the same room to prevent delays and eliminate waiting time. Also, documentation is one example of something that not might necessarily adds value seen from the customers' perspective (Poppendieck & Poppendieck, 2003). However, certain project requires some types of documentations and then the recommendation is to keep it to a minimum.

2.2.3 Scaling agile

When an organisation wants to transform and implement the agile way of working principles, they often start the transformation on the team level (Ambler, 2010). However, to become fully agile, the transformation has to be scaled up throughout the whole organisation. According to the author, two different aspects have to be considered to succeed when scaling the organisation. First, make the scaling on project level in order to discover each team's individual challenges and second, scale throughout the whole department. Studies have shown that transformation on the lower level is fairly simple. However, organisations face their next challenge of scaling up the agile throughout the whole organisation (Vaidya, 2014). To become scaled agile across the entire organisation, Ambler (2010) focusing on five successful factors:

1. *The goal is to get better, not to become agile*- The main goal is to deliver the system in an efficient way, which could be achieved from a combination of agile and traditional principles.
2. *Have a continuous improvement plan*- First, the goal and strategies need to be identified and priorities in order to improve the plan continuously.
3. *Gain some experience*- Create one or more test pilot(s) to get experience and find risks before implementing it on the whole organisation.

4. *Explicitly manage your process improvement efforts*- The teams must constantly reflect on potential improvements and when achieving an important improvement, it is necessary to adapt it across the entire organisation.
5. *Invest in your staff*- Invest in training, education and coaching for all actors within the pilot project in order for them to understand the agile philosophy and why a transformation is needed.

There are three different scaling agile frameworks: Disciplined Agile Delivery, Large-Scale Scrum, and Scaled Agile Framework (Vaidya, 2014). Whereas the last one is the most common and used framework among organisations, and it will be explained more in detail in the next section. Choosing a framework depends on each organisation but commonly for all frameworks are that they are built upon the agile and lean principles.

Scaled agile framework (SAFe)

SAFe is a template for scaling agile principles and tools to organisations (Scaled Agile Framework, 2018). Dean Leffingwell developed it in 2011, as a result of a need for a framework that guides and helps companies to become scale agile. Scaling is done through so-called Agile Release Train (ART). According to Scaled Agile Framework (2018), ART "*is the primary value delivery construct in SAFe. Each ART is a long-lived, self-organizing team of Agile Teams, a virtual organization (5-12 teams) that plans, commits, and executes together*". Depending on the size of the company, SAFe operates in four different levels visualised in figure 2.4: portfolio, large solution, program and team. Even though the framework acts at different levels, it addresses some common factors such as: management of multiple agile teams, program-level planning, management and shared resources, specialised integration teams and management of work in progress (Armani, 2014). Organisations have experienced many advantages when implementing SAFe. Increased commitment, higher employee satisfaction, faster time-to-market, increased quality and increased productivity are some of the benefits that arise when using SAFe.

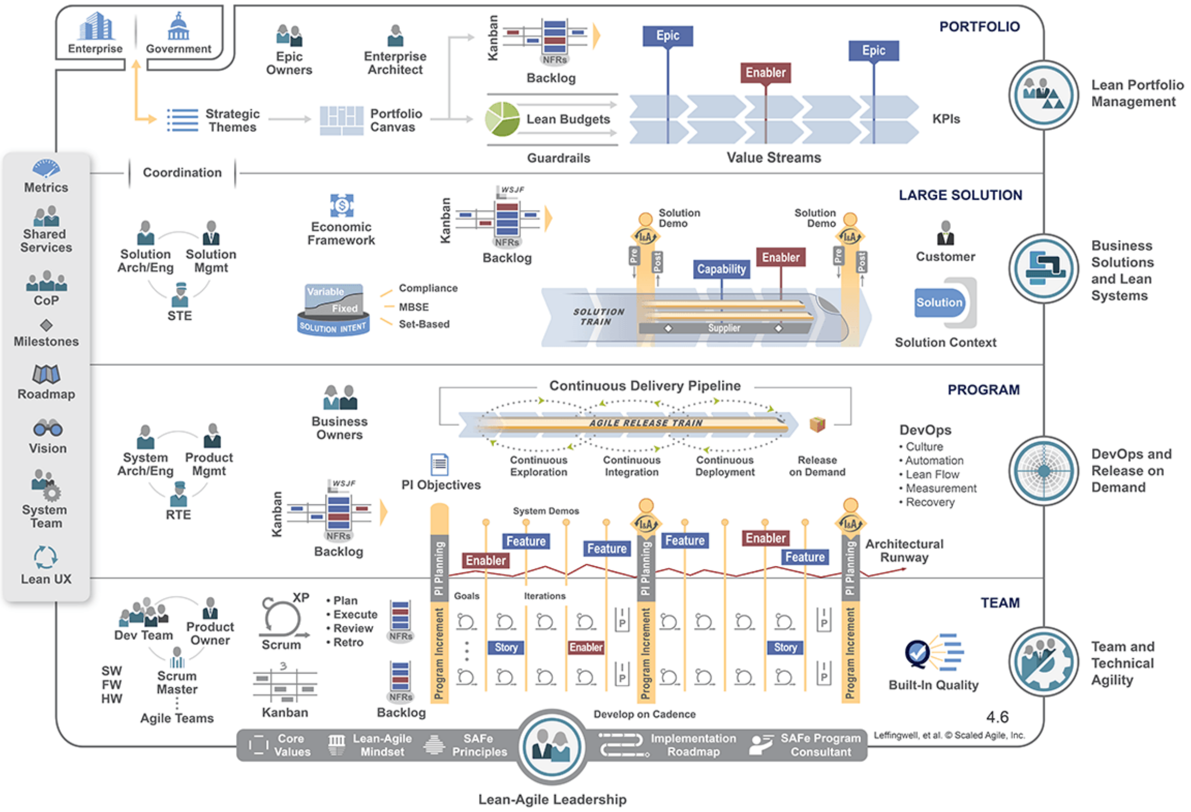


Figure 2.4: The template of SAFe (Scaled Agile Framework, 2018).

Team

The foundation of SAFe is the empowered, self-organising and cross-functional teams (SAFe Team-level, 2018). The team consists of around 5-10 developers including SM, guiding the team and making sure that everything is working. The PO, is responsible for defining user stories* and prioritising the team backlog. Every two weeks, the team should define, build, test and deliver small increments of software, in so-called iteration. An iteration starts with a Program Increment (PI) planning meeting where the team decides what user stories, desired function based on business or customer value, they can deliver. The PI-planning must use face-to-face communication between the different teams to inform and align mission and vision. After the iteration, the team shows the new features through a demo to the PO to make sure they deliver what is wanted. Furthermore, at the end of the iteration, the team discuss what they can improve for the next iteration before starting the cycle again with a new PI-planning meeting. The backlog includes all tasks that need to be complete before the final delivery. Constant communication and effective collaboration are the key aspects of agile teams to working well.

Program

The program level consists of 5-15 agile teams including around 50-125 people called "Agile Release Train" (ART) (SAFe Program level, 2018). This structure is designed to deliver segment or entire solutions in iteration, usually of 10 weeks. Same as the team level, the period starts with a PI-planning meeting regarding goal and vision for all ART participate. After the meeting, each team discusses what they can do in order to meet the goal and reach the vision. If something is not delivered in the iteration, it could be delivered into the next. The Product Manager (PM) is responsible for the content in the backlog. The content could either be functional or architectural. Here, the SM is called Release Train Engineer (RTE) and are acting the same as the SM. The Product Manager, RTE and System functional architectural engineer have weekly meetings to make sure that all teams collaborate to reach the goal.

Large solution
The large solution level includes the requirements, artifacts and processes needed to be able to deliver large scaled solutions within several ARTs (SAFe Large solution, 2018). This level operates on a more complex level and handle solutions that not one single ART can handle. Due to the more complex environment, additional rolls and events are needed. Solution Train Engineer (STE) acting as a guide to the different ARTs and the Solution architect ensure that correct architecture is used.

Portfolio
The portfolio level is the most strategic level in SAFe (SAFe Portfolio level, 2018). Program Portfolio Managers allocate budget and investments for all the ARTs and coordinate them so that each ART deliver solutions at the same time. The strategy is organised centrally, while the execution is done decentralised. Furthermore, the portfolio level supplies agile portfolio and lean governance in order to enable solution.

Core values
The core values of an organisation must reflect and convey the corporate culture in order to fulfill the business goals (Kalenda, 2017). SAFe focusing on four core values, namely *alignment, built-in quality, transparency and program execution*. Alignment refers to the importance of a synchronised organisation to be able to communicate the mission. The built-in quality will secure that appropriate quality is maintained through the whole organisation and support capacity planning. Transparency refers to an environment that reflects openness and security. Leaders should take responsibility for their actions, both positive and negative, so the rest of the organisation can learn and improve from these. Lastly, program execution will support with tools and guidelines to achieve goals and it is important that the whole organisation understand the value.

2.2.4 The agile transformation

Today, many organisations have started to integrate the agile practices and principles into their organisation and thereby transforming from the traditional organisation. Johansson and Heide (2008) state that traditional organisations are always afraid of large organisational changes, but the changes are absolutely necessary and crucial in order to stay competitive on the market.

Challenges of the agile transformation

There are a lot of challenges when transforming an entire organisation. According to Boehm and Turner (2005), the challenges can be classified into three conflicts: development process, business process and people as shown in figure 2.5.

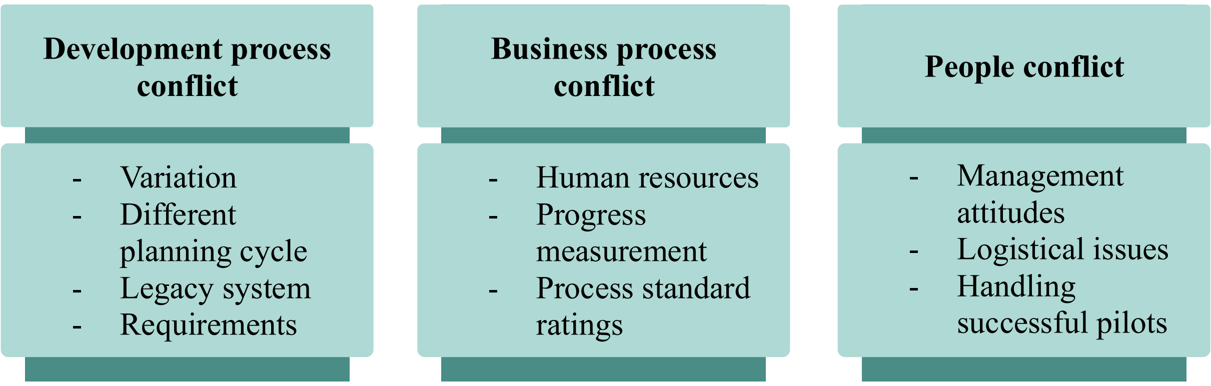


Figure 2.5: Boehm and Turner's conflicts within scaling agile transformation (2005).

Furthermore, Nerur et al. (2005) express that the adaption of the agile principles and practices will pose several challenges because the agile methodology is grounded in an opposing methodology compared to the traditional. The management, the structure and the culture of an organisation will be affected when an organisation that is used to work with a command-and-controlled management change to a leadership-and-collaboration management. The management must balance the right amount of cooperation with self-organised teams that are most suitable for the organisation. The authors (ibid.) claim that it is a huge challenge to implement the agile transformation successfully and that the entire organisation must make an effort. It will require much time and patience to build up a culture of respect and trust among the employees. In addition, Boehm and Turner (2003) argue that people's mindset and behaviors are not easy to change and an organisational transformation might therefore be slow. Tipnis (2018) has summarised the six most common challenges organisations are facing when implement the agile way of working into their organisations, see figure 2.6.

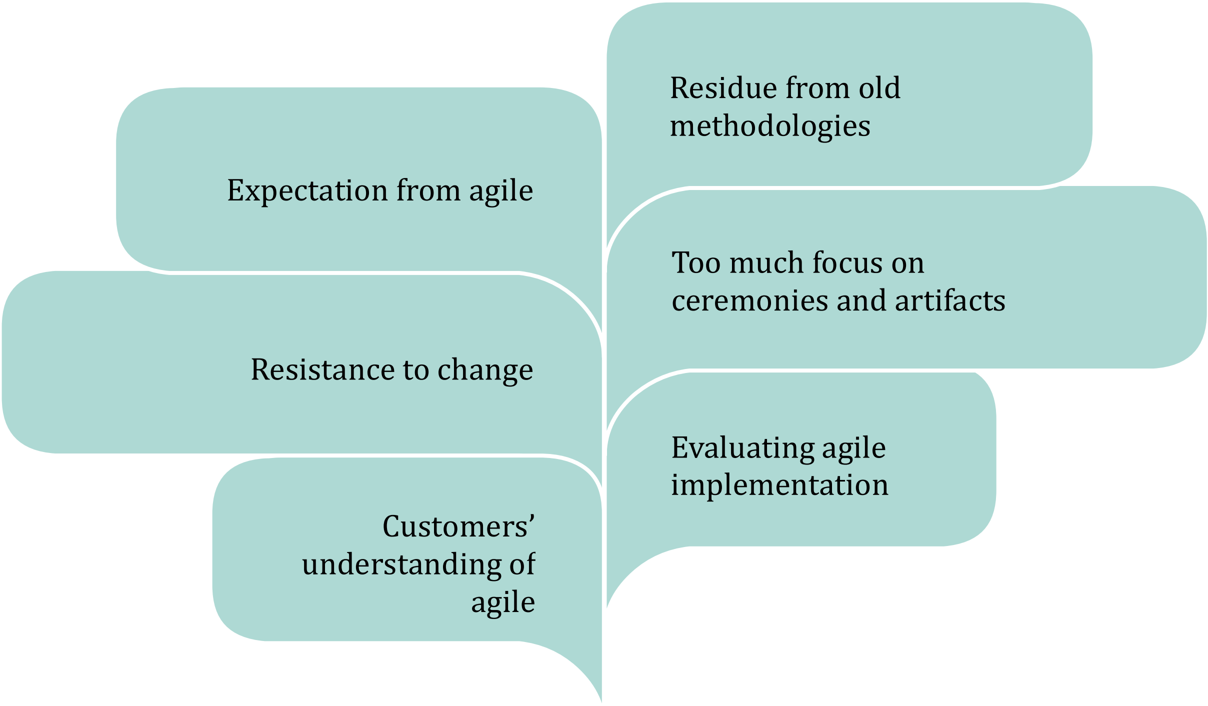

Residue from old methodologies
Expectation from agile
Too much focus on ceremonies and artifacts
Resistance to change
Evaluating agile implementation
Customers' understanding of agile

Figure 2.6: The challenges of the agile transformation (Tipnis, 2018).

1. *Expectations from the agile way of working*- One of the biggest challenges when implementing the agile way of working is to know what expectations to have of the new working method. It is of greatest importance to understand why the organisation has decided to implement it and also understand the fundamental of the methodology.
2. *Resistance to change*- It always a challenge to implement and establish new routines. To overcome this challenge, the employees must be trained in the agile way of working and gain a deep understanding of the agile philosophy. This to prevent resistance and fears from the employees.
3. *Residue from old methodologies*- Old working behaviors is hard to change. Even though employees of an organisation truly understand the importance of implement the agile way of working. The mindset of the old processes and the waterfall culture might still exist.
4. *Too much focus on ceremonies and artifacts*- A mistake from organisations that have implemented the agile way of working does, is to entirely focus on the chosen agile method. However, it is important to understand the new chosen method but the method is just a part of being agile. The organisation must understand the core values of the agile philosophy, that otherwise can be forgotten in the implementation process.
5. *Evaluating Agile implementation*- The evaluation of the agile implementation could be very difficult. In order to overcome this challenge, the organisation must define parameters to measure referring to the agile implementation. Often organisations tend to measure the numbers of agile teams and evaluate if all agile ceremonies are being followed. Although, it would be essential to evaluate the agile implementation in

terms of how it has helped the organisation achieve the initial goals of the implementation.

6. *Customers' understanding of agile-* The customers must have cognisance about the agile transformation. If the customers are not already working agile themselves, they must be informed about how the agile implementation affects them.

In addition to the challenges mentioned above, there are other challenges that have been identified. Boehm and Turner (2005) describe that it can be challenging when agile teams are working with traditional teams or with a customer that works in a traditional way. This due to the different project life cycles of the agile way of working compared to the traditional approach. The agile teams want to deliver functionalities on a regular basis. However, if the customer works according to the traditional approach, the agile teams might need to adjust their deliveries in order to synchronise with the customer. Another challenge that the authors (ibid.) present is that many organisation has distributed development. When implementing the agile principles and practices is can be very challenging if the teams are not in the same geographical location.

Many organisations begin with implementing an agile pilot project and can over time build up a strong agile leadership (De Smet et al., 2018). The yield from the agile implementation is depending on the effort of the organisation. However, when organisation wants to scale up the agility throughout the organisation, the leaders must be willing to embrace the precepts and enhance the capabilities in order to become successful in the implementation. Moreover, Nerur et al. (2005) describe it is important to find customers that are willing to collaborate and actively be involved in the development process. Knowledge sharing among partners will increase the understanding and the quality of the product.

2.2.5 Agile teams

According to Lind and Skärvad (2004), a team could be defined as "*when people work together*". Further, it implies that the involved people are dependent on or linked to each other and therefore in some sense work together. The authors also mean that a team is a group of people that work together and have the same goals. Kormanski (2005) argues that in order to survive as an organisation it is required to create effective teams that consist of the right skills.

Associated with the agile way of working, the concept of self-organised teams has been developed. According to Appelo (2011), organisations have changed from controlling people and giving them directions to focusing on effective learning and empowerment of the teams. The core activities should be on coaching and training the teams and distribute the decisions down to the team level. The teams must also be responsible for coordinate and design their own tasks. Additionally, the author express that it is of greatest importance that the team is responsible for the result, as a part of the empowerment. When individuals feel that they have authority, they will become more motivated related to their daily work. Although, it is equally important that managers are committed to tasks and keep promises in order to build trust.

Cockburn and Highsmith (2001) describe that it is a correlation between the maturity of the team and how agile the team is. A mature team could be related to high commitments to the work and the result, effective communication, and understanding of agility. The authors argue that the key factor of success when building a team is the ability to solve an issue and focus on the team spirit in order to increase satisfaction (Gren et al., 2017). Moreover, Castka et al. (2001) show a study of which factors that is considered to be important for the success of an agile team. The result of the study indicates that human factors, for instance, the group culture, the need of the individual and the existence of measures of performance affect the success of the implementation of agile teams. Cockburn and Highsmith (2001) also state that a successful implementation of any process is heavily dependent on the people and the people, in turn, is dependent on the level of support and coaching from the management and the customers. In other words, team performance is not only dependent on the competence within the team. It is also dependent on organisational context provided by management.

Distributed teams

Distributed software development and distributed teams have increased during the last years due to the globalisation of software development (Lee & Yong, 2010). The trend is to deliver software with high quality at higher speed with lower costs. In order to succeed, many organisations have chosen to use developers located across the world where costs are lower but at the same time where knowledge is available. In the article by Sutherland et al. (2008), the authors discuss three different distributed Scrum teams:

- *Isolated Scrums*- Teams are located in geographically different locations.
- *Distributed Scrum of Scrums*- Teams are located in geographically different locations and are integrated by a Scrum of Scrum that meets regularly.
- *Fully distributed Scrums*- Teams are cross-functional and located in geographically different locations.

Challenges with distributed teams

Hultgren and Lyhammar (2017a) believe that teams that are distributed over different geographic areas have more difficulties of delivering software systems with good quality compared to teams that are located at the same place. According to Lee and Yong (2010), they have found three main difficulties when working with distributed team that are visualised in figure 2.7, mainly: difficulties regarding communication, control and trust.

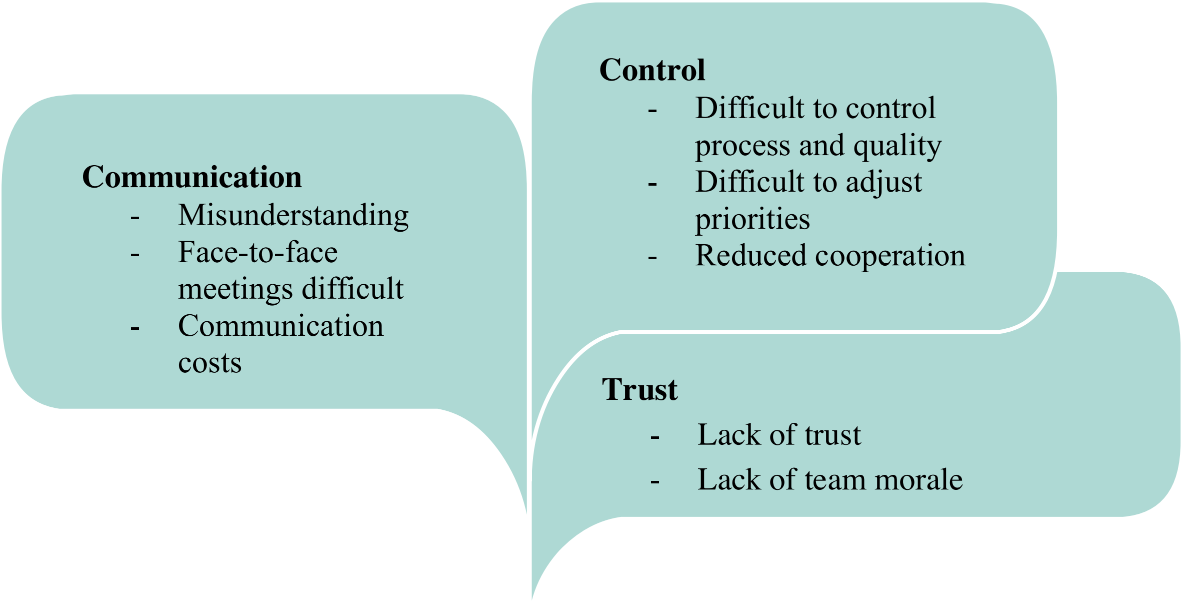


Figure 2.7: Lee and Yong's mapping of challenges of distributed teams (Lee & Yong, 2010).

The difficulties that arise regarding communication are related to misunderstanding that often occurring when all team members do not have access to face-to-face communication. The quality of the product and process is also exposed as control decreases when members are physically at different locations. Further, Lee and Yong (2010) argue that "*teams need to trust each other at the beginning of a project and use effective communication to help build a foundation of trust and respect of cultural differences*". Otherwise it will result in lack of team morale. The authors (ibid.) believe that all these difficulties associated with distributed teams will bring increased costs, which will ultimately be bigger than the savings that distributed teams bring.

Benefits with distributed teams

Hultgren and Lyhammar (2017a) mean that teams must physically be placed together in order to be more efficient. Nevertheless, they believe there are some benefits if a whole team is distributed somewhere else. For example, distributed teams may be required to achieve some skills that would otherwise never be available. Furthermore, they mean that distributed teams can create 24-hour support due to different time zones. For example, in Sweden it is common to work 8-17 but at the same time you want to provide support to customers day and night. If a software bug is discovered, organisations want to solve the defect as quickly as possible. Therefore, it is not uncommon that developers work evenings and weekends in order to solve it. Currently, they mean it is difficult to be able to provide 24-hour support due to the costs it entails. By finding a sustainable phase where teams do not need to work overtime, distributed teams may be a solution. The athours say that by finding developers in for example America, Asia and Sweden, support will be covered 20 hours a day. That will reduce the sprint length dramatically.

How to overcome challenges with distributed teams
Communication is the key to a successful distributed team (Hultgren and Lyhammar, 2017a). To overcome challenges with distributed teams, all team members must sit together as much as possible. Lee and Yong (2010) claim that teams must understand dependencies and split up user stories to facilitate communication. Further, in order to create better communication between distributed teams, Hultgren and Lyhammar (2017a) believe that live windows could improve the communication. By setting up a big screen with live camera and microphone at both teams' location, it wil create better communication between distributed teams, which also is a cheap and simple solution.

The authors argue that organisations must dare to travel a lot more at the start of the implementation of distributed teams and not focusing on cost savings. It will still give a payback at the end. Further, it is important to create distributed workshops to create common goals and visions, which results in a community between the teams. Hultgren and Lyhammar (2017a) have come up with a concept they call the "rolling chair". Where they have a chair at each site that is rotating. Thus, a developer from the native land will work with the team, preferably six months before the developer is replaced. It is a great way to get to know each other and to ensure quality delivery is guaranteed.

2.2.6 Agile Servant leadership

In traditional leadership, leaders are responsible for making decision, developing action plans and anchoring them to teams. The difficulty with traditional leadership is that leaders do not work close to what creates value and therefore they have less understanding of the end product. It may take time to anchor a decision taken from leaders at a higher level since teams do not understand or agree with the decision. The agile transformation requires a new kind of leadership (De Smet et al., 2018). Research confirms that organisational leadership are the biggest barriers and enablers of successful agile transformations. Therefore, agile leadership or the term servant leadership have been introduced.

Within the SAFe framework, servant leadership is described as "*a philosophy that implies a comprehensive view of the quality of people, work, and community spirit*" (SAFe Servant Leader, 2018). Thus, servant leadership means turning upside down on the decision pyramid and instead delegating the power to the team that creates value (Hultgren & Lyhammar, 2017b). The leader should rather create visions and eliminate obstacles. Further, a servant leader should listen, support and educate teams in problem identification and decision-making so that teams understand what decisions they can take by themselves or not. When the team themselves make a decision it does not require any anchoring. However, some decisions must be uniform for the entire organisation and therefore taken by the management. According to SAFe, managers should make the following transformation:

From	To
Deadlines	Goal
Coordinating	Coaching
Directing	Letting team organise
Knowing the answer	Asking the team
Fixing problems	Helping others fix them

Servant leadership enhances and balances organisations. Employees turnover rate decreases since they feel engaged and involved in the decision-making process. By letting the team be responsible for their work, blame game disappears on leaders. To succeed in the agile transformation, leaders must first change their own mindset and behavior before they transform the team and finally the entire organisation (De Smet et al., 2018). Hultgren & Lyhammar (2017b) mean that in order to analyse how a leader's ability to lead others, an employee survey could be done. For example, the team can rate from 1-10 how well a leader performs. Depending on the grade, the team should elaborate why they did not set a higher or lower number.

2.2.7 Agile metrics

A metric is a standard that is used for evaluating or measuring an organisation, which predict software resource requirements and improve software quality (Mukker, Singh & Mishra, 2014). There are many effects of using metrics in the traditional software engineering context (Kupiainen, Mäntylä & Itkonen, 2015). After the agile invention, a demand for new software functionality was required and therefore also new metrics. The traditional software metrics might also be applicable to agile software development. Top-performing software organisations are using metrics and see a lot of benefits with it, while lower performing organisations do not use metrics in the same matter. The benefits of using metrics is to:

- *Plan and estimate projects*
- *Make better decisions*
- *Understand quality and business objectives*
- *Improve software development communication, processes and tools*

Mukker et al. (2014) argue that "*quality metrics are indeed helpful in bringing to focus defect as they occur and prompt the need to comply with project requirements thus preventing avoidable rework at a later stage of the project*". According to a survey in the article, it turns out that the most common used measurements is related to quality, such as automated failure and software bugs.

However, there are different types of metrics: Business metrics, Base metrics, Quality metrics, Product metrics, Process metrics and Testing metrics (Mukker et al., 2014). Further, the different types of metrics can be divided into three main categories for metrics:

- *Informational*– Informs how well things are going
- *Diagnostic*– Identifies areas for improvement
- *Motivational*– Affect the behavior

The authors (ibid.) claim that optimal metrics should be: simple, objective, easily obtainable, valid and robust. Table 2.1 summaries some of the quality metrics that have been found in the studied articles:

Table 2.1: A selection of quality metrics (Mukker et al., 2014; Hultgren & Lyhammar, 2017c, Kupiainen et al., 2015).

Metrics	**Meaning**
Velocity	How much work the team has completed in each sprint
Reliability	Mean time between failure
Maintainability	The mean time it takes to fix a defect after it is found
Burndown chart	Show the work is progressing in each sprint. The total output including hours, velocity or backlog items, a team has completed over an iteration
Test coverage	How many percent automatic tests are available
Defects	How many defects are there on each release
Delivery precision	Committed vs. delivered
The team health	Team chooses what they want to measure and ratify each sprint to see if teams develop and see if they feel better

However, the authors claim that today's metrics are evaluating and measuring individuals rather than the overall project quality and progress (ibid). Hultgren and Lyhammar (2017c) are also skeptical to metrics and mean that metrics are often used incorrectly. It is important not to measure too much and that the metrics has a purpose. Furthermore, they mean that metrics should be replaced when they are no longer used. Therefore, it should not be standardised metrics for an entire organisation. If an organisation wants common metrics they should be chosen with very careful thought. However, organisation might consider of using disposable measurements now and then. For example, value stream mapping (VSM) is a good way to see the effectiveness of an organisation. An item is studied from start until it reaches the end customer to see what value it creates. The benefits of VMS are that bottlenecks are detected and can thus be remedied. The risks associated with metrics are that it often drives to bad behaviours since manager chase numbers to reach results. Even if the measurements improve, the number of the metrics can be misleading since it might be major technical debt behind them. Especially, risk arises if there are rewards linked to the metrics. Metrics can also create a sense of control to see how teams perform. Hultgren & Lyhammar (2017c) finds that senior managers often use metrics as a control measurement and do not inform the purpose of measurement to employees. There should be full transparency on what is being measured and that everyone knows what is being measured. Metrics on individual level should be avoided.

2.2.8 Software and hardware dependencies

There is a limited amount of literature and experience on how hardware development requires high demands on software development in agile practices and methods. Youn and Yi (2014) argue that "*hardware and software are interconnected and require each other, and neither can be realistically used without the other*". Therefore, they both must be developed simultaneously with each other in order to ensure safety, known as co-under development and can be visualised in figure 2.8 (Ronkainen & Abrahamsson, 2003).

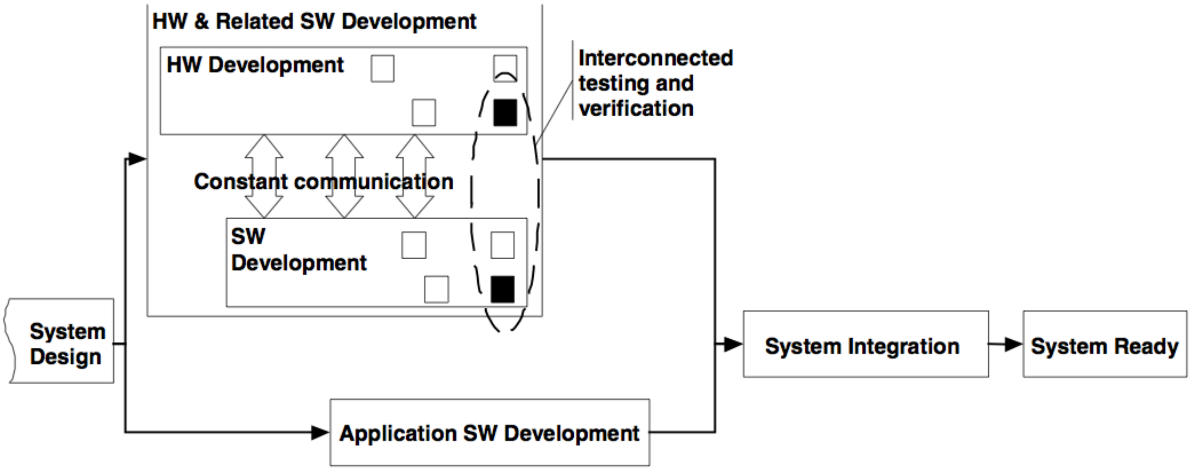


Figure 2.8: Co-under development timeline (Ronkainen & Abrahamsson, 2003).

Youn and Yi (2014) argue for the similarities and differences that exist between the systems. The dependency between the systems consists of that both having a complex development process that requires that a lot of requirements must be met. Further, for safety-critical products, both software and hardware need adapted certification to ensure safety and quality. However, the article indicates that the there are more differences between hardware and software development than similarities. The software is a logical system that consists of an iterative development phase where new functionalities are continuously added to the original code for a final product. Hardware, on the other hand, consists of physical components, which complicates the addition of new functionalities when the product is produced. The differences of small iteration deliveries are the main challenge that makes it hard to combine software and hardware development. Moreover, there is a big difference between software and hardware development relating to costs. Development related to hardware are often characterised by rapidly increasing costs at the end of a project while software projects, on the other hand, have a relatively stable cost change throughout the project. In addition, the cost of change and delays related to component change late in the project is much higher for hardware compared to software.

Dependence difficulties between hardware and software in the agile development

Cprime (2012) describes the difficulties that organisations experience when their product consists of both hardware and software. According to the author, it is important that organisations understand the dependence between hardware and software when implementing agile working methods. It is very common that organisations only introduce agile working methods that are suitable for software development for the entire organisation, such as Scrum, which is not always suitable for hardware development. The author (ibid.) argues that industry companies often have standards, regulations and legal requirements regarding hardware development. Those may contradict to agile way of working and complicates the implementation for Scrum as an agile method for software development. Further, there are difficulties to divide the hardware into the functional module that corresponds to agile software development. By breaking down the hardware functionality, DoD will be more easily defined and will enable better and faster deliveries. Moreover, software development needs thousand test cases before a delivery, while hardware requires less testing. The author claims, "*if the software must wait for the hardware to be created for final testing, this can create testing delays*". Hence, it is very important that testing in software development should be incorporated from day one in co-under development. The author suggests that simulations of the hardware should be developed in order to be able to test the software at a previous stage. However, it is not the same to perform simulation testing as real-life hardware testing.

Due to the challenges that arise regarding the dependencies between hardware and software development, different agile method should be used depending if it is software or hardware (ibid.). Scrum is a good approach when working with software development, but if working with hardware development the author suggests to work according to Commitment-Based Project Management (CBPM). In CBPM, each team member should deliver hardware features until a specific date. Compared to Agile software methodologies, CBPM does not need domain dependency and requires fewer iterations (Solera, 2016). However, both CBPM and Scrum are focusing on the team and can therefore improve communication between representatives of hardware and software (Cprime, 2012).

2.3 Quality

Kodak defines quality as "*those products and services that are perceived to meet or exceed the needs and expectations of the customer at a cost that represents outstanding value*" (Kerzner, 2004). To ensure quality, Kodak argue that focus must be on the following five principles:

- *Customer focus*- Quality is determined by internal and external customer
- *Management leadership*- All levels must work for quality improvement
- *Teamwork*- Everybody have to work together to ensure quality
- *Analytical approach*- Use analytical methods to control quality
- *Continuous improvement*- Continuously work to improve quality

A broad assumption of quality management is that process quality will affect the product quality (Sommerville, 2007). This correlation could be true if the quality is measure within manufacturing development were the quality of the product is related on processes, such as machine involvement. If the quality of the product is not sufficient, improvements on the process will influence the product. However, a process is not that easy to understand and monitor, and therefore the correlation between process and product quality is more complex. In software development, the quality of the process will not necessarily influence the product quality in the same way as in a manufacturing environment. Hence, the process will instead influence internal and external factors such as skills and market pressure, which will lead to fewer defects on software.

2.3.1 Quality in traditional development

The definition of Quality Management has been explained according to Bergman & Kelfsjö (2010) as a "*philosophy or an approach to management that can be characterised by its principles, practices and techniques*". Quality management was created from the manufacturing industry and was in beginning controlled by practitioners. However, in the 80s when the concept was improved and became more advanced, the concept became more important on a strategic and tactic planning level as well. According to Sommerville (2007), the quality of software has significantly improved due to the lately introduction of new technologies and techniques. The author claims that "good" quality management is not about standards and procedures that ensure that the standards are followed, more about creating a quality culture where all everyone feels responsible for quality. This is also confirmed by the core value of the agile practices and principles that strive for the concept of built-in quality and that each team is responsible for quality control.

The quality management is managed by a centralised control system and decentralised execution techniques (Sommerville, 2007). The control manager is responsible for ensuring that the quality is achieved at the very end. The planning, control and execution processes are closed linked together in a loop as visualised in figure 2.9. If one of the processes changes, the other two will be affected. To make sure safety, the control process operates as a two-way feedback system. Data correction can control and affect the execution process, while changes in management can influence the planning process.

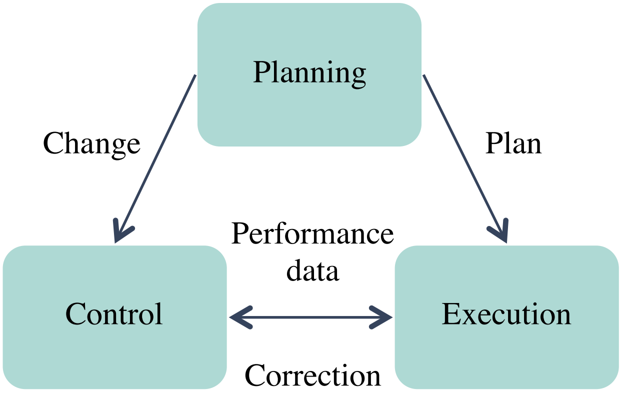


Figure 2.9: Traditional project management planning, execution and control system (O'Conchuir, 2011).

Using a quality management department is especially important for complex and large organisations compared to smaller organisations (Sommerville, 2007). Documentation could therefore be necessary to ensure that nothing is missing or incorrect performed. To ensure quality in each process is accomplished, especially for larger system, three different activities can be structured (O'Conchuir, 2011):

1. *Plan quality*- Appropriated standards and procedures on how to achieve quality must be determined and planned. The tools could differ depending project.
2. *Perform quality assurance*- Establish framework, standards and procedures that correspond to the project. Managers are responsible to ensure that tools and systems are correct due to quality.
3. *Perform quality control*- Managers must control that quality is consistent according to the plan. Include measuring, observing and documenting the outputs from the activities.

2.3.2 Quality in agile development

Organisations have pressure to change their processes and techniques due to quick market changes (Ambler, 2005). The agile working practice has been increasingly popular during the last years. Although, there have been difficulties to ensure that the quality is maintained when transforming and implementing the agile principles. Since the last years focus has been on the transformation, the quality assurance and control lag behind. Therefore, it has been challenging to find relevant literature regarding how to ensure that quality is maintained after the transformation and when scaling up the agile way of working through the organisation.

Ambler (2005) argues that quality should be seen as an inherent part of agile software development. It is of the greatest importance that the quality aspect is analysed and approved after each finished iteration. However, there are few types of research focusing on quality assurance in scaled agile software development (Qumer, 2007). The author means that there is a need to balance the focus on process and people by implementing a lightweight governance system in organisations with large-scale development. This in order to contribute to some disciplines and coordination mechanisms. Also, Ambler (2005) purpose that some disciplines are needed if an organisation want to scale up their agile development.

Built-in quality

Built-in quality is also one of SAFe's core values (SAFe Built-in Quality, 2018). The quality must be guaranteed at every increment to be able to respond to changes. It is everyone's responsibility to ensure that expected quality is reach during the process. The continuously ongoing quality improvement is very important in order to prevent extra or unnecessary work and high cost of change. The cost of change relates to rework and error will be much higher when it occurs late in a process, especially for hardware as shown in figure 2.10.

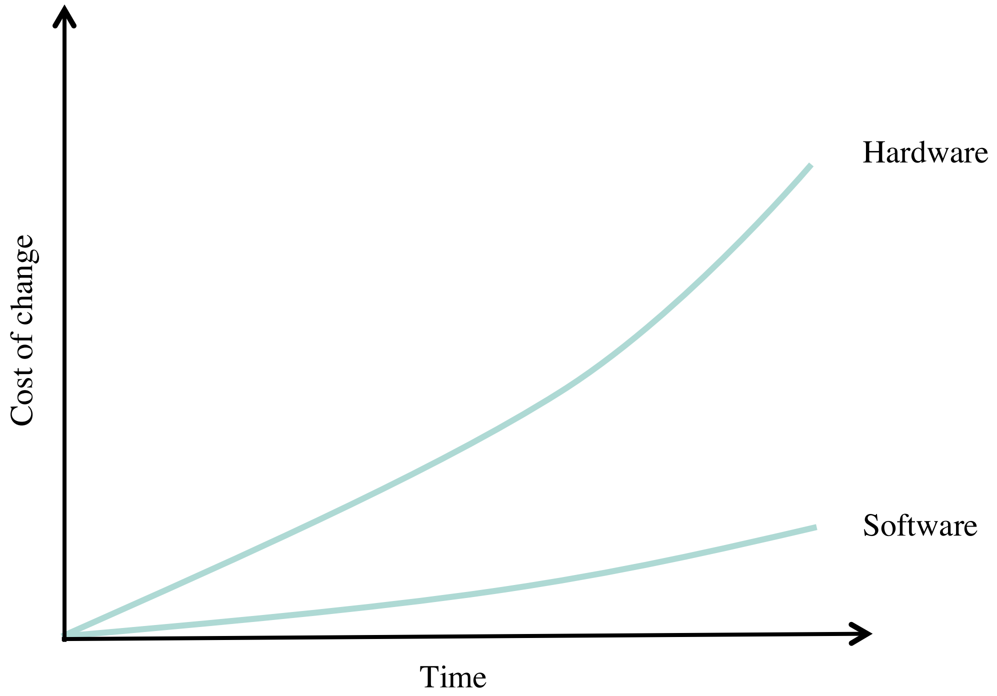


Figure 2.10: Cost of change over time for hardware, firmware and software (SAFe Built-in Quality, 2018).

To minimize the cost of change and secure the highest level of quality, SAFe proposes six different agile software development practices (SAFe Built-in Quality, 2018):

- *Continuous Integration (CI)-* The process of merging all activities from the different ARTs, testing, integrating, validating and developing them. Continuous integration helps to reduce risk of weak quality and improve faster delivery.
- *Test-First-* It is a type of just-in-time practice when first small tests are done before the final solution is created. Through this practice the quality is built from the beginning.
- *Refactoring-* Improve internal structure without changing the external behavior.
- *Pair work-* Various techniques of pair work, some could be used in combination.

- *Collective ownership-* All developers have a collective responsibility for the system and are allowed to change the code if it leads to improvements.
- *Agile Architecture-* The agile architecture principles and values support the design and architecture in a system and at the same time it helps to implementing new system capabilities.

2.3.3 Quality assurance and tools

Quality assurance includes both product quality and process quality (Ullah & Zaidi, 2009). Mnkandla & Dwolatzky (2006) have seen that within agile environments, there are less standardised quality assurances and controls compared to the traditional project management. When working agile with small incremental deliveries, the structure of quality assurance needs to change. Ullah and Zaidi (2009) mean that when working according to the agile principles, the developers within the teams should perform most of the quality assurance activities. The best quality assurance practices are those that ensure quality automatically within the processes. Quality assurance should be developed around testing and feedback. The authors also argue that testing is an essential part concerning quality assurance activities. Furthermore, there are numbers of different quality tools to use as quality assurance. However, the traditional quality tools are hard to implement in the agile way of working (Ahmed, 2018). However, the author expresses that at least organisations should evaluate the possibilities to integrate the traditional quality tools with the agile working practices. By maintaining the design and structure of the traditional quality tools the team can ensure that potential failure points are being analysed in time.

Nabulsi (2015) argues that quality assurance should be a part of the standards. There are many different standards that organisations can or have to follow. Many organisations are implementing these standards to improve quality and to certify the organisation towards the customers. When it comes to the standardised quality management system in the software industry, one of the most common standards is the ISO 9001. This standard represents organisations' quality system and requires that organisations have documentation over all the work processes (Tevell & Åhsberg, 2011). In addition, the ISO 9001 standard is used as a framework for audits in order to certify that organisations follow the working processes in a certain way. It is the organisation that is responsible for developing the routines and writes these down. To be certified, it requires quite extensive documentation. Gallina and Nyberg (2015) among other athours discuss the actual effect of the standards. They argue that the agile oriented working approach can hardly co-exist with many of the comprehensive documenation required standards. In addition, a huge amount of energy is spent to developing and following the rotines and standards rather than achieving a high level of customer satisfaction.

Agile Testing

According to Kronfält (2011), tests are an essential part to secure that the software has the right functionality and works in the intended way. Otherwise, it will result in several problems in terms of cost and time losses. Testing in agile development is a continuous process where the code is developed and tested in small increments. It is recommended to first develop the test before writing the actual code in order to improve delivery by focusing on the intended results (SAFe Test First, 2018). By doing so the tests will help elaborate and define the intended behavior of the system in a better way even before the actual system is coded. The tests provide a definitive statement of what the system does, instead of statements of what the system is supposed to do. In the agile development, the tests should be automated wherever it is possible. In order to succeed with faster continuous software delivery, it is vital to use test automatisation (Jansing et al., 2015). The author argues that test automatisation is suitable for standardise products and repetitive tasks. In table 2.2, the advantages and disadvantages with automated testing are presented.

Table 2.2: Advantages and disadvantages with test automatisation (Jansing et al., 2015).

Advantages	Disadvantages
Faster deliveries- Run tests faster than human developers.	**Knowledge**- Developers do need knowledge of the product and how to write the tests.
Reliable- eliminating defects due to human error.	**Cost of maintenance**- It could be high costs if the tests need to be updated.
Repeatable- Explore how the software responds to repeated performance.	**Difficult to repeat**- No new defects are found when repeating.
Reusable- Tests can be reduce if it is created.	
Better Quality- Run more tests in less time.	
Economical- Reducing costs due to fewer resources needed.	
Independent- Each test can be run by itself.	

As the quality is built-in from the beginning, the tests can be seen as quality control. This due to the required specifications of the system behavior is compared to the actual test cases (Kronfält, 2011). Tests give the opportunity to measure the quality of the software. According to the SAFe framework (2018), agile teams cannot go further in the development process without high quality and they cannot achieve high quality if they do not have continuous tests. However, the biggest challenge due to agile testing is that the test is exponential in relation to every development cycle. This since old functionalities must be tested in combination with the new once.

A/B testing

A/B testing is a method used primarily for comparing systems to each other in order to collect a statistical basis of actual users to determine which one performs better (Sauro & Lewis, 2016). Usually, there are two variants: A and B that are compared to each other, hence the name, but it could include more variants. In A/B testing, one version of a system is modified and compare to the original one (Optimizely, 2018). Often, half of the users are exposed to the original version, while the other half is exposed to the changed version. Customers are randomly exposed by one of the variants of a system and after the test period, the variant that has been most successful are implemented.

In recent years, A/B testing has become a common method for evaluating data collected by customers (Sauro & Lewis, 2016). A/B testing as a method has a great advantage of analysing a huge amount of customer data over many other methods. The method saves time compared to, for examples, interviews and surveys since the method does not have to analyse data about the responders' opinion or feelings. The benefits of using A/B testing are that customer does not even know that they are a part of a survey and therefore, will not consider their actions. Organisations can immediately change the functionality since one of the variants can be proved better than the other. Further, developer's argumentation of using new features or change current one can be strength by A/B testing.

3. Method

This chapter present how the master thesis was conducted, including research strategy, research process, data collection and analysis of data. Moreover, a discussion regarding the quality and ethics of the research is presented and motivated in this chapter.

3.1 Research strategy

There are two main strategies to select from when conducting a research strategy, qualitative and quantitative (Bryman & Bell, 2011). A qualitative approach emphasis mainly on interviews and literature reviews, whereas a quantitative focusing more on samples and mathematical formulas. The authors state that the advantage of a qualitative research strategy is that it results in a huge amount of data, which in turn allows the researcher to adapt the data collection based on the findings. Björklund and Paulsson (2014) also argue that a qualitative research is most suitable if the researcher wants to create deeper understanding regarding the complexity of an issue. Further, the advantage with a quantitative research strategy is that this strategy enables to create theory and patterns, which make it possible to compare the retrieved result with other results. By using a quantitative approach, it will allow a faster interpretation of data and statistical analysis can be used.

The purpose of the research determines whether the study will be of a quantitative or qualitative nature (ibid). In order to answer the research questions for this master thesis, a qualitative research was adopted since the collection of data was carried through a literature review and interviews. Additionally, Bryman and Bell (2011) discuss that qualitative research strategy enables a more flexible structure. Something that is was useful for this report since the problem scope was unknown at the start of the process.

3.2 Research process

The research process of the study is described in figure 3.1. The process of the research began with the collection of data in terms of both literature and secondary data to gain an understanding of quality and the agile way of working. In addition, interview questions were created based on the theoretical framework. Data was continuously analysed since more understanding and knowledge about the subject was obtained from the interviews.

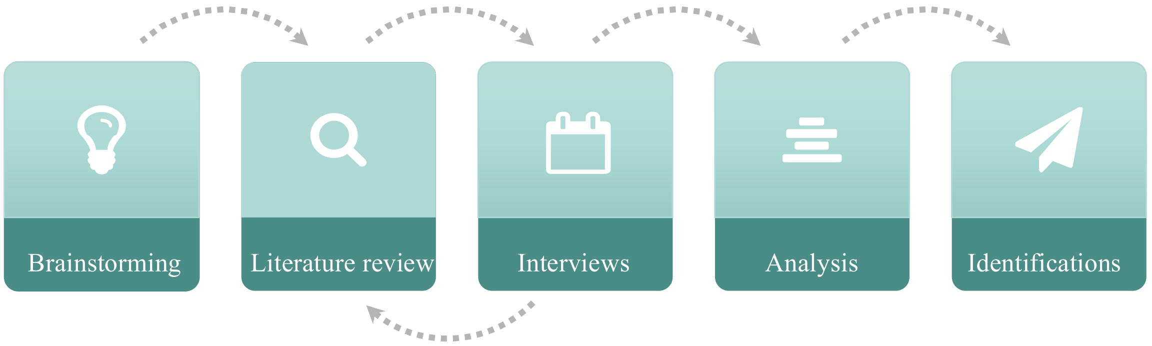


Figure 3.1: The research process of the master thesis.

All the activities within the research processes will be further described in more details in the next sections.

3.3 Data Collection

To be able to answer the research questions and the purpose of the master thesis different data collection methods are required. In the following section, the data collection methods that were used are described.

3.3.1 Literature review

The literature review focuses on quality when working agile, in more specific, how different organisations work with quality when they are working scale agile. The literature review is the foundation for the interview questions and for the analysis of the master thesis.

The primary focus of the literature review was to create an overall understanding before collecting empirical data. The data was collected through books, articles, journals and internet pages. Further, the data was mainly found via Chalmers University library databases and Google Scholar, following identified keywords were used; *Agile, Scaled Agile Development, Agile Practices, Agile Software Development, Quality, Quality Assurance, Software Quality*. It is important that different data from different data sources are used in order to ensure reliability (Grundström, 2013). By mainly use a few sources when conducting a literature review there is a risk of losing the overall perspective. Therefore, a deep literature search was done in order to get an objective picture. When searching for literature within the area of agile development and scale agile, the literature that were found constituted a comprehensive base of literature. The same applies for the research field of quality and quality assurance, were an extensive base of literature could be found. However, there were significantly less literature regarding quality and quality assurance related to scale agile development.

3.3.2 Secondary data

Secondary data is data that is collected at an earlier stage and has another purpose than assisting data for the study (Hox & Boeije, 2005). Most of the secondary data is official, such as official statistic and administrative records from organisations. Secondary data in the form of official statistical and improvement work related documents were provided by organisations via interviews and mail correspondence. It is important that data is collected from different sources to ensure the reliability of the study. However, the collected data must be relevant and useful and the quality of the data must be evaluated.

3.3.3 Interview

Interviews are the most common form of qualitative methods and collect the interviewee's point of view (Kvale & Brinkman, 2009). Interviews gather information that is otherwise difficult to find through other research methods, such as observations and surveys. There are a variety of different qualitative interview methods. Gillham (2008) distinguishes between unstructured, semi-structured and structured interviews. For an unstructured interview, the interviewee has the responsibility to lead the interview while questionnaires are used in structured interviews. A semi-structured interview is characterised by simple and open questions (Flick, 2018). For this this research the semi-structured interview has been chosen as the qualitative interview method, with the same questions for all interviewees. Depending on the answers, supplementary questions were asked, which generated open answers with rich content. The semi-structured interviews were conducted in different organisations to investigate how the different organisations work with quality when working scaled agile.

The research involves 17 semi-structured open-ended interviews, meaning that the questions gave the responder opportunity to lead the interview. The questions were designed according to Flick (2018) and all the interview questions could be seen in Appendix I. The interviews were conducted anonymously with interviewees from various backgrounds, such as different work experiences and different agile experiences. However, common for all the interviewees were that they all are working with software development and have more or less the same position within the organisations. With foundation from the literature review, the first draft of the interview questions was made. In collaboration with the supervisor and employees from Knowit, the questions were validated. From the feedback given, the questions were finalised and improved. For further feedback, the questions were used and tested internally with an agile coach at Knowit as a first pilot interview. No changes were made to the interview questions after the pilot interview. Additionally, the pilot interview was not included in the final report, instead the interview was used as an opportunity to practice before going out on the field. The interview questions were divided into three main parts: the agile transformation journey, how the organisations work with quality and the best agile practices related to quality work. The length of the interviews varied between 60-90 minutes, were conducted in Swedish and translated into English afterwards.

3.3.4 Samplings

The main goal with the interviews was to identify how organisations are working with quality when working scaled agile. As mentioned before, there is a limited availability in previous research regarding this area. Therefore, it was important to find interviewees from different organisations and with a rich knowledge regarding both the agile way of working, quality and software development. An industry research was developed early in the research process to identify interesting organisations from different industries to interview. The supervisor from Knowit assisted a list of different potential interviewees to contact. In addition, the researchers used the snowball sampling technique (Bryman & Bell, 2011). This technique implicates that the interviewees were asked at the end of the interviews to give names of other people they thought would be interested to participate. The recommended people could be both from the same organisation as well as from other organisations. Although, a risk that could be mentioned with the snowball sampling technique, is that the interviewees might recommend other people based on their own interests.

There were 17 interviews in total that were conducted throughout the research process. As can be seen in table 3.1, the interviews included different interviewees with different roles. As the table shows, many of the interviewees have the same role and responsibility within the organisations. Furthermore, most of the interviewees have a management positions within the organisations.

Table 3.1: The case studies interviewees.

Role	Number of interviews
Product owner	1
Release Train Engineer	2
Agile transformation leader	2
Software Lead	1
Quality Management	9
Senior safety architect	1
Professor software engineering	1

The majority of the interviewees are representatives from the automotive industry. Thus, there were also interviewees from the security and surveillance, telecom and information management industry as well. Additionally, one interview was conducted with one professor in software engineering from Chalmers University of Technology in order to gain more knowledge within the area.

3.4 Analysis of data

The majority of the interviewees are representatives from the automotive industry. Thus, there were also interviewees from the security and surveillance, telecom and information management industry as well. Additionally, one interview was conducted with one professor in software engineering from Chalmers University of Technology in order to gain more knowledge within the area.

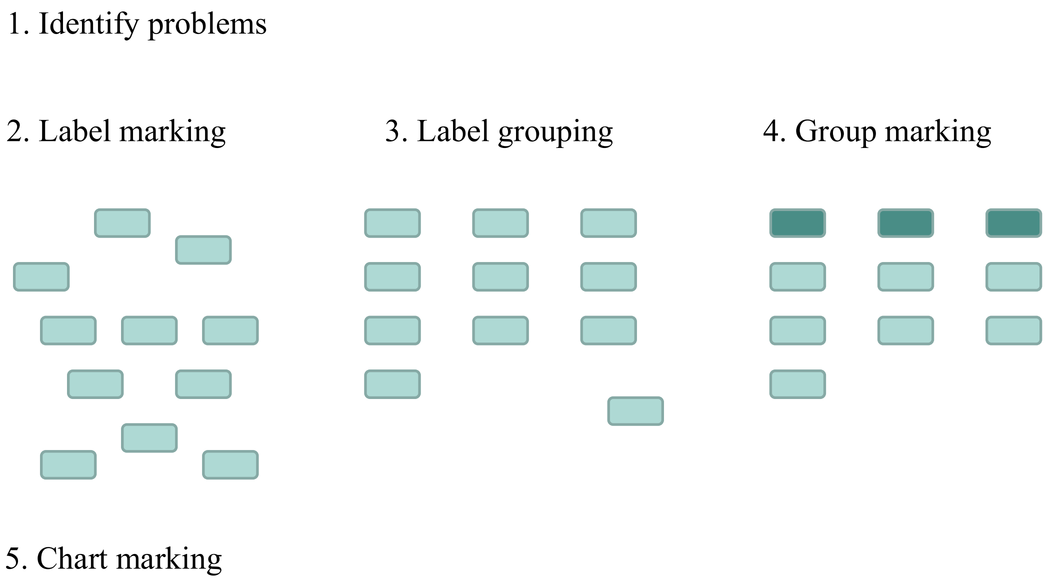


Figure 3.2: Analysis of data according to the KJ-Shiba method (Shiba et al., 1993).

According to the KJ-Shiba method the 5 steps below must be followed:

1. *Identify problems*- First the problem must be identified before doing the next steps.
2. *Label marking*- Thoughts and fact from the interviews was written down on post-it-labels. One issue per label was written, and the post-it labels were put on a wall and duplicated was removed.
3. *Label grouping*- The post-it labels were categorised and the areas that considered belonging to each other were grouped together.
4. *Group naming*- Each group was given a headline.
5. *Chart marking*- Causes, interdependence and connections were analysed between the different groups.

The results from the empirical findings can either be analysed vertical or horisontal (Bohlin, 2018). Vertical analysis means that each interviewed organisation is analysed separately. Horisontal, on the other hand, means that a comparative analysis between organisations is made. Horisontal analysis sometimes requires more information and details, which can be difficult to collect from each organisation. Since the report is anonymous, horisontal analysis of data was chosen. The output from the 17 interviews are divided and compared into different areas.

3.5 Quality of research

In qualitative research, it is of greatest importance to consider the validity, reliability and objectivity of the data with the purpose of establishing and assessing the quality of the research (Bryman and Bell, 2011). Validity, reliability and objectively can be seen as the measurement of the credibility of a study (Björklund & Paulsson, 2014). In brief, the three concepts of credibility could be described as follow:

- *Validity*- The extent to which you truly measure what you intend to measure, in essence absence of methodological or systematic errors. By using different perspectives, the validity increases.
- *Reliability*- The degree of operational reliability of the measuring instruments, basically the extent to which you get the same values if you repeat the investigation. The reliability could be increased by using control questions in surveys and interviews in which the aspects are investigated yet again.
- *Objectively*- The extent to which values affect the study. This be increased by clarify and motivate the different choices made in the study.

When conducting a qualitative research with interviews, there is a risk to miss important facts. This because of the data collection is based on the interviewees' knowledge and perceptions. In order to avoid bias data from an interviewee, the triangulation method was used (Bryman & Bell, 2011). The method implicates that different sources of data are used to prove the same point of view during the research. This was accomplished by discussing the same topic in all the interviews, which entailed that the researchers could identify common denominator among the organisations. The authors (ibid.) state that triangulation increase the reliability of the research.

3.5.1 Supervision

To ensure academic quality and honesty, Erik Bohlin, Professor at the division of Science, Technology and Society at Chalmers University of Technology supervised the work. The supervisor has the background and knowledge required for the chosen research field and was scientifically and quality responsible for the degree project. In addition, Teresa Thorsson CEO Project & Quality Management was the supervisor from Knowit. Teresa and other colleagues from Knowit provided information about the research subject and provided contact inorfmation about potential interviewees to the research. To ensure that language and formalities have the academic standards required by a degree project on Master level, Chalmers Student Writing Center checked the readability of the report and improved grammar.

3.6 Ethics

Bryman and Bell (2011) argue that it is important to consider the ethical aspects of a research. Further, the authors describe four different areas to take into consideration: harm to participants, lack of informed consent, invasion of privacy and deception. To avoid ethical offense during the interviews, the interviewees were given an explanation of the purpose with the interview at the start of the interview. Also, the interviewees were informed how the interview data would be used and that the interviewees would be anonymous in the report. The purpose with the collected data was to create a general view of how organisations are working with quality. Therefore, no data was sent afterward to the interviewees since this was considered to not be necessary.

4. Empirical findings

This chapter will present the empirical findings that are based on the semi-structured interviews with different organisations that are working scaled agile. Further, how the interviewees experience the agile way of work and how they ensure quality will be described as well as the difficulties and best practices associated with it.

4.1 The agile transformation, frameworks and methods

Many of the interviewed organisations are at the beginning of their agile journey and have implemented agile principles for about one to two years back. The interviewed organisations that have worked according to the agile principles for a long period started their journey in the early 21th century. However, there are not many of the interviewed organisations that have worked scaled agile for a longer time period. Some organisations argue that their agile journey started at the team level because they saw a lot of software defects and wanted to change their way of working to make the software more efficient. Further, organisation felt pressure from employees at the bottom of the organisation. It is very common that the employees start to implement the agile working practices since they have had previous experiences from working agile and understand what benefits that could be gained from it. Others say that they wanted to change to a more team-based approach that the agile principles enable. Most of the interviewees pointed out that they started implementing the agile way of working in small pilot projects in different teams and then spread it to the remaining teams and lastly implemented it on a scaled level. However, one organisation did not use a pilot project and instead implemented the agile way of working directly across the entire organisation. This interviewee argues that their agile transformation was successful implemented due to radical transformation through the whole organisation. However, it has still been a long journey for this organisation to become so successful within the agile way of working that they are today, over more than 10 years and yet, they do not consider themselves completely finished.

All organisations that were interviewed used some form of scrum methodology, alternatively working on the basis of lean-agile principles. In general, all interviewees mean that they have modified the chosen agile working method based on the organisation's conditions either by themselves or in collaboration with an external partner. All interviewees are following a fully developed, created their own or have modified an existing scaled agile framework. Many of the organisations work according to or have chosen to modify the SAFe framework. Some of the organisations have added certain roles within the SAFe framework because they felt that they were missing. For example, an organisation has chosen to add a project management function and purchasing function to the existing framework.

4.2 The structure of the organisations

From the interviews different structures regarding the quality department could be found. Most of the interviewed organisations have a combination of more than just one structure. The different structures are described in the following section.

A quality department

The majority of the organisations have the same division of labour as before the agile transformation. The empirical findings show that a quality department still exists for several organisations even if they approach the quality department more as a supportive function. One organisation that has worked agile from the beginning of the 2100 century expresses that they have gone back and forth regarding if they should have a quality department or not. They still have a quality department, but they do not control and check the teams' work as they did before. Today, the quality department is more focusing on the processes, for example, searching for routine errors, secure requirements and that the final verifications are done. Another example from the interviews is one organisation that has a software quality engineer group. This group is working with quality within projects and supports the teams to achieve the process goals. The group is working with around one to three different projects at the same time, depending which phases the projects are in. They are operating as internal coaches and have the responsibility to see that all work is being performed. However, they are not participating in every sprint planning session.

One interviewee means that even if the vision is to excise the quality department, it is not possible. This was also discussed with other interviewees and they are united that the quality department has to remain since they cannot expect that all team members have the knowledge and competencies that is required. Some organisations also mean that there is a high turnover rate of employees that result in new inexperienced people, which strengthen their arguments to keep a quality department. Moreover, most of the interviewees agree that the quality organisation should be more as an auditor department rather than a quality department. One of the interviewed organisations explained that they have a supportive assessment function within their organisation. This support group helps to ensure good quality for all the internal activities but the aim for the future is to have external assessments in order to support the quality.

One interviewee also explicates that they are striving towards a more supportive quality department. Which implies, for this organisation, that the quality department should have the purpose to help and supporting the team members to ensure quality. They mean it would be ideal with one-way communication, were the team contacts the supportive department for help. Nevertheless, today there is a two-way communication instead of one way. They are trying to implement one-way communication, but the issue of today is that the team maturity varies a lot. Also, the quality department has the main responsibility of quality today, which makes the change to a supportive department more difficult.

No quality department

The organisations that have been working agile for a longer period or organisations that have been agile from the beginning have no quality department. The responsibility of quality is distributed to the teams, everything that concerns quality from the start to the end of a project. The interviewees believe that it is very important that teams feel that they are responsible and that it is each and everyone's responsibility to strive for built-in-quality. Most of the organisations that do not have a quality department argue that even if teams are responsible for quality, the main responsible actor to ensure that quality is fulfilled is the PO. An interviewee describe that it is the same team members yesterday as there are today but titles such as “project members” and “project leaders” do no longer exist. The agile way of working has made it clearer that it is the product qwner that “owns” the quality.

Proactive quality department

Most of the organisations argue that the agile principles and frameworks do not describe how to work with proactive quality. However, few of the interviewed organisations mentioned that they were using a proactive quality department to ensure preventive quality. The organisations that were telling that they have a proactive quality department argue that there are many benefits of having one. These departments work extensively with customer data to prevent errors before they even arise. This is described in more detail in section 4.6.5.

Monitored quality department

Based on the interviews, it appears that there was a great variation on how the organisations worked with monitored quality. Some of them have an independent quality department that are performing internal audits and assessments, while others are trying to implement one. One organisation that does have an independent quality department use software quality assurance (SQA) to do assessments on teams. How often audits and assessments are being performed depends on each project. Those organisations that have implemented internal audits are looking at risks for each project scope. After the risk analysis is made, they make pre-notified audits to see how the teams actually work. After each audit, they provide feedback and write a report of corrective action. However, all interviewed organisations do not work as comprehensive with audits and assessments and instead execute random sample audits and assessments and they are relying upon that each team ensures built-in-quality.

External quality department

During the interviews it found that that some of the organisations are outsourcing a part of the quality responsibility to an external partner. Mainly duties to outsource are, for example, the final software testing and customer surveys after a project is completed. Many of the organisations are also using an external service, which ranks the product based on parameters such as customer satisfaction and product quality. Organisations argue that customer surveys are an important step in evaluation and continuous improvement work.

4.3 Leadership

Based on the interviews it turns out that management have different view and understanding of quality. Some believe that it is the management that created the implementation in order to achieve better quality, while others mean that management does not understand the meaning of quality when working agile. Some organisations argue it is difficult to get support from management since they do not understand what technology the team needs in order to improve quality. For example, most of the interviewees claim that test automatisation is needed in order to ensure quality, but at the moment they perform too many manual testing. Some interviewees mean that it may be due to the fact that the management does not understand the importance of test automatisation.

From the interviews it could be stated that managers have taken a new role due to the agile transformation and instead of delegating, they try to coach and support the teams. However, managers find it difficult to know how to act because they do not know what they are allowed and not allowed to say in the agile way of working. Also, managers claim that there is a fear of change because their specialist role will disappear due to the agile transformation and they believe it is easy to fall back on the old way of working. During the interviews, the problems of having distributed teams were also discussed since the leadership can differ between countries. For example, one organisation that have distributed teams in another country experiences that their developer manager does not want to change the way of working and therefore do to prioritise it.

4.4 Team

During the interviews, a large part of the discussions was related to the agile teams. In the following section, the empirical findings regarding the role and responsibilities of the team are presented. The difficulties many of the agile teams within the interview organisations are facing will also be discussed as well as the result the increasing benefits of working in teams.

Roles and responsibilities

Depending on organisation, the number of teams and the team size varied a lot. However, all the interviewed organisations have a maximum of ten people concerning team size. For those organisations that follow SAFe the number of trains varies a lot but the number of teams within one train is around five to six teams.

The findings show that most of the organisations have one PO and one SM per team. The PO is responsible for priorities the activities that should be done and act as the voice of the customer in the organisation. Further, the SM is responsible for optimising the teamwork and eliminating obstacle. Many of the interviewees express that the idea is to have one PO and one SM per team. However, some of the interviewed organisations do not have a SM at the moment and the SM's responsibilities are instead distributed over the whole team. When it comes to the role of the teams and their responsibilities, the overall expression is that the power is now distributed to the teams. The teams should be self-organised and define their own work and solve the problems that might occur. Although different interviewees have described it differently, one object states that the teams are responsible for the “how” when it comes to quality. In more specific, the “how” includes to develop the software and test cases, write code, test the code and break down the quality requirements. Another describes that the teams own the backlog and are responsible to prioritise the backlog, while others have a PO that is responsible for doing that.

When it comes to the quality perspective at the team level, most of the interviewees mentioned that the teams are following DoD in order to secure the quality. One interviewee described that as a team member, you should always ask yourself the questions ”Am I done, should I really be done, have I done the unit tests and are the software reviewed?”. Another interviewee argues that the teams’ responsibility should be to follow their DoD and they are done before the code, unit tests and design documents are updated. Some organisations have also included in their DoD that code must be reviewed, either form another team member within the same team or from another team. Additionally, on interviewee describe that they have defined all the working processes and it is up to each team member to feel responsible for doing all that is required and not taking any shortcuts. The same interviewee means that the goal is that no one should take shortcuts and skipping process steps, but this is a question about maturity of the agile teams.

Further, the interviewees argue that they want the self-organising teams to have a quality perspective in everything they do. One organisation has divided the quality work between the teams and the PO, whereas the teams are responsible for the tasks and stories and the PO is responsible for the features. Thus, another interviewee means that the responsibility for the teams is no different than before and the team members should know what to do.

Several of the interviewees argue that they have changed from a push to pull working system. Organisations are striving for that the team members should pick tasks from the backlog by themselves and not the other way around as it was before when a project manager pushed out tasks to the team. Nowadays, it is also the teams' responsibility to control that they are in line with the plans. Most of the interviewees argue that the most important is to make sure that the teams feel ownership over the deliveries. Further, many organisations express that the teams shall take responsibility and be strongly committed to their work. In addition, one organisation do not only work for that the teams should be self-organised but also should be responsible for the improvement work on a cross-functional level. The managers are only there to support and communicate the goals. Then, the teams are responsible for doing the changes and improvements needed in order to achieve the goals.

Difficulties for the agile teams

The empirical findings show that many of the interview organisations are facing difficulties with the agile teams. The majority of the interviewee arguing the problems with immature teams and that the teams need more coaching and educational support. Many interviewees mean that today the teams are lacking general understanding of what they can decide by themself, which are their mandate, what is included in their role and do they understand what is included in DoD? Also, it is difficult to achieve a common understanding since different individual are experience things differently. For example, one interviewee argues that own responsibility is defined different for different individuals. Another interviewee means that it is hard to implement the ”agile mindset” in the practice. Even though, agile coaches are available with the purpose to make the team feel safe and eliminate obstacles, the agile coaches are struggling to clearly express what is expetect in detail for the team members. Moreover, an interviewee believes that changing mindset is difficult due to that employees are used to work in a traditional way. In the traditional project approach the focus has been on functionality while now it is no point to add functionality if the foundation is not working. For this organisation, the difficulties are to change the team members to handle and solve problems direct. The interviewee argues that it is much easier to fix the problems that are made yesterday compared to something that was done for three months ago. The team members are not used to receive the feedback so fast, but they should always start with the test reports so that they can ensure that the code follows the quality requirements. The interviewee describes that they have worked a lot with this challenge and for those teams that started to work agile around 2010, are now handling defect reports imedeiatly when they receive those. However, the interviewee stresses out that it always easy to fall back to the old working practices.

Another difficulty that arises during some of the interviews is for those organisations that have distributed agile teams. One organisation has testers and developers in the same team, but the developers are located in another European country. This results in that they cannot get the daily exchange and have a very different perception of how to work agile. The interviewee means that the ambition is to have cross-functional teams but there is a big challenge to create those cross-functional teams. Another organisation has also distributed agile teams where both the developers and testers are located in another country and only the teams’ leaders are located in Sweden. This interviewee states that it would have been much easier if all the team members should have been located in the same place. Common for both organisations with distributed agile teams are that the price and the competencies control the team members location. Furthermore, one organisation had co-located geographic teams and have different team set-up in different geographical areas. The interviewee argue that this is due to ensure the right competencies and resources, as well it is a competitive advantage to have several sites where the largest customers are positioned. When capacity is needed, for example, for a bigger project the teams need to collaborate over the borders from different geographical sites. The interviewee describes that they are struggling with communication difficulties, both regarding linguistic barriers and technical.

One interviewee describes that some teams deliveries better value than other teams and questions why it is so. The same interviewee also describes the problem with a high employee turnover rate. There are good times for developers and if you have high competencies it is most likely that you are being transferred to another department that needs help to start up the agile implementation. Additionally, another interviewee expresses the difficulties to transfer into agile teams when you have over 1000 employees, how should the competencies be spread out through the teams? A third interviewee also mentioned that it difficult to create good performing teams that have the right competencies. Further, even if team members have the right competencies, it is still difficult to create well performing teams. The interviewees describe that there are negative sides as well when transforming the organisation into a team-based structure. For example, the individuals with key roles and specialist competencies are moving towards to become one part of the overall organisation. Many of those do not want to let go of its previous roles, which result in that the teams are not one hundred percent t-shaped*.

Benefits of agile teams

It is not only difficulties that have arisen regarding how teams work with quality after the agile transformation. Many of the interviewees mean that it has become much clearer how the teams should work to ensure quality. They are responsible for achieving goals, prioritising the backlog, performing improvement work and are responsible for quality work. They have simply created more self-organising teams. Since the teams working with end-to-end quality, they have become more committed to achieving good quality. Small deliveries provide quick feedback that allows defects to be repaired instantly. One interviewee means that by creating cross-functional teams, faster deliveries are possible, which in turn has created much better communication between all employees. Thanks to the t-shaped teams, there is no gap between the developers and the testers as is used to be. When organisations worked in a traditional project approach it was very common to blame other parts. The developers often experienced that testers only wanted to find defects in their code but when working as a team it both enhance the quality, which saves times. Further, many interviewees believe that the team contributes to reducing quality defects related to the misunderstanding when handover work since the work tasks do not have to been delivered between different parts of the organisations anymore.

4.5 Dependencies

Several interviewees argue that different dependencies, both internal and external, affect the agile way of working within the organisation. In this section the dependencies of the hardware and software development as well as between suppliers, partners and customers will be presented.

4.5.1 Hardware and software

Form the interviews it appears that most of the organisations have some kind of dependency between the hardware and software development. The interviewees argue that these two must collaborate in order to achieve great product quality for the final product. However, today it is difficult to make the hardware and software collaborate in an efficient way since many of the organisations use traditional waterfall methodology for the hardware development. Even if some organisations have implemented some agile practices for hardware development, such as daily standups and retrospective, it is hard to let go of the traditional waterfall methodology. The hardware is not as flexible as the software development due to long and fixed lead times. In addition, some interviewees also describe that it is difficult to deviate from the initial project planning, at the beginning of a new project, for the hardware development. Since the hardware and software development have such a dependency, software development is in most cases included in the initial project planning. Some interviewees mean that the hardware development could be seen as a bottleneck for the agile development.

Today, organisations have been compelled to interact with the small increment deliveries, from agile software development, to be combined with large milestone deliveries from hardware development. Many of the interviewees believe that it is suitable for the traditional approach for hardware development, which includes gateway planning and milestone deliveries. At least, as one of the interviewees argues, the hardware development process could be a combination between the agile and the traditional approach, where gateway planning is used in combination with sprints. Further, it has also been highlighted during the interviews that the hardware and the software development should be more independent from each other during the development process.

The interviewees mean that software development requires rapid change, something that is not advocated by hardware development. In software development, many changes often need to take place late in the development process, which many of the organisations argue can be difficult due to the hardware. The later a change takes place in software development that affects the hardware, the harder it will be to implement the change. Interviewees describe that when a physical component is developed it will be costly, time-consuming and difficult to change. However, one of the interviewed organisations means that they have "solved" the dependency challenge between hardware and software by having representatives from both hardware and software in the board of director. Further, the interviewed organisation argues that the key to success is to have close communication and understand each other's challenges. In addition, they have introduced simulations to reduce the software and hardware dependencies, which enables that the software development and testing phase can start earlier, even if the hardware is not physically available yet. They mean that hardware is not acting as a bottleneck anymore.

4.5.2 Suppliers, partners and customer

From the interviews another type of dependency was discovered, namely relationships between suppliers, partners and customers. The empirical findings show that there is a high variety if stakeholders are working agile or not. Even if they use agile practices it often collides with the organisation's agile approach. For example, one interviewee describe that it is common that suppliers deliver in sprints. However, the sprints and the PI-planning often overlap between the parties. One of the interviewed organisations has, for example, four PI's during one year while one of their suppliers only has three PI's. Moreover, the interviewees mean that some customer and suppliers are behind the agile development. One organisation describes that it is ideal to have something ready to deliver to the customer after each sprint in order to follow the agile practices with short iteration and fast feedback loops. However, if the customer has a traditional planning approach, the customer has usually already decided when they want their deliveries. One example from the interviews is that one organisation has three customer milestones in the period of two PI's. The interviewee states that this results in dislocations and that the customer milestones have to be mapped into the organisation's PI planning, in order to know when the features need to be finished and delivered. Additionally, the interviewees argue that close customer collaboration contributes to joint PI-planning, which facilitates the agile way of working.

Many of the interviewees have dependencies and collaboration with partners in order to deliver their product. They experience that many of their cooperation partners also are in their agile transformation journey, which results in confusion since they need to collaborate but at the same time work independent. Some of the interviewed organisations argue that they need to find collaborative models that can be synchronised between the partners in order to secure quality. Most of the interviewees have many suppliers and they do not have own mandate over the suppliers and cannot affect the suppliers' way of working. However, some of the organisations with fewer suppliers and with high dependencies have created agile contracts. Previously, traditional contract emphasises functionality but when changing to agile contracts, the focus has changed on how they should work together instead. One interviewee describes that this has resulted in better collaboration between the parties, where the organisation and the supplier are seeing each other more like partners instead of competitors.

4.6 Quality

Many of the interviewees express that it is overall very difficult to define quality and what good quality actually is. Concerning the agile methods and frameworks that the interviewed organisations are using, the interviewees argue that it is not clear how the quality perspective is described. For example, one interviewee means that the agile method Scrum solely defines thoughts of how to structure the work and which tools that can be used. Scrum does not provide a clear pattern to follow in order to achieve and ensure good quality and gives little support for the teams. The interviewee means that it is difficult for the team members to know if the quality is good enough. Other interviewees describe that SAFe is just a coordination and organisational model that describes the agile way of working. However, SAFe do not describes the quality perspective in a proactive way and how to secure the quality.

4.6.1 The definition of quality

From the interviews, it could be stated that the notion of quality is complex and hard to define. Even if some of the interviewees were working within the same department at the same organisation, their view of quality was very different. Some quotes from the interviewer's perspective of quality are presented:

Customer Satisfaction
Controlled by the customer
Meet the customer's expectation
The interaction between society and customer
We extend to which we want to exceed our customers' expectation
Influenced by one's competitors and society changes
Data-driven, transparency and ownership
Clear safety where security is included
Understand business value
It is about metrics

The majority argues that quality is related to customer satisfaction. Thus, it is hard for those organisations that only work with product development to define quality from the customer perspective. These organisations do not yet have end customers where they can receive feedback from and the quality is more from a society point of view. Most of the interviewees were talking about reactive quality and built-in quality, which they belive are included in the agile practices and frameworks. However, organisations that work with product development where talking more about proactive quality, which they do not perceive is described in the agile practices and framework. Moreover, the main focus of the interviews, when talking about quality, was on product quality and not so much about process quality. However, those interviewees that work on a more strategic level also include the importance of process quality and define it as continuous improvements of the organisation. There were a lot of reflections about quality when working scaled agile during the interviews. See Appendix II to see the interviewees' thoughts and questions about quality when working scaled agile.

4.6.2 Quality tools and activities

During the interviews, different quality tools and methods were mentioned in order to improve and ensure quality. Most of the interviewees argue that continuous integration, retrospective, DoD and automated tests are helping them a lot with the quality work. Thus, the quality tools and methods that are used varies a lot between the organisations and several of them are still working with traditional quality tools, like for example FMEA, issue-tree analysis, ishikawa diagram, six sigma and PDCA. However, most of the organisations have struggled to break down these quality tools into the agile way of working. Some of the organisations are trying to create a modification of the traditional tools in order to be more suitable for the shorter iteratives. Still, most of the organisations are in their agile transformation journey and are elaborating with the modifications. Further, many of the organisations that were interviewed use the program JIRA to break down customer requirements. From the interviews it appears that organisations want to find new quality tools and methods. Especially, organisations are looking to find new tools and methods to use on an upscaled level since today they are facing difficulties with the final integration and verification. An interviewee mentions that they have particularly difficulties to break down the final testing step and that the final integration still working according to the traditional waterfall model. Even though they have transformed into an agile way of working. They want to reduce the time for the final tests significantly but at the moment, they have difficulty knowing how to do it.

One difficulty that was discussed during the interviews is that developers and management have different levels of knowledge and works with different tools and systems. For example, one interviewee describes that quality manager prefers to work in Excel, something that the developers do not prefer Today, there may be difficulties in the cooperation between the parties and they would like to use common tools and systems that both parties prefer. The problem with project progress was another difficulty that one organisation, along with its customer, experienced. Therefore, in order to solve this, the teams and the customer are gathering at the end of each sprint to demonstrate the progress. When one interviewed organisation implemented demo-events, they have experience that they got a closer relationship and better feedback with the customer. There are also more interviewees claiming that demonstration events are good to ensure the quality.

Most of the interviewees state that testing is by far the most important tool used to ensure quality, which it also was when working traditionally. The difference is that the testing phase must be more flexible and faster when becoming agile. Today, all organisations are trying to achieve a higher level of test automatisation and phase out the manual tests to reduce the time between the feedback loops and also reduce the total lead-time. On the team level, most of the organisations have succeeded to automatise the tests and thereby increase the quality level. However, the challenge for all of the organisations that were interviewed is to automatise the tests on an upscale level. The reasons why the organisations do not have automated tests on a higher level differ from organisation to organisation. Some argue that there is a lack of knowledge and understanding of the importance of the automated tests. While others mean that they need to create an automated testing tool that can reuse the tests. However, some argue that to be fully automated are nothing to strive for since test automatisation only searching for defects due to customer requirements and not finding other defects. Another problem when testing on an upscaled level is to derive which team that is responsible for a software bug. From the interviews, it was discovered that each organisation works differently to track bugs. In most of the organisations the teams are responsible for tracking software bugs even on an upscale level. If a bug is found and not easily can be tracked to a specific team, the tracking process will be very time-consuming. One organisation has implemented a main-track team, which takes over the responsibility of the code after the team reaches a certain test level. The main-track-team is responsible for finding the root cause and then handing over the software bug to the responsible team.

4.6.3 Quality metrics

During the interviews, metrics regarding quality and improvements were discussed. The empirical findings indicate that relatively few organisations have structured metrics related to quality when working scaled agile. Many organisations have not changed or modify the old quality metrics after the agile transformation. Typical quality metrics for the organisations today refer to different quality defects reports. Further, some organisations that do not have any current agile metrics claim that they rely on that teams know what to perform and fulfill their duties. Another interviewee means that they still have traditional measurement methods where they create metrics based on milestone audits. On the other hand, another interviewee argues that they want to create better metrics and implement them in the future. Those organisations that already have implement agile metrics, also have implemented these in a visualised way by having digital boards on the walls. The organisations that are developing new agile metrics related to quality mean that they also want to visualise the metrics. Today, one interviewee describes that the metrics only can be found on internal communication channels, for example, SharePoint, or are informed by mail. However, another interviewee means that it is of greatest importance to developing metrics that everyone understands within the whole organisation. The same interviewee argues that the risk with metrics is that it is difficult to get a good overview of the organisation and instead only focus to improve the metrics.

One organisation that has developed new metrics, is measuring them for all levels within their organisation. On the team level they have four different metrics:

- *Progress*- Status of features and enablers during PI execution.
- *Quality*- Unit test coverage, automation and defect density.
- *Effectiveness*- Defect stories completed and average feature cycle time.
- *Predictability*- The planned amount of work in one sprint versus the actually performed work.

For the higher levels within the organisation an accumulate value from the different teams is summarised. Moreover, the interviewee argues that the metrics create a positive effect on the team's performance and motivates the team members to develop the metrics. For example, the metrics "predictability" contributes to that the team members themselves are responsible for planning how much time a story or a feature will take. The value of the metrics indicates how teams mature over time.

Another organisation that has experiences in agile metrics, measures the number of internal defects reports. They have an established work level agreement, which describes how fast teams should respond to defect reports. In addition, they have decided a maximum number of defect reports that could exist at the same time and how many days these could exist before they must be handled. All the metrics are visualised on boards for the teams in order to see if they need to take actions. They have also developed a four-level scale to handle the defect reports. The scale is designed depending on how critical the defects are whereas the fourth and highest level is the most critical one and implicates a delivery stop until the quality defects are solved. The same interviewee describes that they also have a big focus on throughput time measurements. The throughput time refers to how long time it takes for the teams to get feedback from the higher levels. The organisation has determined how long time it should take for each level to get feedback. This is something very important for the organisation and they are working continuously to reduce the feedback time. The measurement and the continuous work with the throughput time have resulted in faster feedback loops, from days to only hours. This in turn result in huge profits since they can deliver faster to the customer, which make the customer rely more on them and choose their product over competitors.

However, another interviewee indicates that they were measuring and analysing the metrics more often before implementing the agile way of working. Due to the agile transformation, they have received fewer defect reports and therefore they do not analyse the metrics in the same structured manner as before. The interviewee means that teams have a feeling for defect reports and only need to look at them sometimes.

4.6.4 Quality standards, regulations and legal requirements

The empirical findings show that the organisations have many standards, regulations and legal requirements that they have to follow. Some of the standards, regulations and legal requirements that were discussed during the interviews and that many organisations have to follow are ISO 90115, ISO 27000, ISO 26262 and Automotive Spice. Furthermore, some of the interviewees state that, for example, the Automotive Spice is a hard define process standard, which contradicts the agile methodology. Automotive Spice provides rigid traceability from code to the requirements but this, in turn, requires heavy documentation. Another interviewee questions if Automotive Spice actually increases the quality. A design document for each code unit is needed when following Automotive Spice, which requires much administrative work. According to one interviewee, organisations that not follow the Automotive Spice standard, only need to test on the highest level to ensure that everything's fulfills the requirements. However, the same interviewee also discusses that when the product is safety-critical with legal responsibilities, the Automotive Spice standard is useful. Additionally, the interviewees mean that the ISO 26262 standard also requires much documentation and focusing on contracts and planning, which also contradicts the agile methodology.

Another interviewee describes that there is not really an opposed between the Automotive Spice and the agile practices, and the same applies for ISO 26262. Nevertheless, the interviewee cogitates if there is some other way the quality standards could be followed in a more efficient way. Overall, many of the interview organisation do not seem to have found a way to break down the standards, regulations and legal requirements into shorter sprints and are still working with them as they did before.

4.6.5 Proactive quality

According to the interviewees, the agile principles and frameworks are focusing on reactive quality, such as retrospective and tests of the software. In the agile frameworks it is not described how to avoid mistakes in a proactive way. For example, some of the interviewees cannot find any new methods that are describing how to find mistakes in advance, such as FMEA was used in traditional projects. Instead, they are trying to modify FMEA to fit into an agile way of working. However, they have not successfully found a way to break down FMEA into smaller increments today. Therefore, they still have traditional project managers and methods to ensure quality even if they have made an agile transformation. Some organisations have seen that the traditional methods do not match the agile way of working, which has complicated proactive quality work compared to earlier.

Others mean that the agile principles and frameworks are only focusing on fast deliveries and flexibility and are not looking at the product quality in the long-term perspective. Organisations that are working with a product that should last for a longer time see issues with the agile way of working since they have not found a way to ensure long-term product quality. They mean that they have to be sure that the old functions are working to be able to add new functionalities. Due to the issues that have arisen with proactive quality, some organisations have a proactive quality department that is working to prevent future unwanted situations.

Many of the interviewees pointed out that they want to be able to utilise data that they collect from customers in order to proactively prevent mistakes. Some say they have access to high volume data but are not currently using it, while others have less access to customer data. Further, some organisations do not even have any customer data since they are working with product development and do not have products on the market yet. Many organisations believe that through utilising customer data it will generate high efficiency and proactivity in the organisation, and hopefully, they can be able to utilise such data in a near future. However, one of the interviewees indicates that they are already working actively to analyse data collected from customers. They have applied data analytics such as Big data and Artificial intelligence (AI) to find software bugs. By using AI, the organisation can quickly see where the cause of the software bugs is and thereby, identify, derive and correct the bugs even before the customers discover them. Thus, they will increase the quality of what they deliver.

4.7 Best practices

The best quality enhancement activity within the agile practices highly varies from organisation to organisation and also depending on which interviewee that was interviewed. However, some common denominations were found regarding which activities that seem to work best for organisations that are working agile. Several of the interviewees state that test automatisation is the most important activity concerning the software quality. Although, since the empirical findings indicate that not many organisations have come so far with automatisation of tests this practice is rather something that organisations know will improve the quality but not that many have applied yet at a large scale. Further, the majority of the interviewees argue that CI contributes to high quality for software development. They mean that the CI ceremony gives fast feedback and captures problems earlier than before. One interviewee stresses out that the sooner a problem can be revealed, the faster it can be solved and thereby they can ensure quality in an earlier phase. In addition, CI enables more stable tests on an upscaled level.

Some interviewees mean that they are not able to decide which quality enhancement activities that is the best since they have not come that far in their agile journey. One interviewee believes that FMEA will be important even in the future agile journey. The same interviewee also arguing that one of the most important activities for them is that they have changed the names of, for example, quality tools to design tools and quality assurance plan to development plan. The main reason for changing the names is that the word “quality” should not be in focus since quality should be built in within the agile practices and is everyone's responsibility. The interviewee means that it is about changing the mindset and makes it easier for the teams to take and feel responsibility for those tools and activities when there are new names. It is of greatest importance to push the responsibility down to the team and individual level when working agile. Furthermore, several others of the interviewees assert that they already can see how the agile practices contribute to many benefits, even though they have not worked agile for so long. They mean that the retrospective ceremony is one activity that enhances the quality of work because it compels everyone to continuously reflect upon the performance. Additionally, some argue that the PI-planning and other activities like for example, big room planning sessions are also something that has increased the quality. These activities contribute to increased transparency, decisions are clearly made, and different dependencies are emphasised. The tasks are prioritised in a collective way and the teams are united with the goals for the next sprint. Also, the big room planning session gives the teams a chance to highlight their experienced problems so other teams can learn from those experiences.

One interviewee describes that the most valuable activity for them was to develop a standard, which describes the best-known way of working agile. To be able to continuously improve the work, a standard on the working processes is a prerequisite. Otherwise, there is no “attachment” and it is very difficult to spread information to other teams. The same interviewee also claims that it is important to respect the team members’ own wish and give them the opportunities to improve. In addition, the same organisation has currently introduced that 10% of the total workload should be dedicated to improvement work.

Many organisations have also highlighted that the most important activity is to work for that the teams should feel ownership over the quality. One interviewed organisation that has more experiences of working agile means that the quality will increase the most if you let the teams have an end-to-end commitment. Even though this transformation is very hard to implement throughout the organisation, the teams really grow both as a team and as individuals after a while. The same interviewee states that their lead times have almost reduced by the half related to the featuring development since the implemented end-to-end commitment. Further, the activitt CI have contributed to support the achievement of end-to-end commitment and faster feedback loop have been received.

Another organisation that has come a long way in their agile journey, describes that the best activity for them is related to the coordination and synchronisation of metrics for all the different levels within their agile framework. The interviewee also expresses the importance of having the metrics visualised, especially when the numbers of agile teams are rising, and the organisation is scaling up.

Overall, the interviewees believe that the agile way of working contributes to several benefits concerning the quality. They describe that there are several different activities and parts of the activities that contribute to good quality. Some interviewees state that fewer bugs reach the customer nowadays, while others argue that the quality overall has increased. One interviewee also means that the agile way of working facilitates "*doing the right thing in the right time and do it much faster than before*" which is the fundamental purpose of why implementing the agile practice in the first place. Thereby it is much easier to avoid waste and focusing on the most important things. Also, many interviewees mention that without doubt the best agile practice is close customer cooperation. The closer customer cooperation, the better product development.

4.8 A summary of the main empirical findings

From the interviews several challenges were identified regarding quality assurance when working scaled agile. Many of the organisations have not been working upscaled for such a long time. To sum up what have been seen during the interviews the following challenges have been found:

- *Lack of leadership*- From the interviews, it turned out that there is a different view of management's understanding of quality. In some case it is the management that has implemented the agile way of working in order to achieve better quality, while others mean that management does not understand the meaning of quality when working scaled agile. For example, some interviewees said that the management does not understand what type of technology the teams need to improve quality such as test automatisation.

- *Immature teams*- The majority of the interviewees are describing that one of the biggest challenges when working agile is the immature teams. Many organisations have not been working agile for so long, and the teams are lacking in general knowledge of what is included in their roles and responsibilities. Also, it is very difficult to get everyone to strive for the same direction and not falling back on old working behaviors. Another difficulty that was brought up during the interviews was that some of the organisations have distributed agile teams. They have this in order to secure the right competencies and because of the man-hour work are less expensive in other countries. However, the distributed teams are causing difficulties regarding the communication and to create a common understanding of the agile way of working since the do not have the agile mindset. A third challenge that was mentioned in some of the interviews is that it is hard to build well-performing and empowered teams and ensuring the t-shape competences. This due to that organisation is facing a high turnover rate of the employees.

- *Hardware and software dependencies*- It is difficult to interact the software and hardware development since the hardware development still is working due to the traditional practices with milestone deliveries. However, it is hard to let go of the traditional waterfall methodology. The hardware is not as flexible as the software development due to long and fixed lead times.

- *Suppliers, partners and customers dependencies*- From the empirical findings, it could be stated that there is a high variety if stakeholders are working agile or not. Even though a supplier or a customer is using the agile practices it often collides with organisations' agile approaches. The sprints and the PI-planning often overlap between the parties and some customer might not even want small incremental deliveries. Further, organisations with partners that also are in their agile transformation journey, have struggles to collaborate but at the same time work independent.

- *Hard to to break down quality tools into an agile way of working*- Many of the organisations are still working with traditional quality tools today. However, many find it challenging to break down these quality tools into smaller iterations. Organisations want to find new quality tools and methods to use on a upscaled level since they are today facing difficulties in the final integration and verification phase.

- *Lack of agile metrics related to quality*- From the interviews, it could be seen that few organisations have structured agile metrics related to quality. Many organisations have not changed or modify the metrics after the agile transformation. In addition, not many organisations have visualised metrics, and many rely on that the teams know what and how to perform.

- *Standards, regulations and legal requirements contradict to the agile methodology*- For those organisations that are compelled to follow strict standards, regulations and/or legal requirements have challenges to break down these into smaller sprints. Further, many of the standards, regulations and legal requirements require heavy documentation and a big focus on contracts and planning.

- *The way of working related to quality are not described in the agile methods or frameworks*- The interviews show that many organisations find it extremely difficult to describe quality: What that is expected to be included as well as to know when the quality is good enough? The interviewees also arguing that the agile working methods and framework are not describing how to secure quality in a proactive way. Some of the interviewees want to find new methods to find mistakes connected to quality in advance, such as FMEA was used in traditional projects. Especially, for those organisations that are working with products that should last for longer time intervals see many issues with the agile methods and frameworks.

- *Lack of test automatisation*- The majority of the organisations claim that test automatisation is a key factor to ensure high quality. However, most of the organisations still have lots of manual testing. The challenge is to automatise the tests on an upscale level and to create an automated testing tool that can reused the tests. The interviewees believe that the difficulties with test automatisation depend on that the management does not understand the importance of automated tests and that it is expensive to implement.

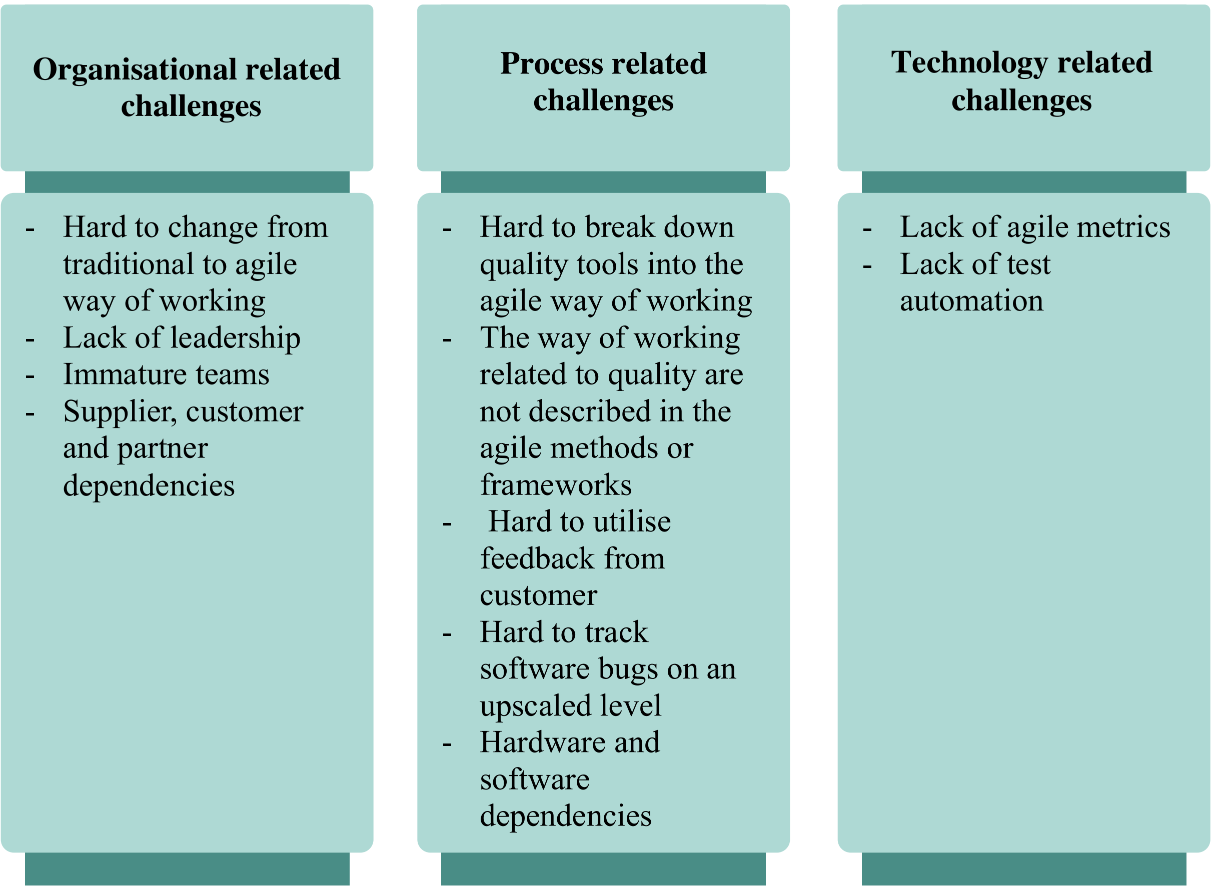
- *Hard to utilise feedback/data from the customers*- Several of the organisations have access to high volumes of customer data but are not currently using it. Some organisations do not have access to any customer data at all since they are in the product development phase and not have any product on the market yet. For those that have the access, the problem is related to that they do not know how to utilise the customer data in order to make the organisation more efficient.

- *Hard to change from traditional to the agile way of working*- It can be stated from the interviews that it is hard to change from the traditional way of working to start working agile. Many organisations still have a quality department even if it should be seen more as a supportive function rather than a depertment. However, many organisations still have a quality department since many teams are immature and needs the support. Also, some of the interviewees mean that the transformation has given the teams more responsibilities but they have no authority.

- *Hard to track software bugs on an upscaled level*- From the interviews, it was discovered that each organisation works differently to track software bugs. Most of the organisations experience it hard to track software bugs on an upscale level. The difficulty is to derive which team that is responsible for a software bug.

5. Analysis

The analysis will combine the empirical findings compared to the literature framework in order to answer the research questions. Further, this chapter will discuss how to overcome the challenges related to quality when working scaled agile.



Organisational related challenges

- Hard to change from traditional to agile way of working
- Lack of leadership
- Immature teams
- Supplier, customer and partner dependencies

Process related challenges

- Hard to break down quality tools into the agile way of working
- The way of working related to quality are not described in the agile methods or frameworks
- Hard to utilise feedback from customer
- Hard to track software bugs on an upscaled level
- Hardware and software dependencies

Technology related challenges

- Lack of agile metrics
- Lack of test automation

Figure 5.1: The dividing of the empirical findings.

The structure of the analysis is divided into three main categories, see figure 5.1, based on the main findings from the empirical data. The challenges are divided into organisational related challenged, process related challenges and technological related challenges.

5.1 Organisational related challenges

The agile way of working has increased in popularity steadily for the last couple of decades. When traditional organisations transfer to the agile methods, they are also facing challenges related to the agile transformation process. Many of the challenges when implementing the agile principles and practices are related to the organisation and the people inside the organisation.

5.1.1 Hard to change from traditional to agile way of working

The literature framework discuss that it will require time and patience form an organisation to change culture due to the agile transformation. Boehm and Turner (2003) argue that people's mindset and behaviors are not easy to change and an organisational transformation might therefore be slow.

The empirical findings show that some of the interviewed organisation has been working agile for many years. However, they are still on their agile journey and there are extremely few organisations that have been working on an upscaled agile level for a long time. Several of the interviewees pointed out that they started implementing the agile way of working in small pilot projects in different teams and then spread it to the remaining teams and lastly implemented it on an upscale level. This is in line with Ambler (2010), which argues that organisations first should create pilot projects. It is important to gain lots of experience and identify the risks before implementing it throughout the whole organisation. In addition, according to an article published by De Smet et al. (2018) it is common for organisations to first begin with small pilot projects. Then the organisations could over time build up strong agile practices. However, one of the interviewed organisations had not used any pilot project when implementing the agile way of working. Instead they implemented the agile way of working directly across the entire organisation. This is not accordingly to what De Smet et al. (2018) describe but the interviewee means that their agile transformation was successful implemented due to the radical transformation through the whole organisation. Thus, Nerur et al. (2005) discuss that the result from the agile implementation is depending on the effort of the organisation. When transforming in such a radical way it requires that the whole organisation make a huge effort. Thus, the same authors argue that when an organisation wants to scale up the agility throughout the organisation, the leaders must be willing to embrace the precepts and enhance the capabilities in order to become successful in the implementation.

Furthermore, when there are only few teams within the organisation, the agile transformation do not requires such a change. The literature shows that transformation on the project level is fairly simple. However, there are challenges when the organisation is scaling up agility throughout the whole organisation (Vaidya, 2014). Also, Tipnis (2018) describes that one of the challenge when transform from a traditional organisation is that employees are remaining old methodologies. The author indicates that old structures and working behavior are difficult to change. This corresponds to the empirical findings. The majority of the interviewed organisations have the same division of labour as before the agile transformation. The empirical findings also show that the quality department still exists for several organisations even if they are trying to use the quality department more as a supportive function. However, the organisations that have been working agile for a longer period or organisations that have been agile from the beginning have no quality department. The responsibility of quality is distributed to the teams, everything that concerns quality from the beginning to the end of a project. These organisations have succeeded to distribute and break down the responsible for the quality to the team level.

5.1.2 Lack of leadership

The agile transformation requires a new kind of leadership, so called servant leadership (De Smet et al., 2018). The empirical findings describe that organisational leadership are the biggest barriers and enablers of successful agile transformations. Based on the interviews it turns out that management have different view and understanding of the quality work. Some mean that it is the management that created the implementation in order to achieve better quality, while others mean that management does not understand the meaning of quality when working scaled agile. Some organisations argue it is difficult to get support from management since they do not understand what type technology the teams need in order to improve quality. According to De Smet et al. (2018), leaders must first change their own mindset and behavior before they transform the team and finally the entire organisation to succeed with the agile transformation.

It is not only the organisational culture that changes when implementing the agile working practices, it is also the organisational structure according to Nerur et al. (2005). The leadership is changing from command and control to become coaching and collaborative. During the interviews, it was discussed that managers have taken a new role due to the agile transformation. Instead of delegating, they mean they should coach and support the teams. However, some of the interviews objects mean that there is difficult now when having a decentralised organisation. Today, it is difficult for managers to know what they have for authorities. According to Hultgren and Lyhammar (2017b), a servant leader should create visions and eliminate obstacles. Further, the agile leaders should listen, support and educate teams in problem identification and in the decision-making process so that teams understand what decisions they can take by themselves or not. It is difficult for the management to balance the right amount of cooperation with self-organised teams within the organisation. Further, the authors suggest that an employee survey could be done in order to analyse the ability of the leadership within the organisation. For instance, the team can rate the leader from 1-10 based on how well the leader perform. Depending on the grade the team together with the leader should elaborate which areas that could be improved.

5.1.3 Immature teams

During the interviews, a large part of the discussions was related to the agile teams. The majority of the interviewees described that one of the biggest challenges when working agile is the immature teams. All the interviewed organisations were striving to push down the responsibilities to team level. Further, the interviewees argue that they want self-organising teams that define their own work and have a quality perspective in everything they do. This goes in line with the literature and the agile way of working. The foundation of SAFe consists among others of having self-organising and empowerment teams (SAFe Team-level, 2018). Also, Appelo (2011) describes that the teams should be responsible for coordinate and design their own tasks. It is of the greatest importance that the team is responsible for the result, as a part of the empowerment. It was also expressed by most of the interviewees, that one of the most important things is to make sure that the teams feel ownership over their work. This due to the difficulties to change the organisation and that the management still has strong control within the organisation. Some of the interviewees mean that the responsibilities have been distributed to the teams but the authority is still on a management level. Many interviews objects mean that today the teams are lacking of general understanding of what you can decide by yourself, which are your mandate and what is included in your role. Appelo (2011) argues that it is first when the individuals feel that they have authority, they will become more motivated related to their daily work. Although, it is claimed from the literature that it will take a long time to reach this since it induces an organisational cultural and structural change (Tipnis, 2018). Cockburn and Highsmith (2001) mean that within this area, supporting and coaching is very important from the management.

Another challenge that was mentioned by some of the interviewees is that it is hard to build well-performing and empowered teams and ensuring the t-shape competences. This due to that organisation is facing a high turnover rate of the employees. According to Castka et al. (2001) the group culture and other human factors are considered to be important for the success of an agile team. Further, Cockburn and Highsmith (2001) describe that the level of team maturity reflects upon how agile the team will be. A mature team could be related to high commitments to the work and the result, effective communication, and agile planning.

The challenges by having distributed agile teams were also highlighted during the interviews. One organisation has testers and developers in the same team but the developers are located in another European country because of the man-hour work are less expensive there. While another organisation has co-located teams with different teams set-up in different geographical areas. This in order to have the right competencies and resources. Also, in order to have a competitive advantage by having several sites where the largest customers are positioned. However, the distributed teams are causing difficulties regarding the communication and the understanding of the agile way of working. Lee and Yong (2010) describe that distributed teams have increased during the last years due to the globalisation of software development. Organisations want to deliver software with high quality at higher speed with lower costs. In order to succeed, many organisations have chosen to use developers located across the world where costs are lower but at the same time knowledge is available. The interviewees with distributed teams state that it would have been much easier if all the team members could have been located in the same place. Hultgren and Lyhammar (2017b) argue that there are many difficulties that arise when having distributed teams. The authors also describe that the challenges related to distributed teams could be overcomed by better communication. Implement live windows, which means setting up a big screen with live camera and microphone at both teams' location, could increase the communication. Further, Lee and Young (2010) recommend to enforcing meetings, hold frequent voice and email exchange in order to improve the communication. In addition, it is important to create distributed workshops to create common goals and visions, which will result in a community between the teams.

5.1.4 Supplier, customer and partner dependencies

Nerur et al. (2005) describe that it is important to find customers that are willing to collaborate and actively be involved in the development process. The interviewees argue that close customer collaboration facilitates the agile way of working. Further, the empirical findings show that there is a high variety if the customers of the intervived organisations are working agile or not. One interviewee describes that their teams want to deliver functionalities on a regular basis. This due to following the agile practices with short iteration and fast feedback loops. However, if the customer has a traditional planning approach, the customer has usually already decided when they want their deliveries. Tipnis (2018) argues that even if the customers are not working agile, it is extremely important to inform and make the customers understand how the agile way of working will affect them and what benefits that could be gained from this.

However, even if the customers use the agile principles and practices, some of the interviewed objects argue that, for example, PI-planning and other agile activities often overlap between the different parties. Boehm and Turner (2005) describe this challenge regarding to have different sprint and/or project life cycles. The authors recommend that the teams might need to change and adjust their deliveries in order to synchronise with the customer. The authors also imply that the management should focus to put lots of attention and effort to synchronise the teams and the customers, since it is not easily done. Additionally, one interviewee recommended creating agile contract with the supplier in order to solve the challenge with unsynchronised project cycles. This is feasible when organisations have few suppliers and the dependencies are high between the parties. However, most of the interviewees have many suppliers, which means that they do not have own mandate over the suppliers and cannot affect their way of working on their own.

5.2 Process related challenges

The biggest challenges related to process includes problem to break down traditional quality tools, standards, regulations and legal requirements into smaller sprints. Further, problems related to utilisation of customer feedback, track software bugs on an upscaled level and dependencies between hardware and software will be analysed in this section.

5.2.1 Hard to break down quality tools into the agile way of working

Most of the interviewed organisations still work with traditional quality tools. The empirical findings indicate that many of the organisations cannot find any new methods that describe how to find and prevent defects in advance, such as FMEA are used in the traditional project approach. From the interviews it appears that organisations want to find new quality tools and methods, especially on an upscaled level. Many are trying to modify the traditional quality tools to be more suitable for the agile way of working. However, they have not successfully found a way to break down these into smaller increments and therefore still are using the traditional quality tools to ensure quality. Additionally, many interviewees argue that the traditional methods do not even match the agile way of working. Thus, Ahmed (2018) means that by maintaining the design and structure of the traditional quality tools the team can ensure that potential failure points are being analysed in time.

Moreover, some organisations are facing problems when the responsibilities are distributed to the team level. Some teams do not have the knowledge on how to use traditional quality tools. One of the interviewees describes that they have changed the namnes from ”quality tools” to ”design tools” in order to change the mindset within the organisation. Also, they are trying to change the system, which the quality tools are being used in so that the teams can start to use these tools more frequently without needing help.

5.2.2 The way of working related to quality are not described in the agile methods or frameworks

Many of the interviewees express that it is overall very difficult to define quality and what good quality actually is. Concerning the agile methods and frameworks that the interviewed organisations are using, the interviewees argue that it is not clear how the quality perspective is described. Scrum does not provide a clear pattern to follow in order to achieve and ensure good quality and gives little support for the teams. Timperi (2004) also claims that there is little focus in literature how agile methodologies working with quality assurance practices. However, the SAFe framework (SAFe Built-in Quality, 2018) addresses six agile software development practices to ensure built-in-quality:

- *CI*
- *Test-First*
- *Pair work*
- *Collective ownership*
- *Agile Architecture*

Even though SAFe describe quality practices, some interviewees claim that SAFe is just a coordination and organisational model. SAFe do not describe the quality perspective in a proactive way and how to secure the quality. Most of the interviewed organisations were talking about reactive quality and built-in quality, which could be seen in the agile practices and frameworks. However, organisations that work with product development where talking more about proactive quality, which they do not perceive is described in the agile practices and frameworks.

Moreover, the main focus of the interviews, when talking about quality, was on product quality and not so much about process quality. Sommerville (2007) also argues that the process quality is more complex and not that easy to understand and monitor in the same way as the product quality. However, the interview objects that operate more on a strategic level also include the importance of process quality and define it as continuous improvements of the organisation which they not can find in the methodology. Organisations that are working with products that should last for a longer life cycle see issues with the agile way of working since they have not found a way to ensure long-term product quality. They mean that they have to be sure that the old functions are working to be able to add new functionality. Due to the issues that have arisen with proactive quality, some organisation has a proactive quality department that is working to prevent future unwanted situations in advance. In addition, Qumer (2007) means that there are few types of research focusing on quality assurance in scaled agile software development.

5.2.3 Hard to utilise feedback from customer

Many of the interviewees point out that they want to be able to utilise data that they collect from customers in order to proactively prevent mistakes. Some say they have access to high volume of data but are not currently using it, while others have less access to customer data. According to Sauro and Lewis (2016), A/B testing is a method that can collect and analyse huge amount of data in order to improve an organisation. In A/B testing, one version of a system is modified and compare to the original one. Often, half of the users are exposed to the original version, while the other half is exposed to the changed version. Customers are randomly exposed of one of the variants of a system and after the test period, the variant that has been most successful are implemented. However, in this approach, A/B testing will act as retroactive quality method rather than proactive. Many organisations believe that data collection from the software will generate high efficiency and they hope that they can use data proactively in the near future. An organisation that works with proactive quality is analysing data collected from customers. They apply data analytics such as Big data and Artificial intelligence (AI) to find software bugs. By using AI, the organisation can quickly see where the cause of the software bugs is and thereby, identify, derive and correct the bugs even before the customers discover them. Thereby, they will increase the quality of what they deliver.

5.2.4 Hard to track software bugs on an upscaled level

A problem that was discovered during the interviews was to derive which team that is responsible for a software bug on an upscaled level. From the interviews, it was found that each organisation works differently to track bugs. In most of the organisations, the teams are responsible for tracking software bugs even on an upscale level. However, they found it very difficult and time consuming to find the root cause when defects arise. Unfortunately, there are few studies on how to track software bug on an upscale level. One organisation has worked extra to solve this problem. This organisation has implemented a main-track team, which takes over the responsibility of the code after the team reaches a certain test level. The main-track-team is responsible for finding the root cause and then handing over the software bug to the responsible team. Further, the same organisation has decided a maximum number of defect reports that could exist at the same time and how many days these could exist before they must be handled. They have developed a four-level scale to handle the defect reports. The scale is designed depending on how critical the defects are whereas the fourth and highest level is the most critical one. The fourth level implicates delivery stop until the quality defects are solved.

5.2.5 Hard to break down standards, regulations and legal requirements

All the interviewed organisations are following some kind of standards, regulations and/or legal requirements. Depending on which industry the organisation is operating in, there are more or fewer standards, relations and legal requirements that have to be followed. Most of the interviewees state that the way they work with standards etcetera have not changed after they implemented the agile way of working. They also mean that the different standards, regulations and legal requirements do not affect their agile working methods. Although, some of the interviewees express that the standards are conflicting with the agile way of working. They want to examine if the quality standards could be followed in a more efficient way in order to be more suitable for the shorter iterations. To follow standards and to be certificated it requires heavy documentation and a high focus on contracts and planning, which contradicts the agile manifesto, which states ”*working software/product over comprehensive documentation*” (Beck et al. 2001). However, no one of the interviewed organisations has found a way to break down the standards, regulations and legal requirements into shorter sprints. Also, not many of the interviewees do not consider that this is a problem. Further, there is not much literature in the field either which making it difficult to anlyse this challange.

5.2.6 Hardware and software dependencies

Many of the interviewees have some kind of dependency between hardware and software development. Today, organisations have difficulties to find collaboration between hardware and software since many of the organisations use the traditional waterfall methodology for the hardware development. However, organisations argue that these two must collaborate in order to achieve great product quality. Youn and Yi (2014) also mean that software and hardware are interconnected and must be developed simultaneously in order to ensure quality and safety.

All interviewed organisations that have dependencies between software and hardware experience that it is difficult to find an agile way to work with hardware. Even if some organisations have implemented some agile practices for hardware development, such as daily standups and retrospective, it is hard to let go of the traditional waterfall methodology. Some interviewees argue that the hardware development process could be in a combination between the agile and traditional waterfall approach, where gateway planning is used in combination with sprints. Other interviewees mean that the hardware and software development should be more independent from each other during the development process. Cprime (2012) argues that it is very common that organisations only introduce agile working methods that are suitable for software development for the entire organisation, such as Scrum. However, chosen agile methods are not always suitable for hardware development. Therefore, it is important that organisations understand the dependence between hardware and software when implementing agile working methods. In order to create good collaboration, Solera (2016) suggests that hardware development should work according to Commitment-Based Project Management (CBPM). By breaking down the hardware functionality, DoD will be more easily defined and will enable better and faster deliveries. In CBPM, each team member should deliver hardware features until a specific date. Compared to Agile software methodologies, CBPM does not need domain dependency and requires less iteration.

The differences of small iteration deliveries are the main challenge that makes it hard to combine software and hardware development (Youn & Yi, 2014). The organisations that have software and hardware dependence also find the challenge of that software development requires rapid change. Something that is not advocated by hardware development. In software development, many changes often need to take place late in the development process, which many of the organisations argue can be difficult due to the hardware. Further, software development needs thousand test cases before a delivery, while hardware requires less testing (Cprime, 2012). Therefore, software development should be incorporated from day one and a solution is to run simulations of the hardware in order to be able to test the software at a previous stage. This has been realised in some organisations, which made them introduce simulations to reduce the software and hardware dependencies. That enables the software development and that testing phase can start earlier, even if the hardware is not physically available yet. However, according to the literature, it is not the same to perform simulation testing as real-life hardware testing.

5.3 Technology related challenges

The technology related challenges are divided into two parts, namely, the lack of agile metrics and lack of test automatisation.

5.3.1 Lack of agile metrics

The empirical findings indicate that relatively few organisations have structured agile metrics related to quality and that they have not changed or modified the metrics after the agile transformation. Kupiainen et al. (2015) also claim that they have seen a demand for new software metrics after the agile invention but that traditional software metrics also might be applicable to agile software development.

Further, many organisations want to find new agile metrics for the entire organisation, which contradicts to the literature. According to Hultgren & Lyhammar (2017c), there should not be any standardised metrics for the entire organisation. If an organisation wants common metrics they should be chosen with very careful thought. They mean that metrics should only be team-specific and metrics that are no longer used must be replaced. Therefore, it is difficult to use common metrics for the entire organisation. Furthermore, the authors mean that as few measurements as possible should be used, since heavy measurement not corresponds to agility. One interviewee indicates that they are performing less measuring after the agile transformation since they have received fewer defect reports. Therefore, they do not analyse the metrics in the same structured manner as before. The interviewee means that teams have a feeling for defect reports and only need to look at them sometimes.

From the empirical finding, it was discovered that interviewees believe it is difficult to get a good overview of the organisation and that managers often only focus to improve metrics. Both Mukker et al. (2014) and Hultgren and Lyhammar (2017c) also argue that metrics often are used incorrectly. Instead they are used to reach numbers rather than to improve overall project quality and progress. The literature argues that even if the value of metrics increases, the numbers can be misleading since it might be major technical debt behind them. However, Kupiainen et al. (2015) mean that if an organisation using metrics correctly, they will make better decisions, understand quality and business objects and improve software development communication, processes and tools.

One organisation that has developed new metrics, use four different metrics at team level; *Progress*: status of features and enablers during PI execution. *Quality*: unit test coverage, automation and defect density. *Effectiveness*: defect stories completed and average feature cycle time. *Predictability*: the planned amount of work in one sprint versus the actually performed work. The interviewee means that the metrics "predictability" contributes to a positive effect on the team performance and motivates the team members to develop the metrics by themselves. However, Hultgren & Lyhammar (2017c) are skeptical to metrics since they argue that metrics often are used as a control measurement and that team feels monitored. Therefore, it often drives to bad behaviors among employees, especially if there are rewards related to the metrics. On the other hand, the interviewee stated that when team members are responsible for planning how much time a story or a feature will take, such as predictability metrics, they would feel engaged in the decision-making process. Therefore, it is important that organisation inform the purpose of measurement to employees (ibid). There should be full transparency on what is being measured and that everyone knows what is being measured.

5.3.2 Lack of test automation

Most of the interviewees state that testing is by far the most important tool used to ensure quality, but the testing phase muste be more flexible and faster when becoming scaled agile. This is also supported by Kronfält (2011), which means that tests are an essential part in order to secure that the software has the right functionality and works in the intended way. Without successful tests, it will result in several problems in terms of cost and time losses. During the empirical study, it was discovered that few of the interviewed organisations had full test automationsation on an upscaled level. However, all the interviewed objects argue that they are trying to achieve a higher level of test automatisation and phase out the manual tests. In that way, the time between the feedback loops and total lead-time will be reduced. According to Jansing et al. (2015), test automatisation will among others create faster deliveries, eliminate defects due to human errors, reduce cost due to fewer resources needed and thereby, the overall quality will be better. However, the authors argue that test automatisation requires that developers have good knowledge about the product and how to write the tests. Further, it could be difficult to reuse automate tests when adding new functionality. The difficulty to reuse automated tests was also discussed during the interviews. Many interviewees argue that they need to create an automated testing tool that can be used more than once. This is also argued by the SAFe framework (2017). The framework describes that the biggest challenge due to agile testing is that the testing is exponential in relation to every development cycle. Old functionalities must be tested in combination with the new once. Further, as Jansing et al. (2015) says, developers need good knowledge of test automatisation. However, some interviewees mean that the reasons why they do not have automated tests on a higher level is a lack of knowledge and understanding of the importance of the automated test from the management side. Organisations mean that it is sometimes hard to use test automatisation since the tests only searching for defects due to customer requirements and not finding other defects that could be critical.

6. Discussion

This chapter will discuss the findings of the report based on the analysis. First will the challenges be discussed followed by the best practices to ensure quality when working scaled agile.

6.1 Challenges

Based on the empirical findings, challenges that were discovered have been grouped into three areas, namely organisational related challenges, process related challenges and technology related challenges. All of these will be discussed in the following chapter.

6.1.1 Organisational relate challenges

To do an organisational transformation and implement changes related to the organisational structure and way of working are challenging. The research was conducted at several organisations and it was difficult to identify if the organisational related challenges that were found, had any common denominator. This because it was a large variation between the organisations of how far they had proceeded within the agile journey. There were also a large variety of different challenges internally at organisations, depending on interviewee and which department that was interviewed. Several organisations have had teams that have been working agile since the beginning of the 21st century. However, it is within the last two years that most of the interviewed organisations have scaled up the agile way of working throughout the organisation. All the interviewees expressed different challenges that they saw regarding quality when working scaled agile. Also, all of the interviewees were very interested to take part of the final result of the research. This seems to indicate that it is both very difficult to implement the agile way of working through an entire organisation and to ensure quality. Since all the interviews were semi-structured, each interviewee decided which area that was most interesting to discuss. In addition, the majority of the interviewees have the same role and responsibility within the different organisations. In the beginning of the research it was decided to interview people with an overall view of the organisation and knowledge about both the agile transformation and quality work. Afterwards, the researchers believe that in order to get a more objective picture of the challenges, interviews with other people with different roles and responsibilities within organisations would have strengthened the result. Furthermore, many of the interviewed organisations have a very extensive organisational structure. According to the agile framework the responsibility of quality should distributed to the teams, which also was confirmed by several of the interviewees. However, the researchers believe it was hard to understand who had the main responsibility of the quality. Many of the organisations have remained the same organisational structure, with a quality department, as it was before the agile transformation. The argument for this was mainly that the knowledge and competencies of the team members could not be taking for granted. The researchers are in the option that to have the same organisational structure just after the agile transformation is understandable. To be successful in the agile way of working, the researchers belive that the main responsible actor must be at team level. It can be difficult for the teams to feel truly responsible when there still is a quality department. However, many of the interviewees argue that they do not believe that the quality department will ever be removed, only it will be more supportive.

During the interviews, the researchers experienced that managers have a different view and understanding of quality. According to the literature, organisational leadership are the biggest barriers and enablers of successful agile transformations. Therefore, the agile transformation requires a new kind of leadership. All interviewed managers claimed that they are pushing down the responsibility regarding quality to the team level. The researchers got the impression that the responsibility had been transferred to the teams, which correspond with the agile practices. However, since only a few interviews with RTE and PO was executed and no one with software developers, the conclusion might be questioned. Have the teams been giving authority to really make more decisions? Or have they only received more duties? To get a better picture and input of how teams are responsible for quality, interviews with team members should be executed. According to the literature, the team members must value and trust each other in order to become successfully agile. The agile way of working requires empowered, self-organising and cross-functional teams. The researchers experienced that many teams have a lot of dependencies between each other. When a team is immature, the other teams are suffering because of the dependency. During the interviews, a common view was that teams are still immature while some teams that have been working agile for a longer time have come a long way in the agile journey.

The empirical findings also show that there is a high variety if suppliers, partners or customers working agile or not. The literature describes that it is of greatest importance to find customers that are willing to collaborate and actively be involved in the development process. The interviewees argue that close customer collaboration facilitates the agile way of working. Most interviewees said that when a customer or supplier is hundred percent involved, it leads to successful projects. However, it sets higher requests on cooperation between the parties. The researchers felt that sometimes there was too much dependency between the parties. The researchers found out that one organisation did not even know what their partners should deliver, which caused confusion among the teams and their work. Therefore, it is extremely important to inform and make the customers and partners understand how the agile way of working will affect them and what benefits that could be gained from it. Further, the researchers mean that it may not be optimal all the time to plan and work in short sprints with suppliers, partners and customer in cases where greater forward planning is needed or if ambiguity prevails.

6.1.2 Process related challenges

Regarding the process related challenges many of the interviewees describe the difficulties with breaking down traditional quality tools and practices into shorter sprints aligned with the agile way of working. According to Gallina and Nyberg (2015), the traditional quality tools are in many cases not suitable when implementing the agile practices. However, the researchers have not found any literature that explains how to modify the traditional quality tools or what tools to adapt instead. Further, standards, regulations and legal requirements should be broken down into smaller iteratives as well, but it seems that the interviewed organisations have not prioritised this yet. However, many interviewees have described that the standards, regulations and legal requirements contradicts to the agile way of working. This due to the heavy docummenations that are needed in order to follow the standards, requirements and legal requirements. The researchers believe more extensive studies within this area have to be done in order to investigate how organisations acutually could break down the standards, regulations and legal requirements into smaller sprints. In addition, the researchers express that many of interviewees might have a lack of knowledge regarding this topic. The researchers often had to explain the question in more detailed when the question was asked during the interviews and still, the majority of the answers were not that comprehensive from the interviewees.

Another challenge that was found in the empirical data was to derive which team that is responsible for software bugs when working on an upscaled level. For many organisations it is the teams themselves that are responsible for tracking the software bugs even on higher levels. However, many interviewees state that it is very difficult and time-consuming to find the root cause when defects arise. Unfortunately, the researchers have only found few studies on how to track software bug on an upscale level. Further, there was a common view by the interviewees that the way of working related to quality is not fully described in the agile methods or framework. It is overall very difficult to define quality and what “good quality” actually is. The literature also expresses that quality must be better described so that the agile methods could be applied throughout organisations. It should be clarified that there are many activities that describe how to work with quality. However, when scaling up there is a lack of descriptions regarding the quality aspects. The researchers agree, that even though there are frameworks for how to coordinate the agile work within large organisations, it is just frameworks and not ways of working.

Some of the organisations are facing challenges related on how to use the agile practices for product development when there are no final customers yet. It has also been mentioned that to succeed with the agile way of working, close interactions with the customer and fast feedback loops are needed. Moreover, it has been discussed how to work with proactive quality and many of the agile activities focusing on reactive quality. Thus, according to the literature the agile way of working was developed because of an increasing complexity due to software systems in a market of rapid changes. The researchers question whether the agile practices and principles actually have the purpose to include the proactive quality. The agile way of working seeks to work in short sprints and continuously change the requirements since the end-product, in many cases, are not known from the beginning. The researchers contemplate if there should be a proactive quality department that only focusing on the proactive quality and the teams should be responsible for the reactive quality. Further, many of the interviewees point out that they want to be able to utilise data that they collect from customers in order to proactively prevent mistakes. By utilise customer feedback, they believe that the quality will be improved. The researchers also believe that the quality will increase if organisations are using customer feedback. Constant feedback in order to respond to changes corresponds to the agile framework. Some organisations mean that they have access to customer feedback data but they do not use it. This might be because they are still in their agile journey and focus on the transformation and not prioritise to utilise the customer feedback. The researchers consider that as soon as an organisation become more comfortable in the agile way of working, they should focus on the data collected from the customer. The customer is important in order to improve both the product quality and the overall efficiency of an organisation.

The majority of the interviewed organisations have some kind of dependency between hardware and software development but have difficulties to find a collaboration model between them. Often hardware development uses the traditional waterfall methodology, which not corresponds to agility. During the interviews, organisations argue that these two must collaborate in order to achieve great product quality but they do not know how to achieve it. The researchers believe that hardware development might not have to become full agile, which is also discussed in the literature. Instead, the organisation should focus on finding new methods or modifying existing hardware methods. The researchers found that those organisations that have worked with the dependence between software and hardware for a long time have focused a lot on simulation. In this way, the challenges to deliver software in small increments could be solved and thereby be incorporated from day one.

6.1.3 Technology related challenges

The empirical findings indicate that relatively few organisations have structured agile metrics related to quality and that they have not changed or modified the metrics after the agile transformation. Many organisations want to find new common agile metrics for the entire organisation. This corresponds to the literature that also sees a demand for new software metrics after the agile invention. However, the researchers believe that organisations should create metrics based on the teams rather than for the entire organisation. Each team has different duties and experiences different issues related to them. However, the researcher considers that some metrics related to quality are important in order to see the development progress. During the interviews, the organisations argue that the quality has been improved after the agile transformation. However, the researchers find that organisations had difficulty in strengthening their arguments on how the quality has been improved. They could not by data show how much the quality has increased. Thus, it would have been interesting to see how much the quality has evolved through metrics.

Most of the interviewees stated that testing is by far the most important tool used to ensure quality, but testing must be more flexible and faster when becoming agile. The literature also supports that testing is an essential part in order to secure that the software has the right functionality and works in the intended way. However, during the interviews, it was discovered that few of the organisations had full test automatisation on an upscale level. All organisations argue that they are trying to achieve a higher level of test automatisation and phase out the manual tests. However, the researchers question the trust of automated tests.

Absolutely, it is believed that automated tests can lead to faster deliveries and better quality. Thus, it must be kept in mind that there is a person who creates the automated tests. Developers must therefore have enough knowledge of the product and how to build automated tests based on customer requirements. In addition, it is resource-intensive to implement automated tests and it could be difficult to reuse automate tests when adding new functionality. Additionally, no interviews with developers have been made to confirm the importance of automated tests. One organisation means that it is sometimes hard to use test automatisation since the tests only searching for defects due to customer requirements and not finding other defects that could be critical. Further, one interviewees argue that the reasons why they do not have automated tests on a higher level are lack of knowledge and understanding of the importance of automated tests from the management side. Considering that all interviewees have mentioned that test automation is the key to good quality, the researchers also believe that it is necessary. However, the organisations must consider the challenges related to test automatisation when implementing it.

6.2 Best practices

From the interviews, it was found that there was a great variance of different activities that were seen as quality enhancing for the organisations. All the interviewees agreed on that the agile way of working have increased the overall quality. Even those organisations that recently have implemented the agile way of working on an upscale level experienced increasing quality. Some interviewees had difficulties to mentioned just one agile activity that enhance the quality. They argued that all the activities together contributed to better quality.

When it comes to the most popular activity that was discovered from the empirical data, CI was stated by many interviewees as one of the most quality enhancing activites. CI contributes to that the quality can be secured in an earlier stage than before. Moreover, many organisations have highlighted that the most important activity is that the teams are feeling ownership over the quality. This corresponds to the literature that also argues the team should have the responsibility of the quality. However, the researchers experienced that the interviewees argued that the team should be responsible for the quality but, in practice, it did not always work out properly. Many pointed out that the teams are immature and therefore it is difficult to hand over the responsibilities to them.

Further, the majority of the interviewees argue that the PI-planning increases the quality. This activity contributes to increasing transparency, decisions are clearly made and different dependencies are emphasised. The researchers believe that the PI-planning generates better collaboration between teams since dependencies are discussed during this ceremony.

The interviewees describe that the retrospective ceremony is also one activity that enhances the quality because it compels everyone to continuously reflect upon the performance. This also corresponds to the literature and the researchers also consider that it is important to evaluate the performance. However, how the information from retrospective is shared to the rest of the organisation was not discussed during the interviews and could therefore be investigated further. Lastly, after the agile transformation, many interviewees mention that without doubt the best agile practice is close customer collaboration. The closer collaboration, the better product development. The researchers think that close customer collaboration is important to be able to make fast changes upon market requests.

Some of the interviewees could not say how much the quality has increased or which activity that was perceived as the best. Many organisations have not come as far within their agile journey and especially not when working scale agile. Several of the interviewees talked about activities that they believed were good from a quality point of view. The researchers experienced that some of the interviewees were very careful when answering and it seems that many of the best practices were not fully implemented yet.

7. Conclusion

In this chapter the conclusion of the research questions will be presented. For research question one, recommended actions on how to improve the identified challenges will also be presented.

RQ1: What are the challenges due to quality for organisations that work scaled agile?
Based on the empirical finding, several challenges have been discovered. The researchers have categorised the challenges into three main areas, namely: organisational related challenges, process related challenges and technology related challenges. Table 7.1 illustrates the three areas that are connected to challenges that organisations are facing related to quality when working scaled agile.

Table 7.1: The challenges regarding quality when working scaled agile.

Challenges	
Organisational related challenges	• Hard to change from traditional to agile way of working • Lack of leadership • Immature teams • Supplier, customer and paters depencies
Process related challenges	• Hard to break down quality tools into the agile way of working • The way of working related to quality are not described in the agile methods and frameworks • Hard to utilise feedback from customers • Hard to track software bugs on an upscaled level • Hard to break down standard, regulations and legal requirements • Hardware and software dependencies
Technology related challenges	• Lack of agile metrics related to quality • Lack of test automatisation

RQ2: Which solutions could be implemented to overcome these challenges?
The conclusion for research question one results in recommendations that will bring positive implication for organisations if they are implemented. The recommendations are divided in the same three categories as for the challenges, namely: organisational challenges, process related challenges and lastly, technology related challenges.

Recommendations for organisational challenges

- *Continuously evaluate the leadership*- The managers must have an agile mindset since they are the responsible actors that can change and improve the organisation. The managers need to spread their knowledgement to the rest of the organisation. Therefore, the managers should continuously evaluate themselves through employee surveys.

- *Invest in more education and coaching*- The manager must educate and coach the teams to ensure that enough knowledge about quality is created. When the teams have enough knowledge, managers can let go of the control and transfer the responsibility to the teams. Also, there will be no need for quality departments.

- *Create agile contracts*- The agile contracts focus on how the parties should work together and synchronise the work. It is important that from the beginning state what to expected from respective part. Also, good communication contributes to good cooperation between the parties.

Recommendations for process related challenges

- *Create a main track team that are responsible for software bugs on the highest level*- The main-track team takes over the responsibility for the code after the teams reach a certain test level. The main-track-team is responsible for finding the root cause and then handing over the software bug to the responsible team. By this solution the teams can focus on other value adding tasks instead.

- *Implement simulations*- Simulations will reduce the software and hardware dependencies. This in turns enables that the software development and testing phase can start earlier, even if the hardware is not physically available yet. The challenges to deliver software in small increments could be solved and thereby be incorporated from day one.

Recommendations for technology related challenges

- *Create metrics at team level*- Create agile metrics at the team level that are simple, objective and easily obtainable. Further, metrics should be replaced when they are no longer used. Try to avoid metrics for the entire organisation.

- *Create automated tests that can be reused*- By invest resources to create automated tests that can be reused, fast feedback can be obtained and more tests can be run in less time. Also, when adding new functionality to the system it will not require new creation of tests.

RQ3: What are the most important agile practices in order to ensure quality?

From the research, it can be concluded that it was a great variance of what organisations perceived to be the most important agile practices in order to ensure quality. In figure 7.1 the most common agile practices that were perceived to be the best from a quality point of view, found from the research are presented. Additionally, there was a joint agreement that education should be carried out for the employees in order to understand the agile way of working.

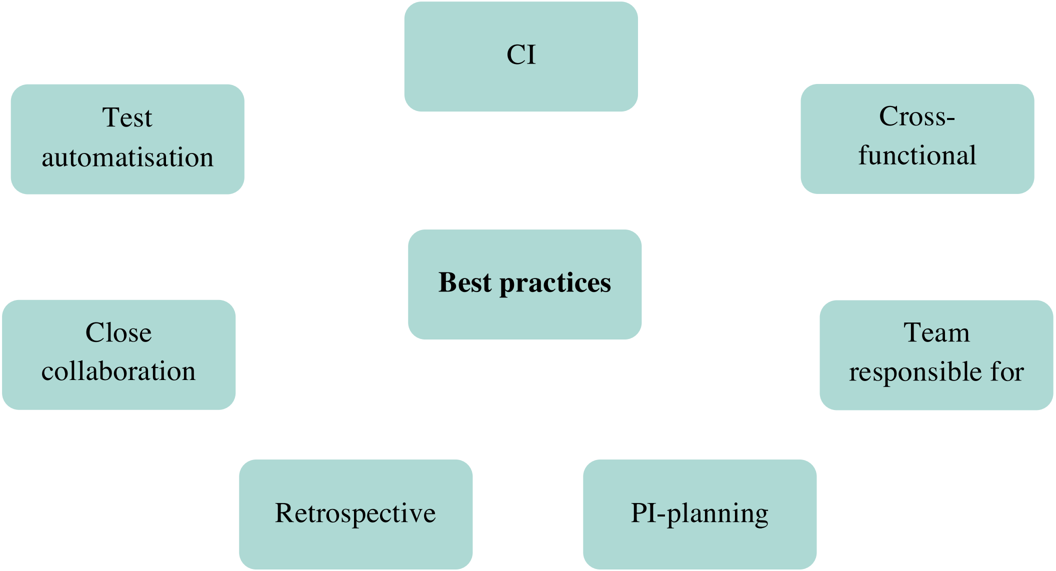


Figure 7.1: The best practices to ensure quality when working agile according to the interviewees.

- *CI*- The CI ceremony gives fast feedback and captures problems earlier than before. After a new function is approved, it should be integrated as soon as possible into the system, which will contribute to high quality for software development.
- *Cross-functional teams*- The cross-functionality ensure that a team have the right competencies and can be self-organised in their work.
- *Team responsible for quality*- When the team feel that they have the responsibility for quality that will be higher committed to their work.
- *PI-planning*- The PI-planning enhances the understanding of the dependencies between the different teams and increases the overall transparency within the organisation.
- *Retrospective*- Gives the team time to reflect upon the performed work and evaluate how to improve the work for the future development.
- *Close collaboration with customer*- Closer collaboration leads to that faster feedback will be received and the more flexible the organisation can be to changing requests.
- *Test automatisation*- When the tests are automated the quality can be secured faster.

8. Future research

The purpose of this master thesis was to investigate how different organisations are working with quality on a scaled agile level. The report answers the research question and addresses the challenges related to quality when working scaled agile. Moreover, the report describes the best experienced agile practices that were found from the interviews. However, the report only gives an overall picture of how organisations work with quality. Further research at a more detail level on how organisations ensure quality when working scaled agile must be investigated. The researcher addresses three different areas that could be of interest to investigate for further research:

- *How to break down traditional quality tools into an agile way of working.*
- *How to utilise feedback from the customer to make the organisations more efficent.*
- *How to break down standard, regulations and legal requirements into shorter iterations.*

In conclusion, the researchers hope that the report gives an insight into the challenges and best practices related to quality when working scaled agile. Further, the researchers also hope the report will encourage future research within the area.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods. Review and analysis. *VTT Publications 478.*

Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sarwar, S. Z. (2010). Agile software development: Impact on productivity and quality. *2010 IEEE International Conference on Management of Innovation & Technology*. doi:10.1109/icmit.2010.5492703.

Ahmed, J. (2018). *Using Six Sigma Tools In An Agile Software Project.* Retrieved from: https://www.isixsigma.com/tools-templates/using-six-sigma-tools-agile-software-project/

Ambler, S. W. (2010). Scaling agile: an executive guide. *IBM agility at scale.*

Appelo, J. (2011). *Management 3.0: leading Agile developers, developing Agile leaders*. Pearson Education.

Armani, F. (2014). *Scaling Lean Agile*. Retrieved from: http://www.slideshare.net/tangram77/scaling-lean-agile-mini-iad-2014

Bardhan, I.R., Krishnan, V. V., & Lin, S. (2007), *Project performance and the enabling role of information technology: An exploratory study on the role of alignment*, Manufacturing & Service Operations Management, Vol. 9, No 4, 579-595.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. & Kern, J. (2001). *Manifesto for agile software development*.

Beck, K., & Gamma, E. (2000). *Extreme programming explained: embrace change*. Addison-wesley professional.

Bergman, B., & Klefsjö, B. (2010). *Quality from customer needs to customer satisfaction*. Studentlitteratur AB.

Bhasin, S., 2012. *Quality assurance in agile – a study towards achieving excellence.* In: Agile India 2012, 17–19 February, pp. 64–67. Retrieved from: http://dx.doi.org/10.1109/AgileIndia.2012.18

Björklund, M., & Paulsson, U. (2014). Academic papers and theses. *To write and present and to act as an opponent. Translation: Christina Nilsson-Posada. Studentlitteratur AB, Lund.*

Boehm, B.W. & Turner, R., (2003). *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison Wesley.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, *22*(5), 30-39.

Bryman, A., & Bell, E. (2011). Ethics in business research. *Business Research Methods*.

Castka, P., B Ambler, C. J., Sharp, J. M., & Belohoubek, P. (2001). *Factors affecting successful implementation of high performance teams.* Team Performance Management: An International Journal, 7(7/8), 123-134.

Cooke, J. L. (2012). *Everything you want to know about Agile: how to get Agile results in a less-than-Agile organization*. IT Governance Ltd.

Cockburn, A., Highsmith, J. (2001). Agile software development: The people factor. *Computer* 34 (11), 131–133.

Cohen, D., Lindvall, M., & Costa, P. (2003). Agile software development. *DACS SOAR Report*, *11*, 2003.

Cprime. (2012). *Challenges of Adopting Agile in Combined Hardware and Software Environments*. Retrieved from: https://www.cprime.com/2012/08/challenges-of-adopting-agile-in-combined-hardware-and-software-environments/

De Smet, A., Lurie, M., and St. George, A. (2018). *Leading agile transformation capabilities leaders need to build 21st-century organization.* Retrieved from: https://www.mckinsey.com/business-functions/organization/our-insights/leading-agile-transformation-the-new-capabilities-leaders-need-to-build-21st-century-organizations

Dr. Winston W. Royce (1970). *Managing The Development Of Large Scale Software Systems*. Retrieved from: http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf

DSDM Consortium (2014). *DSDM Agile Project Framework - Handbook.* Ashford, Kent, UK: DSDM Consortium.

Flick, U. (2008). *Managing quality in qualitative research*. Sage.

Gallina, B., & Nyberg, M. (2015). *Reconciling the ISO 26262-compliant and the Agile Documentation Management in the Swedish Context.* In CARS 2015-Critical Automotive applications: Robustness & Safety.

Gillham, Bill, and Emily Jamison Gromark. *Forskningsintervjun: Tekniker Och genomförande*. Studentlitteratur, 2008.

Gren, L., Torkar, R., and Feldt, R. (2017). *Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies,* Journal of Systems and Software, vol. 124, pp. 104–119.

Grundström, C. (2013). *Kandidatexamensarbete: En skrivguide.* Linköpings universitet.

Gustavsson, T. (2014) *Agile: Konsten Att slutföra Projekt*. Liber.

Gustavsson, T. (2011). *Agil projektledning*. Sanoma utbildning.

Hammarberg, M., & Sunden, J. (2014). *Kanban in action*. Manning Publications Co.

Hox, J.J. & Boeije, H.R. (2005). *Data collection, Primary vs. Secondary.* Encyclopaedia of Social Measurement Vol. 1, 593-599.

Hultgren, E., & Lyhammar, D. (2017, 07, 27a). *Agila Distribuerade Team* [Audio podcast]. Retrieved from: https://agilpodden.podbean.com/page/4/

Hultgren, E., & Lyhammar, D. (2017, 12, 13b). *Agile Servant Leadership* [Audio podcast]. Retrieved from: https://podcasts.nu/avsnitt/agilpodden/22-agilt-servant-leadership

Hultgren, E., & Lyhammar, D. (2017, 04, 19c). *Agila KPIer* [Audio podcast]. Retrieved from: https://agilpodden.podbean.com/e/5-agila-kpier/

Hultgren, E., & Lyhammar, D. (2018, 08, 22). *Agila Kanban med Marcus Hammarberg* [Audio podcast]. Retrieved from: https://agilpodden.podbean.com/e/39-agila-kanban-med-marcus-hammarberg/

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004, September). *Software quality and agile methods.* In Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International (pp. 520-525). IEEE.

Jansing, D., Novillo, J., Cavallo, R., & Spetka, S. (2015). *Enhancing the Effectiveness of Software Test Automation* (Doctoral dissertation).

Johansson, C., & Heide, M. (2008). Speaking of change: three communication approaches in studies of organizational change. *Corporate communications: an international journal*, *13*(3), 288-305.

Joseph W. Yoder , Rebecca Wirfs-Brock , Hironori Washizaki, QA to AQ part four: shifting from quality assurance to agile quality: "*prioritizing qualities and making them visible*", Proceedings of the 22nd Conference on Pattern Languages of Programs, October 24-26, 2015, Pittsburgh, Pennsylvania.

Kalenda, M. (2017). *Scaling Agile Software Development in Large Organizations* (Doctoral dissertation, master's thesis, Faculty of Informatics, Masaryk Univ).

Kerzner, H. (2004). *Advanced project management: Best practices on implementation*. John Wiley & Sons.

Kormanski, C. (2005). *Change and conflict in group development*. Counseling and Human Development, 37(7), 1.

Kronfält. R. (2011). *Kvalitetssäkring i ett Scrumteam.* Retrieved from: http://sast.se/pdfs/Kvalitetssakring_i_ett_Scrumteam.pdf

Kupiainen, E., Mäntylä, M. V., & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies. *Information and Software Technology*, *62*, 143-163.

Kvale, S., Brinkman, S. (2009), *Den kvalitativa forskningsintervjun*, 2:a uppl, Lund, Studentlitteratur AB.

Lee, S., & Yong, H. S. (2010). Distributed agile: project management in a global environment. Empirical Software Engineering, 15(2), 204-217.

Liker, J. K. & Meier, D. (2006). *The Toyota way fieldbook: a practical guide for implementing Toyota's 4Ps*. New York, McGraw-Hill.

Lind, J.-I., & Skärvad, P.-H. (2004). *Nya team i organisationernas värld* (2:a upplagan uppl.). Malmö: Liber Ekonomi.

Mnkandla, E., & Dwolatzky, B. (2006). Defining agile software quality assurance. In *Software Engineering Advances, International Conference on* (pp. 36-36). IEEE.

Mukker, A. R., Singh, L., & Mishra, A. K. (2014). *Systematic review of metrics in software agile projects*. Compusoft, 3(2), 533.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. Communications of the ACM, 48(5), 72-78.

O'Conchuir, D. (2011). *Overview of the PMBOK Guide.* Springer.

Optimizely. (2018). *A/B Testing*. Retrived from: https://www.optimizely.com/optimization-glossary/ab-testing/

Paasivaara, M., Lassenius, C., & Heikkilä, V. T. (2012, September). Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work?. *In Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (pp. 235-238). ACM.

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development An Agile Toolkit.* Upper Saddle River: Addison Wesley.

Ronkainen, J., & Abrahamsson, P. (2003, May). Software development under stringent hardware constraints: Do agile methods have a chance?. In *International Conference on*

Extreme Programming and Agile Processes in Software Engineering (pp. 73-79). Springer, Berlin, Heidelberg.

SAFe Built-in Quality. (2018). *Built-in Quality.* Retrived from: https://www.scaledagileframework.com/built-in-quality/

SAFe Large Solution Level. (2018). *Large Solution Level*. Retrived from: https://www.scaledagileframework.com/large-solution-level/

SAFe Portfolio level. (2018). *SAFe Portfolio level*. Retrived from: https://www.scaledagileframework.com/portfolio-level/

SAFe Program Level. (2018). *Program Level*. Retrived from: https://www.scaledagileframework.com/program-level/

SAFe Servant Leadership. (2018). *Release Train Engineer and Solution Train Engineer.* Retrieved from: https://www.scaledagileframework.com/release-train-engineer-and-solution-train-engineer/

SAFe Team-level. (2018). *Team-level.* Retrived from: https://www.scaledagileframework.com/team-level/

SAFe Test First. (2018). *Test-first.* Retrieved from: www.scaledagileframework.com/test-first/

Sauro, J., & Lewis, J. R. (2016). *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann.

Scaled Agile Framework. (2018). *About SAFe*. Retrived from: https://www.scaledagileframework.com/about/

Schwaber, K., and Sutherland, J. (2013). The scrum guide. *The definitive guide to scrum: The rules of the game. Scrum. org, 268*.

Shiba, S., Graham, A. & Walden, D. (1993). *A new American TQM: four practical revolutions in management*, Productivity Press, Portland, Oregon.

Sommerville, Ian. (2007). *Software engineering*. Harlow, England ; New York: Addison-Wesley.

Solera, J. (2016). *Project Acceleration and Commitment-Based Project Management*. Retrived from: https://www.agilealliance.org/wp-content/uploads/2016/01/2009-NC-PMI-AE-Project-Acceleration-and-the-Story-of-Program-Catalyst-new-template-compressed.pdf

Sutherland, J., Schoonheim, G., Rustenburg, E., & Rijk, M. (2008, August). Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. In *Agile 2008 Conference* (pp. 339-344). IEEE.

Tevell, E., & Åhsberg, M. (2011). *Positive and Negative quality effects in Distributed Scrum Projects: An industrial case study.* Chalmers University of Technology, Gothenburg.

Tipnis, P. (2018). *Common Challenges Faced By First-Time Agile Organizations.* Retrieved from: https://www.knowledgehut.com/blog/agile/common-challenges-faced-by-first-time-agile-organizations

Ullah, M. I., & Zaidi, W. A. (2009). *Quality Assurance Activities in Agile: Philosophy to Practice*. School of computing Belkinge Institute of Technology, Blekinge.

Vaidya, A. (2014). Does dad know best, is it better to do less or just be safe? Adapting scaling agile practices into the enterprise. *PNSQC. ORG*, 1-18.

Wysocki, R. K. (2011). *Effective project management: traditional, agile, extreme*. John Wiley & Sons.

Youn, W., & Yi, B. J. (2014). Software and hardware certification of safety-critical avionic systems: A comparison study. *Computer Standards & Interfaces*, *36*(6), 889-898.

Zelkowitz, M. (2004). *Advances in Computers:* Advances in Software Engineering (Vol. 62). Elsevier.

Appendix I- Interview questions

English Version

Warm up questions

- What is your role in the organisation?
- For how long have you worked for the organisation?

1. Agile transformation

1.1 When did you start your agile journey and what were the main reasons for doing that?

2. Methods/framework

2.1 Which agile practices have you adapt? And why?

- How is the quality perspective described within these working practices?

2.2 Do you have any standards, regulations or legal requirements?

- How do you work to ensure these are followed and how are these adapted to the agile way of working?

3. Quality

3.1 Briefly, what is quality for you?

3.2 How is the responsibility due to quality organised today?

- How has responsibility/quality work/roles changed after you adapted the agile practices?

  If they have traditional quality organisation:

  - What should happen to dare to let go and give more control to the teams?

  If the teams are responsibility for the quality:

  - In what way are they responsibility?
  - What was required to dare in order to let go of the quality department?
  - How long did it take to do this?

3.3 Which tools do you use in order to ensure built-in-quality?

- How do you ensure the quality on a higher level than the team-level?
- What metrics do you use to assess quality? Product quality? Process quality?

3.4 What is the biggest improvement potential with quality assurance / control today when working agile?

3.5 How do you handle quality defects in the product, what are you doing on respective level?

3.6 Do you have any problem solving method to address problems?

4. “Best practice”

4.1 What are the most important quality enhancing activities in the agile framework you work according to?

4.2 Are there any agile activities that you are missing?

Ending interview questions

- If you had the opportunity to ask a question about quality to an organisation that is working agile, what would you have asked?
- Is there anything else that you would like to tell about quality and agile way of working that you think we did not ask?
- Do you have another person in mind that we could interview?
- Could we contact you for further questions or perhaps meet again for an interview if needed, when we have come further with our interviews and analysis?

Swedish Version

Uppvärmningsfrågor

- Vilken roll har du på företaget?
- Hur länge har du arbetat på företaget?

1. Agil transformation

1.1 När startade er agila resa och vad var den främsta anledningen att ni gick över till de agila arbetssättet?

2. Metod/ Ramverk

2.1 Vilka och varför agila arbetssätt har ni valt att jobba utifrån?

- Är kvalitets perspektivet beskrivet i dessa arbetssätt?

2.2 Har ni några standarder, regelverk eller lagkrav att förhålla er till?

- Hur arbetar ni med att säkra efterlevnad av dessa, och är det anpassat till ett agilt arbetssätt?

3. Kvalitet

3.1 Kort, vad är kvalitet för dig?

3.2 Hur ser ansvarsfördelningen för kvalitet ut idag?

- Finns det en specifik kvalitetsorganisation?
- Hur har kvalitetsarbetet/arbetsfördelningen förändrats sedan tidigare?

Om traditionell kvalitetsorganisation:

- Vad ska behöva hända för att ni ska våga släppa taget och ge mer kontroll till teamen?

Om team är ansvariga för kvalitet:

- På vilket sätt är de ansvariga?
- Och vad var det som krävdes för att våga släppa taget om kvalitets organisationen?
- Hur lång tid tog det att göra detta?

3.3 Vilka verktygsstöd använder ni för att säkerställa built-in-quality?

- Hur säkerställer ni kvaliteten på en högre nivå än team-nivå?
- Vilka mätningar (metrics) använder ni för att bedöma kvalitet? Produktkvalitet? Processkvalitet?

3.4 Vad är det största förbättringspotentialerna med kvalitetssäkring/ kontroll idag när ni arbetar agilt?
3.5 När ni hittar kvalitetsbrister i produkten, hur hanterar ni det i respektive nivå?
3.6 Har ni någon problemlösningsmetod för att angripa problem?

4. "Bästa praxisen"
4.1 Vilka är de viktigaste kvalitetshöjande aktiviteterna i det agila ramverk ni jobbar enligt?
4.2 Finns det någon aktivitet kopplad till kvalitet som du tycker saknas?

Slutfrågor

- Om du hade möjlighet att ställa en fråga om kvalitetsarbete i agila miljöer, vad hade du frågat då?
- Är det något vi missat att fråga som du gärna berättar om kopplat till kvalitet och agila arbetssätt?
- Har du tips på någon ytterligare person att intervjua?
- Går det bra att vi kontaktar dig om ytterligare frågor uppstår när vi kommit vidare i analysprocessen?

Appendix II- Reflections about quality when working scaled agile

At the end of each interview, the following question was asked "*If you had the opportunity to ask a question about quality to an organisation that is working agile, what would you have asked?*" The question was asked in order to find additional difficulties that organisations are facing related to quality when working scaled agile. Below are some of the questions that organisations want to have answers to:

- *How has the organisational culture changed and how does that change in such cases affect the quality?*
- *Automotive Spice requires a high degree of vulnerability, does this really improve quality?*
- *How do you follow up a team? Should you even follow up? Who is responsible for the quality?*
- *How do you handle backlog refinement and how do you address stakeholder views in these backlogs?*
- *If the team discover a process problem, do you have a system so that the problem reaches the right person?*
- *How do you think about security-critical issues? When do you dare to release a product on the market?*
- *How do you ensure incremental quality in small parts? How is quality assurance split up? How can FMEA be broken down so it lasts for several years?*
- *Do you have a documented problem-solving method to address problems?*
- *What metrics are measured and how do you control the business with them?*
- *How can you get feedback from the community? How do you proactively work with it?*

The quest for quality when working scaled agile

A case study of the challenges and the best practices related to quality within software development

VIKTORIA ANDRÉASSON
SOPHIA JACONELLI

Department of Technology Management and Economics
Division of Science, Technology and Society

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019