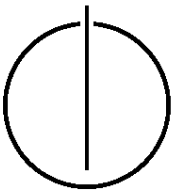
DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Information Systems

An Empirical Study of a Large-Scale Agile Transformation of an Online Retailer

Melisa Uluer



DEPARTMENT OF INFORMATICS

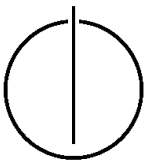
TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Information Systems

An Empirical Study of a Large-Scale Agile Transformation of an Online Retailer

Eine empirische Studie einer großen agilen Transformation eines Online-Händlers

Author:	Melisa Uluer
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Ömer Uludağ, M. Sc.
Submission Date:	March 16, 2020



I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, March 16, 2020 Melisa Uluer

Abstract

With the ongoing digital transformation organizations are confronted with new emerging challenges e.g., constantly changing customer demands or the constant emerge of new innovations. And with the success of applying agile methods and the possibility to overcome multiple challenges that arise due to the ongoing digital transformation more companies are increasingly adopting agile methods at scale. Primarily developed for the use of small teams, using agile methods became attractive due to the advantages that arise from the large-scale application of agile methods for larger projects in larger companies. However, adapting agile methods at scale also comes with challenges e.g., the company's overall resistance to change or the company's culture contradicting with agile values. As a result, the organization's processes, policies and culture have to reflect agile values and they need to develop an agile mindset if they aim to succeed in large-scale agile application. The goal of the organizations is therefore to achieve the capability to react quickly and with flexibility to technical innovations, new business possibilities and unforeseen environmental changes. To achieve this goal, they initiate a large-scale agile transformation which is defined in this thesis as the switch of an organization from a different development approach or work organization concept to the application of agile methods on an organizational level with at least two teams. While research on large-scale agile transformation mainly focuses on the challenges and success factors, there is a lack of empirical studies on adopting agile methods in large organizations as well as the lack of understanding agility in the enterprise context and the lack of focus on the impact of the large-scale agile transformation. This thesis aims to fill this existing gap in research by conducting a case study on a large-scale agile transformation of a German online retailer. This thesis provides a framework for organizations to determine their actual state of agility within the organization as well as their target state. By conducting a total of 17 interviews with people from different fields within the organization, the actual state regarding the large-scale agile transformation as well as the target state of the German online retailer could be determined by means of categories that enable the distinction of agile methods and plan-driven methods since they have contradicting emphases. Using the socio-technical systems theory, the impact of the large-scale agile transformation on actors, tasks, technologies, and structures were identified. In addition, a total of 7 challenges, 4 barriers, 6 success factors, and 8 lessons learned were identified.

Contents

List of Figures

List of Tables

Outline of the Thesis

CHAPTER 1: INTRODUCTION

The first chapter gives an overview about the motivation for this thesis and presents the objectives. In addition, an overview about the research approach is given, followed by a short outline.

CHAPTER 2: FOUNDATIONS

Chapter 2 defines all relevant and necessary terms.

CHAPTER 3: RELATED WORK

An overview of the process and impact of a large scale agile transformation based on literature is given in the first section of this chapter and followed by related work on challenges that can occur in a large scale agile transformation as well as identified success factors for a large scale agile transformation.

CHAPTER 4: CASE STUDY

This chapter contains the conducted case study and its results. First the case study is described and followed by the results of the case study structured according to the research questions.

CHAPTER 5: DISCUSSION

This chapter discusses the key findings of this thesis and describes the limitations.

CHAPTER 6: CONCLUSION

The last chapter summarizes all findings and proposes approaches for future work.

1. Introduction

This chapter gives an inside into the motivation in Section 1.1 of this thesis followed by the research objectives in Section 1.2 and the research approach in Section 1.3.

1.1. Motivation

Organizations are confronted with multiple challenges due the ongoing digital transformation. These include the rapidly changing customer demands, the increasing market dynamics, and the continuous emergence of novel advancement in IT as well as the adaption to these. Those challenges require organizations to rethink how they actually interact with customers, how they define value propositions, leverage data, and organize internal operations [22]. Additionally, existing competitors already develop new products and services using digital technologies, thus there is a need for reacting quickly to fast changing market demands [20].
One popular option to address those challenges is to introduce agile methods to the organization. Despite being criticized for the inadequate attention to design and architecture, the difficult application in larger projects, and the restriction to small, co-located teams, agile methods are widespread and popular in industry and companies are increasingly adopting agile methods to become more flexible and adaptive [1, 18]. A survey of VersionOne, published on May 2019, reported that 97% of the respondents' organizations practices agile methods [11]. Agile development methods promise not only a greater satisfaction of customers and a higher quality of products by fast delivery, continuous interaction, and transparency, but also enable a quicker reaction to changes in environments [17]. Additional reasons for adopting agile include increasing the productivity, reducing the project costs as well as the risks, and improving the team moral [11]. Even in non IT related environments there are shown benefits like better collaboration in the team, an increased customer interaction, increase in productivity, and in increase in speed [36].
Consequently, companies are adapting agile methods to their needs and not only are the methods applied by firms outside the IT and software industry but also employed beyond their traditional application areas e.g., IT related projects and software development. Initially developed for the use in small teams, using agile methods became because of the benefits attractive for larger projects in larger firms, emerging in the large-scale application of agile methods [14]. However, large-scale agile development methods also come with challenges [14, 21, 33]. These challenges consist e.g., of the company's overall resistance to change, the lack of skills or experience within a agile environment, and the

company's culture contradicting with agile values [11]. Consequently, in order to succeed at the large-scale application, the organization's processes, policies, and culture must all reflect agile values and they need to develop an agile mindset. Their goal is to become agile in the enterprise context which is defined as "the ability to sense an to respond swiftly and flexibly to technical changes, new business opportunities and unexpected environmental changes" [21]. The process to achieve this goal is called the large-scale agile transformation.
Research on large-scale agile transformation mainly focus on the challenges that the large-scale agile transformation entail, for example the general chance resistance, coordination challenges in multi-team environment, and hierarchical management and organizational boundaries, as well as success factors to carry out the agile transformation successfully, e.g., ensuring management support, training and coaching, and team autonomy. Based on this, a research gap can be identified existing of a lack of empirical studies on adopting agile methods in large organizations as well as the lack of understanding agility in the enterprise context and the lack of focus on the impact of the large-scale agile transformation. In this context, this thesis' objective is to address this research gap by conducting an empirical study at a large online retailer in Germany, focusing on the

- actual state
- impact on the organization
- emerged success factors and barriers
- lessons learned

of the agile transformation.

1.2. Research Objectives

This thesis aims to investigate a large-scale agile transformation of a German online retailer and defines therefore the following four research questions.

Research Question 1: What is the actual state regarding the large-scale agile transformation?

The objective of the first research question is to identify the actual state of the large-scale agile transformation by means of categories that were identified in literature and enable the comparison of organizations by means of these categories. These categories can be split up into two contradicting dimensions and allow the distinction between agile and plan-driven methods as they have different emphases. Furthermore, it is to find out the desired state regarding the large-scale agile transformation of the organization.

Research Question 2: Which impact has the large-scale agile transformation on the organization?

The second research questions focuses on the impact of the large-scale agile transformation in the organization by means of the socio-technical systems theory. This theory is used to analyze the changes induced by the agile transformation and to describe the impacts of the agile transformation on the four components, namely people, structure, task, and technologies.

Research Question 3: What are success factors and barriers of the large-scale agile transformation?

This question addresses the success factors, challenges, and barriers that could be identified so far by the organization through the large-scale agile transformation.

Research Question 4: What are the lessons learned of the large-scale agile transformation?

The last research questions addresses the lessons learned that could be identified by the organization through the agile transformation so far.

1.3. Research Approach

In order to achieve the in Section 1.2 mentioned objectives, a single case study has been considered appropriate in this thesis since it investigates a complex, real life issue where humans and their interactions are involved [37]. This case study is a holistic single-case study since this case study analyzes the agile transformation as a whole.
According to Yin [45] and Runeson and Höst [37], the case study research process consists of five steps. In the first step, the so called designing step, the objectives are defined and the case study is planned. In this context, the four research questions were defined.
The second step of the case study research process is the preparation for collecting evidence for the case study followed by the third step, to collect evidence. Interviews were considered as the single source of evidence for this case study since it enables the focus on the case study topics and gives personal explanations [45]. For the data collection method, a combination of quantitative and qualitative data collection was chosen since this ensures a greater understanding of what is being studied [37]. A total of 17 structured interviews were conducted of which were six employees from the software development department, four of the product department, three of the marketing department, one of the operation department as well as the VP of Engineering, the COO, and CEO.
In order to be able to answer the first research question, first a structured literature review was conducted. As suggested by Brocke et al. [43], the structured literature review consists of five phases. In phase one the review scope and research question were defined. In the next phase of the structured literature review, the key terms *Agile Transformation, Agile (Software) Development, Large-Scale Agile (Software) Development, and Large-Scale Agile Trans-*

formation were determined as relevant. Thereafter, these key terms were used in the search process phase. The databases Google Scholar, IEEE Explore, SpringerLink, and OPACplus of the technical university of Munich were considered for detecting relevant and qualitative literature. In the fourth phase a total of four articles were analyzed for categories. These categories are used to determine the actual status regarding the agile transformation and furthermore to determine the target state. The last phase of the structured literature review, creating a research agenda for further research, was not performed in this context. Figure 1.1 illustrates the structured literature review process for this thesis.

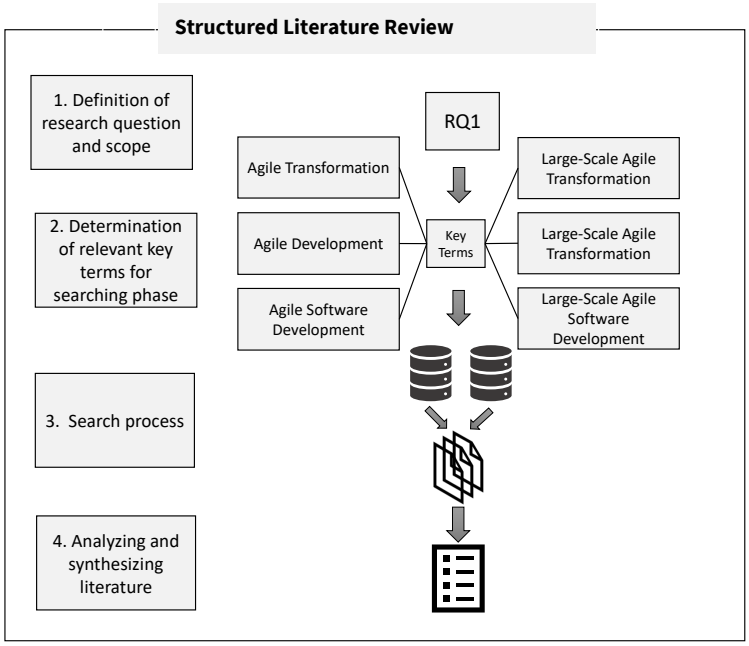


Figure 1.1.: Visualization of the structured literature review

To capture the impact of the agile transformation on the organization, the sociotechnical systems theory is used as a guideline by the preparation of the interview guide. The remaining research questions are addressed by open questions.
In the fourth step of the case study process, the quantitative data is analyzed by means of statistics whereas the qualitative data is analyzed by coding using a deductive approach as described by Cruzes and Dybå [13] and thereafter translated into categories. In this context, only codes occurring more than twice were considered as relevant and aggregated into one category. Figure 1.2 illustrates the process of the case study conducted in this thesis.

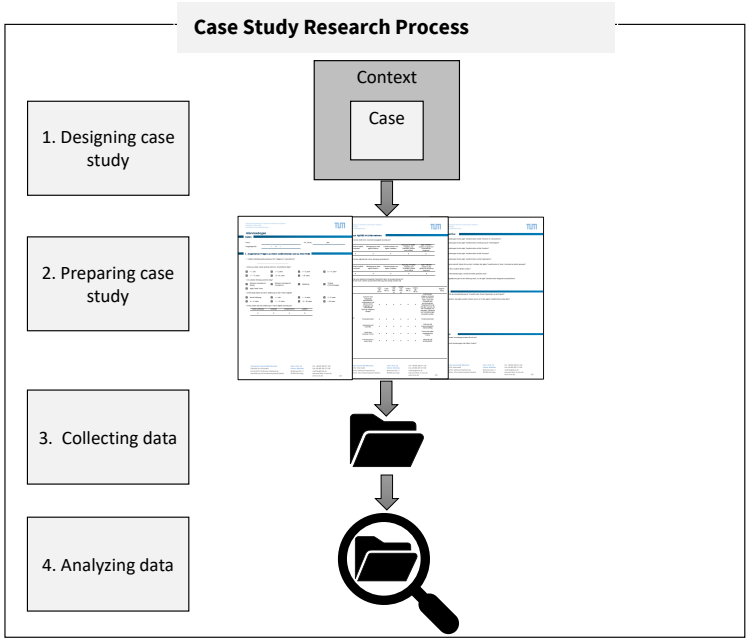


Figure 1.2.: Visualization of the case study research process

The remaining of this bachelor's thesis is structured as follows: In the second chapter, relevant foundations are explained, which are necessary to understand the basics of the work. Chapter 3 includes related work on the process and impact of a large-scale agile transformation as well as related work on challenges and success factors emerging through a large-scale agile transformation. The fourth chapter presents the case study at the German online retailer and answers the objectives of this work and thereafter discusses in the fifth chapter the research validity in the limitations section. The last chapter summarizes the findings of this thesis and presents possible future work in this research area.

2. Foundations

This chapter provides an overview of necessary foundations for this thesis. The first section explains the socio-technical systems theory which is chosen to serve as a guideline to examine the impact of the large-scale agile transformation on the organization. The second section provides a definition for lean development and the framework Kanban. This section is followed by the definition of agile development and the most used agile framework, Scrum. The last section provides a definition for the term large-scale agile transformation and explains the Spotify Model as well as the framework Scaled Agile Framework. The frameworks Scrum, Kanban, and Spotify were selected because the online retailer has used or is currently using them. In addition, the framework Scaled Agile Framework (SaFe) was chosen due to being one of the most used frameworks for scaling agile.

2.1. Socio-Technical Systems Theory

The Socio-Technical System (STS) is a theory that views a firm as an organizational work system [9] and was initially used to design or redesign an organizational work system [8]. This theory differentiates between "two *jointly* independent, but correlative *interacting* systems" [8] in an organization, namely the social system and the technical system [8]. The social system is composed of "the attributes of people, the relationships among people, reward systems, and authority structures" [8], whereas the technical system is composed of "the processes, tasks, and technology needed to transform inputs to outputs" [8]. Four of this terms are according to Leavitt [29] particularly significant - structure, actors, tasks, and technology [29]. Whereby structure and actors form the social system, and task and technology the technical system. These four components and their connections are shown in Figure 2.1 and explained in the following.
Structure is considered as systems of communication or systems of authority [29]. Actors represent people within the organization with their qualifications to carry out the work including their skills, knowledge, and relationships with others [29]. Tasks are defined as processes in order to develop products or services [29]. Technology stands for problem-solving inventions like work-measurement techniques or computers [29].
The arrows in the illustration point in both direction in order to demonstrate the interdependence's of these four components. Consequently, a change in any one of the four components is followed by a change in the other ones [29]. It is therefore considered preferable to improve the social and technical components together rather than just to improve one of the four components at the disadvantage of the others [31].

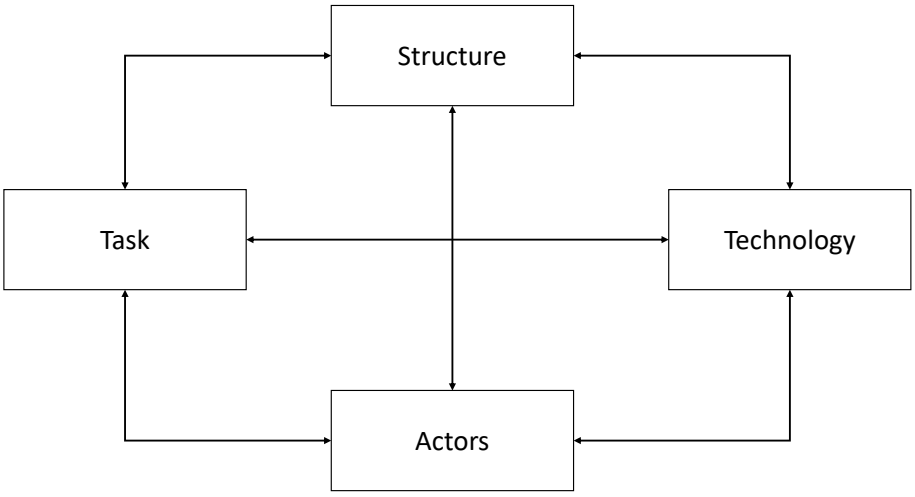


Figure 2.1.: Socio-technical model by [29]

The socio-technical systems theory can be considered as a guide to analyze mechanisms and outcomes in a work system that result with a change in the organizational work system [30]. Bostrom et al. [8] assume that the results of a work system are the result of correlations between the social and technical system. Consequently, it is necessary to deal with both systems [8].

According to this foundation, in order to scrutinize the large-scale agile transformation, the effects are analyzed by those four components. In this context, it will be possible to depict the impact of the agile transformation on the organization.

2.2. Lean Development

2.2.1. Definition

Lean development describes the application of the lean production concept for product development which has its origins in the Toyota Production System (TPS), where it was first used to reduce time and increase the production with the focus on eliminating "non-value-adding activities" [32].
The lean production concept strongly contrasts with craft or mass production [44]. In craft production, highly skilled employees use uncomplicated but adaptable [44] tools in order

to develop exactly what the customer asked for by doing one item at a time [44]. Custom made furniture or clothing are two examples of what craft production is. This however is too expensive for many people. Mass production employs skilled experts that design the product in order to be made by nearly unskilled workers using expensive machines with a single purpose resulting in a massive amount of standardized products. Because the change to a new product is associated with high costs, the standard design is maintained as long as possible [44]. Lean production on the other hand, is a combination of the advantages of both above described types while it eliminates the high costs of the craft production and the strictness of the mass production. Teams of multi-skilled workers at all levels of the organization are employed at this type of production who use highly flexible, automated machines in order to produce "volumes of products in enormous variety" [44]. In addition, lean production quests endlessly for perfection.
The definition of lean development can be summarized by the five lean concepts [32]:

- **Value** which is defined by the customer.
- **Value stream**. Every step in the process is stated on a map.
- The production process needs to **flow** continuously.
- **Pull**. Only products are built when they are requested by the customer.
- **Perfection** is striven for in the production by constantly identifying and eliminating waste.

These five principles can be supported by several tools, e.g., feedback, iterations, leadership, and testing [32].

2.2.2. Kanban

Kanban is a method for constantly improving service delivery using evolutionary improvement with the objective of a smooth and fast work flow. It is neither an agile software development method nor has his origins in software engineering but is an alternate approach to agility and even has been described as a method to improve the organizational agility and was adapted from the lean manufacturing approach. There are no predefined roles or processes in Kanban, the goal is to visualize the work and improve it continuously [32].
Kanban defines six core practices which are [32]:

- *Visualize the work, the work flow and business risks.* Each step needs to be visualized in the process. This is usually done on a Kanban Board, shown in Figure 2.2 with notes to display the work flow which are run trough from left to right.
- *Limit work-in-progress (WIP).* Goal is to keep the work which is currently in progress to a minimum in order to improve the work flow, reduce coordination costs, and increase the focus.

- *Make policies explicit.* All policies applicable to the process need to be documented.
- *Manage flow.* Monitoring, measuring, and analyzing work flows provide an illustration of the work flow, opportunities for improvement, and whether specific changes served to improve the process.
- *Implement feedback loops.* Feedback is encouraged in Kanban in order to learn about the process and the effects of changes.
- *Improve collaboratively, evolve experimentally.* Great importance is attached to the culture of continuous change and improvement.

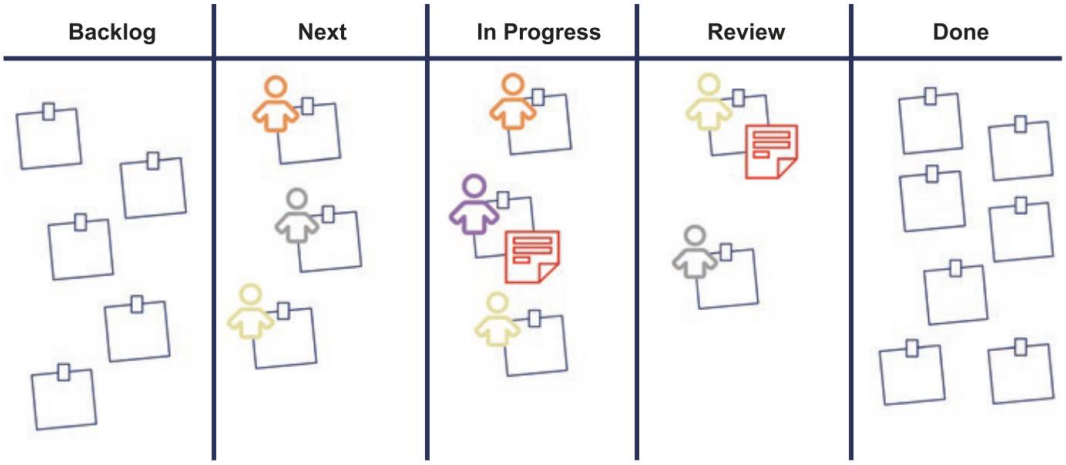


Figure 2.2.: Example of a Kanban board [38]

The introduction of Kanban can be divided into four steps [38]:

1. Splitting up and displaying work process into work tasks.
2. Creating a Kanban board with the columns "Product Backlog", "Next", "In Progress", "Review", and "Done".
3. Transferring previously created work tasks to notes or cards and assigning them to respective columns.
4. Prioritizing Product Backlog and transferring the tasks to the column "Next" by the Product Owner. Employees pick work tasks and work at them.

2.3. Agile Development

2.3.1. Definition

Alternatively to plan-driven software development methods e.g., waterfall model or spiral model, which focus on up front planning and processes [6] and are usually characterized by formal communication, a controlling management style, and defined tasks or activities [33], the agile methodologies are defined by the *Manifesto for Agile Software Development* [5]. *The Manifesto for Agile Software Development* is a set of four values and 12 principles which was published in 2001 by 17 authors and serves as a definition for agile as well as a basis for the development and deployment of agile frameworks [32][5]. It was initially intended to be used in software development but it's field of application has been extended by now [32]. The four values of the agile manifesto are named and described in the following:

- *Individuals and interactions over processes and tools*. Unmotivated employees who do not interact as a team will not deliver value to the customer, even if they are using the best processes and tools [32].
- *Working software over comprehensive documentation*. Agile development focuses on the delivery of working software and assures the customer thereby good process. Therefore, documentation in agile development is kept minimal and is extended with every delivery [32].
- *Customer collaboration over contract negotiation*. Contracts only contain the most necessary information, i.e. clarification of the collaboration, but not detailed documents that for example include all the requirement specifications. Value is placed on a close collaboration with the customer and on the understanding that a adequate product can only be developed with such a close collaboration [32].
- *Responding to change over following a plan*. In contrast to traditional development, upfront planning is not a part of the agile development. Plans are made mostly at a high or mid level and are adjusted in parallel with the development [32].

The above explained values on the left are consequently valued more than the values on the right.
Following on the values, 12 principles were worded in order to support those [4].

- The highest priority is the customer satisfaction through early and continuous delivery of valuable software.
- Changing requirements are welcomed even late in development. Agile processes make changes usable for the competitive advantage of the customer.
- Deliver frequently working software, from a few weeks to a few months, with a preference for the shorter time span.

- Business people and developers need to work together on a daily basis during the entire project.
- Set up projects around motivated people. Give them the environment and support they need and have confidence that they will get the work done.
- The face-to-face conversation is the most efficient and effective method of transmitting information to and within a development team.
- Progress is primarily measured by a working software.
- Agile processes encourage sustainable development. Sponsors, developers, and users should be capable of maintaining a steady speed for an indefinite period of time.
- Agility is enhanced by continuous attention to technical excellence and good design.
- Simplicity is essential since it is the art of maximizing the amount of work not done.
- The best architectures, requirements and designs are created by self-organizing teams.
- At regular intervals, the team thinks about how it can become more effective and then coordinates and adapts its behaviour accordingly.

Furthermore, agile is not about 'doing Agile', it is about 'being Agile' [32] and consequently having a agile mindset in the organization is essential for successfully applying agile methods. An agile mindset means that the organization or a person has internalized the agile tools, practices, principles, and values to a degree that agile is their standard way of working or interacting with the environment. Thus, introducing a agile method without internalizing the values will most likely lead to not being able to achieve the full potential of the agile method and therefore will not help to improve the organizations performance or to generally profit from the benefits of agile methods [38].

2.3.2. Scrum

Scrum is a framework to manage work on complex products and enables the employment of various processes and techniques and is defined as follows:

> "A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value" [40]

It was defined and used before the agile manifesto [42] and does not serve as a process, technique or definitive method [40]. Initially developed for managing and developing products, Scrum has become one of the most used agile frameworks by now and is used for products, services, and the management of a organization [11, 32]. The founder of Scrum, Ken Schwaber and Jeff Sutherland, are today maintaining a guide which contains

all information about the framework [40]. Scrum defines three artifacts, three roles, and five events which are illustrated in Figure 2.3. The objective of scrum is to divide a project into smaller parts and implement them in short iterations of two weeks, which are called Sprints [6].

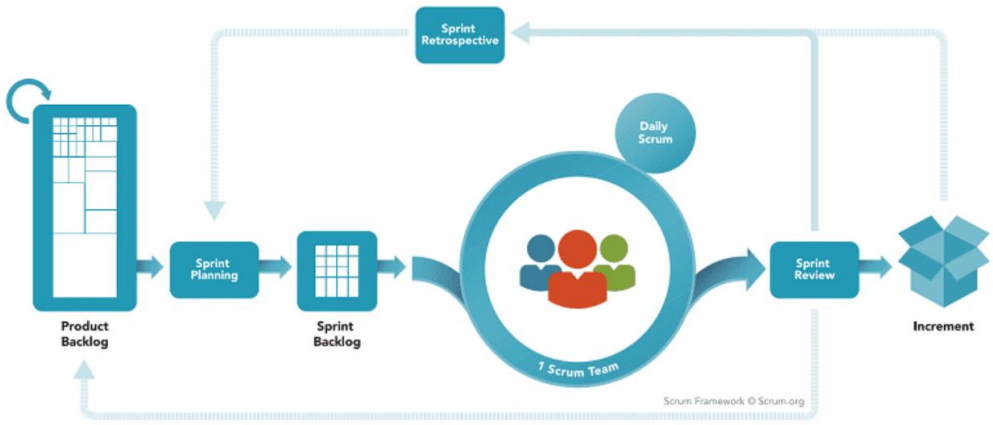


Figure 2.3.: Scrum model [3]

Scrum Artifacts are the Product Backlog, the Sprint Backlog, and the Increment.
The Product Backlog is an ordered list, consisting of all features, functions, requirements, enhancements, and fixes that are needed for the product to be developed. It serves as the single source of requirements. Items with higher order are normally clearer and more detailed than items that are of lower order [40].
The Sprint Backlog is a chosen number of Product Backlog items for one Sprint including a plan for realizing the Sprint goal as well for delivering the Product Increment. It is a highly visible, real-time picture of the work that the development team intends to do throughout the Sprint [40].
The Increment is the amount of Product Backlog items completely implemented during a Sprint including the value of the increments of all previous Sprints. The Increment must meet the "Definition of Done" and be in a usable condition [40].

The Scrum Team consists of the Development Team, a Product Owner, and a Scrum Master. Scrum has been designed for small, cross- functional, and self-organizing teams to ensure the high flexibility, creativity, productivity and adaptability of them.

The Development Team works on the product. They are self-organizing and cross-functional with all necessary skills in order to create the product. The optimal Development Team size is between three to nine members [40].
The Product Owner is one person and part of the Scrum Team and is responsible for maximizing the value of the product as well as for the product backlog including content, availability and ordering. The Product Owner decides whether to release the Increment or not [40].
The Scrum Master is one person and responsible for promoting and supporting Scrum. The Scrum Master helps everyone of the team understanding the theory, practices, rules, and values. He also helps those outside the team to identify helpful interactions with the Scrum Team and if necessary to change those interactions in order to maximize the created value by the Scrum Team [40].

Scrum uses prescribed events to create regularity and to minimize the need for additional meetings. All events have a maximum duration.
The Sprint is an iteration of one month or less during which an Increment is created. Each Sprint has the same duration during one development effort and starts immediately after the conclusion of the previous Sprint. Sprints contain the Sprint Planning, Daily Scrums, the Development Work, Sprint Review, and the Sprint Retrospective. A Sprint can be cancelled before the duration is over by the Product Owner if the Sprint Goal becomes obsolete [40].
The Sprint Planning is a plan which includes the work to be performed in one Sprint. It is created by the entire Scrum Team. It answers the questions, what to do in the upcoming Sprint and how to achieve the work needed to deliver the Increment [40].
The Daily Scrum is a daily 15 minutes meeting for the Development Team. At the Daily Scrum, the work for the next 24 hours are planned, the work of the last 24 hours are discussed, and the emerged impediments are mentioned in order to solve them as soon as possible [40].
The Sprint Review is the last meeting of a Sprint in order to inspect the Increment and adapt the Product Backlog. The Scrum Team as well as the stakeholders attend to this meeting. The Increment is presented and demonstrated by the Development Team [40].
The Sprint Retrospective takes place after the Sprint Review and before the next Sprint Planning and serves as an opportunity for the Scrum Team to inspect themselves and to create a plan for improvements for the following Sprints [40].

When comparing Kanban and Scrum, it is apparent that they are two opposites. For example, Kanban defines neither roles nor processes, which are clearly specified for Scrum. Additionally, products and services are delivered continuously in Kanban, whereas Scrum defines a period for development and delivery [32].

2.4. Large-Scale Agile Transformation

2.4.1. Definition

Originally designed for application in small, non critical projects with small teams, agile methods gained much prestige because of their claiming advantages [16]. There is no existing clear definition for the term *large-scale*, however Larman and Vodde [28] stated in their article about scaling agile development, based on their experience, that the term *large* refers to 800 people on average [28]. All of these people were working on one product while being spread over five locations. Also, Dingsøyr et al. [16] created a taxonomy of scale to define the term *large-scale* specifying that *large-scale* refers to a number of two to nine teams working on one common product [16]. In addition, *very large-scale* refers to over ten teams [16]. Dikert et al. [14] defined *large-scale* based on several studies as at least 50 people or more than five teams consisting of six to seven team members developing one product or working on one project together [14]. These studies demonstrate that the term *large-scale* is defined by the number of persons or teams working on one common product or project. Moreover, there are distinctions of agility made in an organization by Power [35]. In every context, "large" refers to the size of the organization. The differences are made in the context of having a team being agile in the organization, developing a product using agile methods with several teams, and the organization itself being agile [35]. The first one describes one team in a large organization using agile approaches and at the same time being autonomous and independent. This represents agile in a organization but differs from *large-scale* even if every team would work agile autonomously and independently [35]. Several teams developing one product together using agile approaches refers to the second differentiation, large-scale agile development. This type of agility is characterized by a higher effort in coordination and communication [35]. The last distinction refers to the agility of the organization itself which is defined as "an organization's ability not only to sense, but to respond swiftly and flexibly to technical changes, new business opportunities and unexpected environmental changes" [21].
In conclusion, this thesis uses the following definition of large-scale agile transformation:

> The switch of an organization from a different development approach or work organization concept to the application of agile methods on an organizational level with at least two teams.

To initiate such a large-scale agile transformation, frameworks like the Spotify Model or Scaled Agile Framework (SAFe) can be applied. Both frameworks are explained in the following.

2.4.2. Spotify Model

The Spotify Model is used for scaling agile and got his name from the company Spotify AB which is an online music streaming platform. Henrik Kniberg and Anders Ivarsson

published in 2012 a paper which describes the structure of Spotify existing of multiple teams in order to work successfully at a large-scale and extended this with two videos describing the culture at Spotify [24, 25, 26]. Figure 2.4. illustrates the described structure at Spotify.

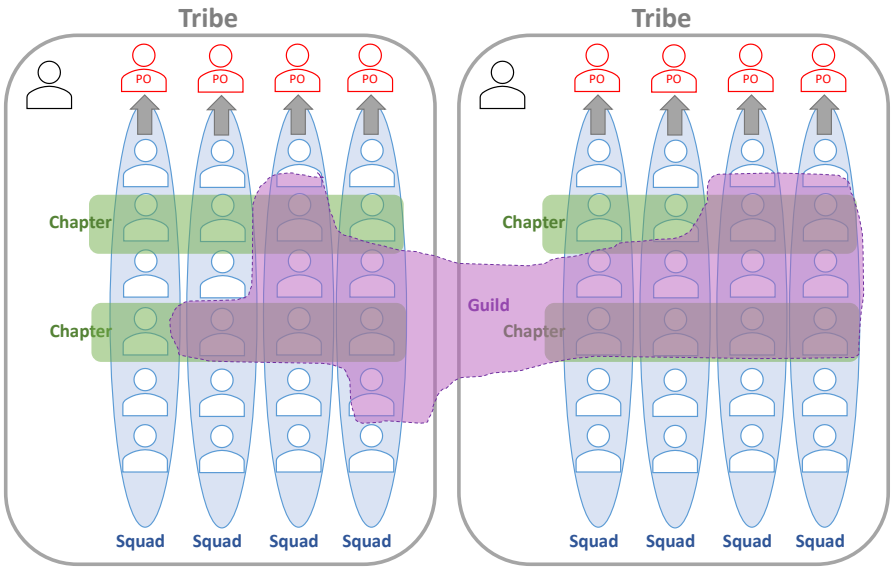


Figure 2.4.: Spotify model [26]

The model consists of four units:

- Squads
- Tribes
- Chapters
- Guilds

Squads have similarities with the Scrum Team. They work together, have all needed skills and tools for designing, developing, testing, and releasing the product, and are self-organizing. The decision on how to work is left to them, they can use Scrum, Kanban or a mixture of both. Each existing Squad has its own long-term goal to achieve and should be in direct contact with their stakeholders. They also do have the opportunity to get in contact with an Agile Coach whose mission is to help them and improve their way of working, similar to the Scrum Master. One Product Owner is responsible for one Squad and, similar to Scrum, for the prioritization of the work to be done. Squads are supposed

to spent 10% of their time on a so called hack day. This way, people are encouraged to try new things and get in touch with other team members [26]. Squads do not have a specific given technology stack, but technologies are recommended by Squads to each other [25].
A compilation of several Squads with goals in related areas is called **Tribes**. One lead is responsible for one Tribe and makes sure Squad members have the best possible space within their Tribe. Usually, Squads of one Tribe are co-located and one Tribe exist of less than 100 people. Squads of one Tribe get together and show the remaining Tribe what they delivered or on what they are currently working on. Dependencies between Squads are tried to eliminate and the goal is to be loosely coupled and tightly aligned [26]. If Squads need to work together due to a specific big project, they meet daily for synchronization, visualize their progress, and do a demo every 1-2 weeks in order to keep the dependencies to a minimum. In addition a small leadership group is needed to keep a eye on the big picture, a tech lead and product lead are useful for big projects [24, 26].
In order to ensure communication across Squads, Chapters and Guilds were created. **Chapters** exist of people with similar skills and competencies, and are in the same Tribe. They meet on a regular basis to exchange their challenges and expertise. For each Chapter, there is a Chapter Lead who has traditional responsibilities e.g., setting salaries. Additionally, the Chapter Lead is part of one Squad in order to make sure the lead stays in touch with reality. **Guilds** have a wider range compared to Chapters. They often reach across several Tribes or even the whole company. They are a collection of people who would like to share their knowledge, tools, and practices. Consequently, a Guild consists normally of all chapters working on the same area. Nonetheless, everyone can join. One Guild Coordinator is responsible for coordinating one Guild [26].
Their **culture** is based on agile principles and their corporate goal is to achieve high alignment and high autonomy. Since culture is about people, they focus on motivation, community, and trust rather than structure and control. Also, each failure is a learning for them. They use it as a long term strategy. If they fail fast, they will learn fast and consequently improve fast. Some Squads have a fail wall where they show their latest failings [24, 25].
Henrik Kniberg stated in one blog post, that the Spotify Model was not intended to serve as an framework to scale agile but rather an example of how Spotify works [23].

2.4.3. SAFe - Scaled Agile Framework

SAFe is one of the most used frameworks to scale agile approaches in organizations [11]. It is based on knowledge and provides practices, principles, and competencies to scale Lean, Agile, and DevOps [39].
It promises a quicker adaption and respond to competitive threats, identification and delivery of customer value, as well as maintenance of the quality [39]. In addition, 30% of the employees' engagement increases, productivity increases up to 35%, quality improves up to 35%, and the adaption of SAFe enables a 50% faster time-to-market [39]. SAFe defines seven competencies with which four different configurations of SAFe are possible. Figure 2.5 illustrates the Full Scale configuration. In the following, first all competencies

are explained and afterwards all four possible constellations of SAFe.

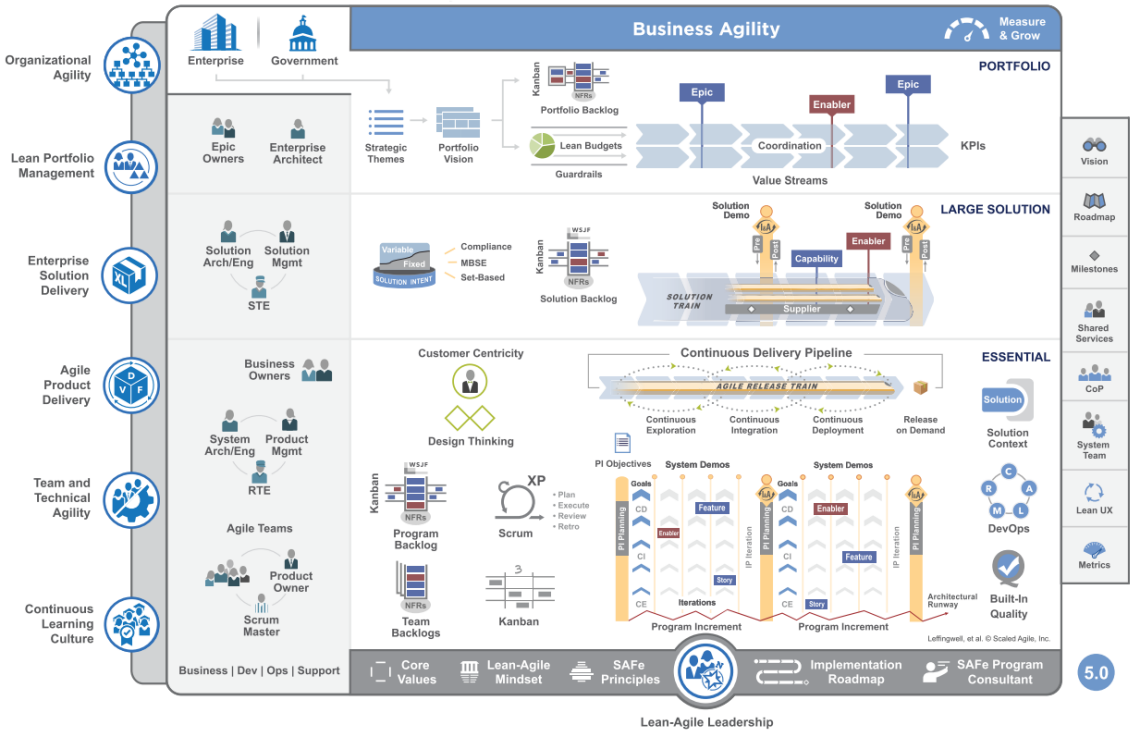


Figure 2.5.: Full Scale SAFe configuration [39]

Competencies

Lean-Agile Leadership. Since only leaders have the authority to change and improve the system, managers or leaders, should develop a lean-agile mindset, and adapted the four SAFe core values as well as the ten SAFe principles, in order to enable them to believe in people being able learning new behaviours which will therefore allow the change of norms throughout an organization. Also, people can learn lean and agile ways of thinking through leaders, for example by offering their guidance to help people increase their accountability, recognize and manage their emotions or those of others, and be self-motivated and therefore encourage those of others. Lastly, this transformation necessities an organizational change effort. Leaders' challenge is therefore to support this change by e.g., communicating the need for change in an inspiring and motivating way, and teaching the involved individuals the values, principles, and practices of lean and agile [39].

Continuous Learning Culture. This competence provides values and practices to encourage the entire organization to continuously increase the knowledge, competence, performance and innovation. For this, all employees have to keep learning and growing which encourages the ability of the organization to transform itself to gain a competitive advantage.

Also, having a innovative culture has the effect that employees are motivated and enabled for exploring ideas for improving existing products, trying new ideas for new products, as well as improving impediments [39].
Team and Technical Agility. Agile teams are described as the key for business agility. This competence includes the lean-agile skills, principles, and practices which are required by the team in order to create products. The agile teams, equal to the team in Scrum, apply agile principles and practices and have the needed skills to develop a product in short iterations. The team members are committed to one common goal. Since a broad range of expert skills can not be found in a single agile team, several agile teams need to collaborate, which is called the Agile Release Train (ART). This constellation develops one ore more solutions. To be able to respond to market changes and deliver value in the shortest time possible, agile teams follow quality standards and processes in order to enable a so-called collective ownership of e.g, artifacts or codes [39].
Agile Product Delivery. This competence is a customer oriented approach to define, build, and realize a continuous flow of products and services to customers and users. The so-called customer centricity is described as a mindset in which the customer plays the most important role as well as each decision made is demanding on the customer. This mindset allows among other things to focus on the customer, think and feel like them, and realize the customer's needs. The in Scrum known Sprints are called Cadence in SAFe. This is used to ensure that important events happen regularly and predicatively [39].
Enterprise Solution Delivery. This competence covers the use of lean-agile principles and practices to specify, develop, deploy, operate and enhance software applications, networks and systems. To do so, lean-agile practices need to be applied in order to coordinate all needed activities to create systems and afterwards decommission these. In addition, the coordination of trains and suppliers serves for alignment and guidance of the extended value streams towards a common business and technology missions. Lastly, to release minimally useful systems, a fast, economical, continuous supply pipeline is required. This enables organizations to achieve much earlier learning with less investment as well as even starting generating earnings sooner. Since the goal is to gain a competitive advancement and to get to the market before the competition, these systems have to be designed to support continuous deployment and release as required [39].
Lean Portfolio Management. This competence describes a modernized way of portfolio management. Through this competence, strategy and execution are sympathized by using lean, agile and systems thinking. This demands on the collaboration of Strategy & Investment Funding, Agile Portfolio Operations, and Lean Governance. The first aspect, consisting of enterprise executives, business owners, and enterprise architects, ensures that business targets are met by creating and maintaining the solutions needed for that goal. The Agile Portfolio Operations allows for operational excellence by coordination and supporting decentralized program execution. It requires collaboration of the Agile Program Management Office (APMO) or a Lean-Agile Center of Excellence (LACE), with engineers, scrum masters, and evolving technical disciplines. The latter, Lean Governance, consisting of APMO, enterprise architect, and business owner, is responsible for providing an overview

of e.g., spending, audit, and measurement [39].
Organizational Agility. Organizational agility is described as the ability of a organization to respond quickly to challenges and opportunities arising from rapidly changing markets. Thus, more flexibility and adaptability is required. Therefore, everyone involved in solution or product delivery, has to be trained in lean as well as agile methods including their principles and practices. In addition, teams applying these principles, need to improve the business processes in order to achieve lean business operation. In addition, when the organization is able to continuously examine and observe market changes it is able to reorganize agile teams and ARTs better in order to address new opportunities [39].

Configurations
Four configurations exist in order to implement SAFe which are explained in the following.
Essential SAFe is the simplest configuration and serves as a starting point for implementation. This layer includes all four core competencies Lean-Agile Leadership, Continuous Learning Culture, Team and Technical Agility, and Agile Product Delivery [39].
Large Solution includes in addition to the first layer the competence Enterprise Solution Delivery. This configuration focuses on coordinating multiple ARTs and suppliers as well as meeting compliance and regulatory standards [39].
The *Portfolio* configuration serves as the simplest way to achieve business agility. It includes in addition to the *Large Solution* the competence Lean Portfolio Management [39].
Full SAFe is the most comprehensive configuration and exists of all three configurations, more respectively all seven competencies [39].

3. Related Work

This chapter provides an overview about current research on large-scale agile transformation, its process, impact on the organization, challenges, and success factors, in order to justify the existing research gap and the lack of empirical studies. The first section gives an overview about the few research papers that focus on the agile transformation's process and the impact on the organization. Section 3.2 summarizes research on the identified challenges that occur within a organization adopting agile methods and practices, as well as success factors that will likely lead to a successful implementation of the agile methods and therefore to a successful large-scale agile transformation.

3.1. Related Work on the Process and Impact of a Large-Scale Agile Transformation

Fuchs et al. [19] - Becoming Agile in the Digital Transformation: The Process of a Large-Scale Agile Transformation
Fuchs et al. [19] refer to the lack of empirical research "on the process, challenges, and actions" [19] in the field of the large-scale agile transformation and conducted a multiple case study which examines the process of the large-scale agile transformation in two organizations and also to what extend challenges and action do shape the transformation process. They conducted a total of 16 interviews. The socio-technical systems theory was used as a guide to examine the process. First, they examined the large-scale agile transformation process of each organization. Subsequently, they could derive three agile phases from the large-scale agile transformation. Within each agile phase, first a radical change occurs which increases the organization's agility radically. The radical change is followed by a incremental change, here the organization's agility increases only minimally or not at all for a long time. This minimal to no increase in the organization's agility is followed by challenges which develop into barriers. These barriers can in turn be solved trough specific actions. Figure 3.1 illustrates the large-scale agile transformation process accordingly. Addressing the second research question, they specify that occurring challenges and taken actions during the large-scale agile transformation do have a great impact on the process of the large-scale agile transformation. These challenges and action result in a "episodic change". Additionally, they were able to identify three main barriers occurring in a large-scale agile transformation, namely "coordination of different organizational worlds", "difficult selection of the right people", and "suitability of agile methods" [19].

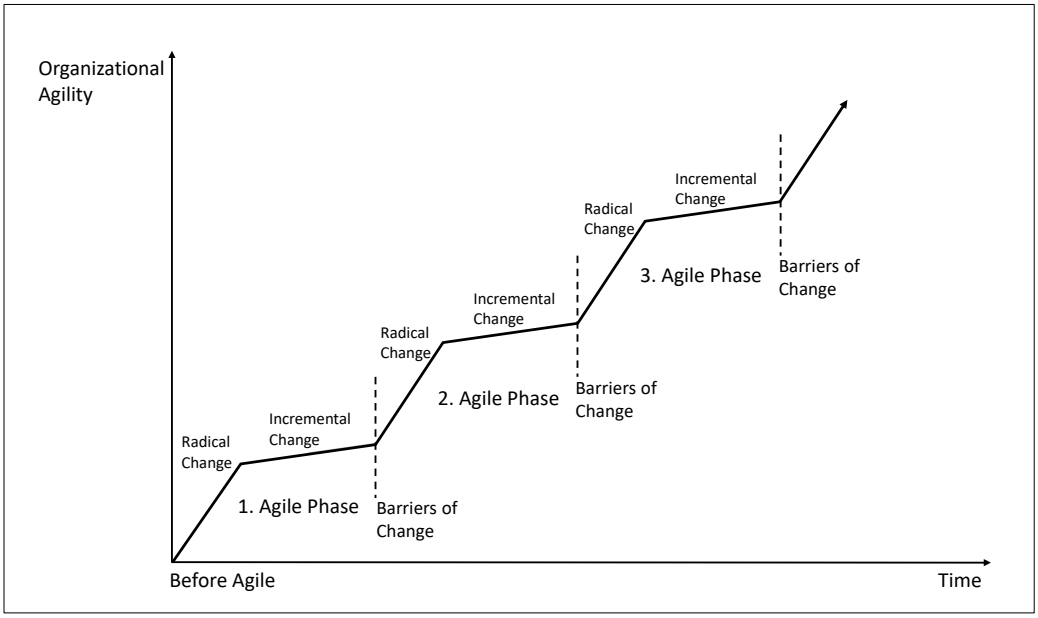


Figure 3.1.: Large-scale agile transformation process by [19]

Gerster et al. [20] - "Agile Meets Non-Agile": Implications of Adopting Agile Practices at Enterprises"

Gerster et al. [20] address the lack of empirical studies on the impacts of agile methods on a organization and the understanding of agility in the enterprise context. Therefore, this multiple case study focuses on the impacts and challenges that arise when companies adopt agile practices. In order to investigate the impacts, Gerster et al. (2018) [20] use the socio-technical systems theory as a guideline. They conducted the case study at ten companies each of different industries. The case study reveals areas that are affected by the adoption of agile practices. The areas are classified into the components of the socio-technical systems theory, namely actor, task, technology, and structure. The in the following described impacts represent only a section of all identified impacts in order to provide a better understanding.
Dimension "actor" is affected e.g., in the job-profiles and roles. In agile settings there are no more supervisors who tell people what has to be done, instead every employee needs so self-organize, in addition, new roles emerge, e.g., the agile coach or product owner. Also, it is necessary to change the culture of the organization according to agile principles for a successful agile transformation. The task dimension is split into two categories, namely products and processes. The product definition, in particular the question how

to design products, shifts from being project and input-oriented to product and output-oriented. Processes change in terms of budgeting and controlling. Because agile processes focus on short time delivery this contradicts with long-term planning, particularly budget planning. Most companies adopted a "product-oriented budget allocation" [20] as a solution. Furthermore, in order to be able to control and measure the outcomes, despite the fact that agile relies on self-organizing teams, which makes control more difficult, companies defined new metrics e.g., percentage of builds automated and time to deploy, and the percentage of reduced baseline defects. Technology affects the software development, IT-Architecture, and software tools. Agile software development is performed in short iterations and with continuous planning instead of up-front planning. Consequently, due to the short planning phase, it was pointed out to have a stable technology stack e.g., Jira for products definition, Eclipse for coding etc. Structure is affected in two ways. First, the fact that agile and non-agile units exist at the same time caused conflicts due to different goals, processes etc. especially when they needed to communicate with each other. Companies found ways to overcome this challenge by adding experts in teams. Second, that conflicts were caused because of resources between products. In order to solve the conflict how to ensure access to occasionally needed experts, companies assigned experts to related products as a solution [20].

Laanti et al. [27] - Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation

Laanti et al. [27] address the research gap that despite the increasing adaption of agile practices and methods, there are only a few studies that examine the large-scale agile transformation, and to provide evidence, that agile methods do have a positive impact on the user's opinion they conducted a case study at Nokia with over 1000 participants to investigate what effect the duration of the use of agile or traditional methods has on the opinion about agile methods. To examine the opinions, they made nine statements which each participant needed to rate from 1 being "totally disagree" to 7 being "totally agree" whereby 4 represented "neutral". The statements are "agile development increases the effectiveness of development", "agile development increases the quality of the product", "agile development increases the transparency of development", "agile development increases collaboration", "agile development makes work more fun", "agile development makes work more organized/planned", "agile development increases the autonomy of development teams", "agile development enables the earlier detection of bugs/errors/defects", "agile development makes work less hectic". In total, all statements were rated with a mean greater than 4, except the last statement "agile development makes work less hectic" which has a mean of 3.64. So it can be derived, that the opinions in general tend to be more positive than negative. In addition over 50% stated, they would not go back to the traditional way of working. Resulting of the examination of the relation of the duration of agile experience and the opinion towards agile methods, the more experience one has in agile methods, the more positive is their opinion towards agile methods. On

the contrary, it has been found that many years of experience with traditional, non-agile methods can lead to negative opinions in certain areas e.g., with increasing experience in non-agile methods, less people stated that agile development increases transparency in development. Furthermore, they investigated what challenges and benefits exist for each opinion. They were able to identify the top three challenges and benefits for each group. Challenges for the group supporting agile methods are "deployment of agile methods", "requirements management/iterative planning", "dependencies, co-operation, work distribution, subcontracting", whereas challenges for the "negative group" are "deployment of agile methods", "requirements management/ iterative planning", "resourcing/effort management". Identified benefits for the group rejecting agile methods are "visibility and transparency", "requirements management/ iterative planning", "productivity/focus/efficiency/ predictability" [27].

Paasivaara et al. [34] - Large-scale agile transformation at Ericsson: a case study

Paasivaara et al. [34] investigated the large-scale agile transformation at Ericsson. Ericsson decided to adopt agile methods and practices to develop a new product using them and profiting of their known benefits. The objective of this research was to find out why and how the agile transformation was initiated, what challenges occurred at this time and how the organization managed these challenges. They conducted in total 45 interviews. Resulting, the organization had three main reasons to adopt agile methods: they were not satisfied with their current working method, saw agile as a corporate strategy and simply the need to adapt agile, and lastly there was a need to speed up the development process. In terms of how the agile transformation did proceed, they could identify four phases, namely "knowledge transfer and component-based teams", "introducing agile", "finding common ground through value workshops", and "towards continuous integration and deployment" [34].
As a first step, teams were assembled which worked without any given process, neither agile nor traditional. After that the organization built component based teams - several teams responsible for one component. Each expert of this teams was grouped into a "virtual team" every time the development of a new feature began. This caused challenges i.e. team members of the virtual team would perform their own tasks instead of working as a team together or members saw the component teams as their team colleagues and not the members of the virtual team. In sum, this structure was not flexible enough and incapable to react quickly enough to market demands.
The second phase, the introduction of agile, can be split up into four additional phases. First, they created a pilot team with volunteers which worked well at the beginning. One volunteer was however member of a component team which caused problems due his absence. Management decided to abort the pilot team and decided to "start a full-scale agile roll out with cross-component, cross-functional teams" [34]. A total of ten teams were created distributed in three countries. After adjusting the setup due to insufficient knowledge in each team, the organization understood that they could not achieve "a full-scale agile roll

out with cross-component, cross-functional teams" because every component contained in the product required expert knowledge. As a solution, the organization built teams that were specialized in use cases which included several components. This way, it was not necessary that each team member has expert knowledge on all existing components.
After that a total of 15 teams existed. However, challenges emerged e.g., communication between teams did not take place at all, and there was no common organizational value to follow. They created five core values as a solution - One organization, Step-by-step, Customer collaboration, Passion to win, and Fun. To ensure that the values are not neglected, coaches helped the teams realize these values.
The last phase focused on the goal of continuous integration and deployment. For that, they built three new teams with enough knowledge focusing only on that goal and implemented two measures to achieve that goal. After that, they were able to to release six times a year and set the additional goal to deploy new features as soon as they are finished.
In this context, the authors emphasize that this phases shall not represent a guideline on how to conduct agile transformation, but only represent the phases of the large-scale agile transformation at Ericsson. In addition, they could identify a total of 14 challenges and, except for three challenges, their respective solution [34].

3.2. Related Work on Challenges and Success Factors in a Large-Scale Agile Transformation

Boehm et al. [7] - Management Challenges to Implementing Agile Processes in Traditional Development Organizations

The basis of this paper was the research review of the University of Southern California Center for Software Engineering, who identified approximately 40 barriers for the adaption of agile methods into traditional, plan-driven organizations. These barriers could be categorized in "nonproblems", "problems only in terms of scope or scale", and ""significant general issues needing resolution" [7]. Boehm et al. [7] focused on the latter two barriers and identified three main categories, namely development process conflicts, business process conflicts, and people conflicts, each consisting of three to four challenges, as well as suggestions how to address these challenges and eliminate them.
Development process conflicts are described as the "most obvious" [7] problems, which is the challenge of combing the new agile processes with the longstanding processes of the organization, consisting of variability, different life cycles, legacy systems, and requirements.
Secondly, the business process conflicts consist of the human resources, progress measurement, and process standard ratings.
They highlight that people conflicts are the "most critical in improving management" [7] and the most important to be aware of for adopting agile practices. These people conflicts consist of management attitudes, logistical issues, handling successful pilots, and change

management [7].

- Management attitudes: The project managers role in an agile setup is to coach the team members and support them as needed as well as to protect the team from external factors in order to prevent impediments. Migrating from a traditional point of view with given processes and a certain role for each employee can cause problems.
- Logistical issues: For a sufficient and successful adaption of agile methods and practices, team members need to be collocated.
- Handling successful pilots: Describing success factors to adapt agile methods can cause a negative effect on people. For example, to suggest firing a manager or splitting up the team may give the impression that new things could be dangerous for ones career.
- Change management: Adopting agile means that something new appears, which leads to the fact that opponents of the new quickly emerge. This complicates the work of the team members to a great extend.

Dikert et al. [14] - Challenges and success factors for large-scale agile transformations: A systematic literature review

Dikert et al. [14] could identify 35 challenges which can be divided into nine categories and also 29 success factors for adopting agile methods and practices which can be divided into eleven categories by reviewing a total of 52 papers with 90% of the chosen papers being "experience reports". The most outstanding challenge categories are "agile difficult to implement", "integrating non-development functions", "requirements engineering challenges", and "hierarchical management and organizational boundaries" [14].
Agile difficult to implement is the most mentioned challenge. This category consist of the misunderstanding of agile concepts, where the purpose of agile was not clear or misunderstood e.g., the purpose has been seen in faster product delivery only, documentation was not considered necessary, and teams presenting an unfinished product. Furthermore, there is missing guidelines in literature to implement such agile methods and practices which makes the adaption more difficult. Additionally, the poor adaption of agile will lead to challenges e.g., organizations tend to customize agile practices and methods to their needs, which likely results in leaving out defined processes e.g., in Scrum, which will lead to problems like not achieving the initial goal or developing a agile mindset. Some cases also show that people return to the familiar, old way of working after learning agile practices caused by stress and pressure. Lastly, implementing agile methods with high enthusiasm and leaders becoming agile fanatics, will lead to the problems, that this enthusiastic attitude will fade if benefits are not seen immediately and groups could quickly form that are either for or totally against agile [14].
Integrating non-development functions in this context refers to introducing agile practices and

methods to other organizational departments. The emerging challenges are that other departments e.g., marketing or user experience and design are often unwilling to change in general. In addition, adapting to the incremental delivery causes challenges especially in the user experience or design function because of their long term view. Human resources in organizations use rewarding practices, which is seen as being against the team mindset in agile setups. Lastly, launching products are made difficult with agile methods because e.g., marketing needs three months for the product launch preparation which is however not possible in a agile setup when the product is changing constantly [14].
Requirements engineering challenges are caused e.g., because of the non existing "high-level requirements management" in agile frameworks which is necessary in large development, especially when requirements are set by different stakeholders and developers are not able to reach each stakeholder. Another challenge is that the given requirements need to be refined by developers due to being not specific enough. Furthermore, creating a appropriate user story is hard for product managers. They usually hand them in big pieces which is followed with the problem that the development teams needs to spend additional time to divide the big pieces into appropriate user stories. Lastly, long term planing is difficult to make due to the typical agile backlog giving only short term goals [14].
Hierarchical management and organizational boundaries consist of the challenges that the role of the middle manager in agile is unclear, the management still thinks in waterfall model, bureaucracy is still kept, and that the internal groups of people kept the same. The middle managers' role in agile is not to command and control anymore, but to encourage the team members for self-organization. Cases showed that managers who became Scrum Masters gave the developers the impression that their were still managed. Many middle managers have difficulties shifting their mindset to agile which has impact even after adopting agile practices. They e.g., still attach great importance to up-front project planning and reports on cost and progress. This way, agile team members were asked to produce heavy documentation and to follow processes. So, the actual bureaucracy of the old way of working is still carried out while the middle management is not willing to work in agile. Lastly, reorganizing teams permanently because of too few experts will make the planning difficult for teams [14].
Identified success factors categories that were the most outstanding are management support, choosing and customizing the agile approach, training and coaching, and mindset and alignment.
Management support during the agile transformation is described being crucial. Their support needs to be ensured as they are seen as a "key role" for making changes and have the authority and power to solve problems. In some cases, managers could bring opponents to accept the agile practices. Also, shown management support, like organizing workshops or visiting sprints will lead to higher motivation of the employees. But in order to gain the support of the management, the management needs to be educated on agile first. Uneducated managers on agile will lead to the fact that they are not willing to let go of their command and control way of managing which will not lead to self-organizing teams [14].
Choosing and customizing the agile approach is required in the agile transformation. Agile

approaches should be customized to the organization an its need rather then following guidelines. In addition committing to one approach will make comparing work between teams, relocating people, and predicting the process easier. Also, helpful was to map the agile practices to the old way of working step by step instead of getting rid of the old way first. Lastly, a success factor is to keep the organization and the processes simple. For example by simplifying the organizational chart and focusing on engaging the teams rather than on processes and tools [14].
Training and coaching everyone on agile methods and practices in terms of knowledge, mindset and attitude towards the new way of working, improves the changes of succeeding in the agile transformation and prevent from failing. Coaching teams is described as essential in succeeding at the transformation and can lead to failure if not. Coaching teams can also solve experienced problems within the team and help the team understanding agile practices [14].
Mindset and alignment, in particular agile mindset and alignment, is necessary for adopting agile practices. To succeed at developing a agile mindset, coaches or scrum masters should focus on the agile principles and values rather then agile practices. A common mistake of coaches or scrum masters is that they try to implement the practices first rather then internalizing the values. Organizing social events in the organization will help building a agile mindset as well. People having fun at the events will connect agile working with those events, which will make people more motivated in applying agile practices or social events will help increase the team bonding. Furthermore, a common understanding and commitment to the agile transformation must exist trough out the organization in order to succeed at it. Lastly, the influence of agile communities has a positive impact on the agile transformation [14].

Campanelli et al. [10] - Assessing Agile Transformation Success Factors

Campanelli et al. [10] could identify a total of 23 success factors, which can be divided into six groups, namely customer, management, organization, process, team, and tools by conducting a single case study at a software development company. They ranked the success factors by their difficulty to implement by executing the Rasch algorithm using a specific software. Accordingly, measurement model, training, agile champions, and new mindset/role are the hardest success factors to implement, whereas customer involvement, self-organizing teams, changes in mindset of project managers, and cultural changes are the easiest to implement or are already implemented. Each success factor is not explained in detail in this case study. The assignment of the success factors to the categories is not considered in this context [10].

Abrar et al. [2] - Motivators for Large-Scale Agile Adoption From Management Perspective: A Systematic Literature Review

Abrar et al. [2] were able to identify a total of 21 success factors from the management per-

spective and the respective factor that indicates the frequency how often it was mentioned in the papers by conducting a literature review including 58 research papers. Accordingly, "leadership strong commitment and team autonomy", "cooperative organizational culture", "team competency agile development expertise", and "training and learning and briefing of top management on agile" are the most mentioned success factors. The success factors are not explained in detail in this paper [2].

4. Case Study

This chapter presents the conducted case study and includes the results of it. In Section 4.1 the data collection as well as the respondents profiles are presented. Subsequently, in Section 4.2 the online retailer and additional information of their large-scale agile transformation are described based on the participants statements. Section 4.3 presents the derived results of the the conducted case study in this thesis starting with the analysis of the actual state and target state regarding the large-scale agile transformation followed by the impacts of the large-scale agile transformation on the organization. Thereafter, the results of the challenges, barriers, success factor, and lessons learned are presented.

4.1. Data Collection

In order to investigate the large scale agile transformation, data was collected by means of a total of 17 semi-structured interviews using a combination of qualitative data collection and quantitative data collection. In total there were six employees from the software development department, four of the product department, three of the marketing department as well as one agile coach, the VP of Engineering, the COO, and CEO. The interviews proceeded as follows. First, every participant was asked personal questions e.g., role description, experience with agile which took 5 minutes on average. Thereafter, questions about the agility in the organization were asked e.g., the maturity of the business agility and the classification of the categories which took 30 minutes on average. This part was followed by the retrospective where the impact on the organization, challenges, barriers, success factors, and lessons learned were asked. This part also took 30 minutes on average. This was followed by the outlook and the feedback which took 5 minutes each. The interview questions are shown on Appendix A.1. The agile coach and the COO were asked additional questions to receive additional insights about the agile transformation (see Appendix A.2.). The remaining interviewees were not asked due to insufficient knowledge about the background of the large-scale agile transformation. Table 4.1 gives an overview about all conducted interviews.

Respondents profile
35% of the participants report being employed in the company for 3 to 5 years followed by 29% being employed for 6 to 10 years. 24% state their years of company affiliation being between 1 to 2 years and 12% of being more than 15 years. Figure 4.1 summarizes the distribution of the years of company affiliation. The most common field of the respondents

is "Software Development" with a percentage of 35%, whereas the second most common field is "Product" with 23%. Marketing and the top level management represent 18% each. Figure 4.2 summarizes the distribution of the fields of the participants. 47% of the participants report their years of experience with agile methods being between 3 to 5 years, followed by 23% reporting between 1 to 2 years of experience with agile methods. 18% state their experience being between 6 to 10 years. "no experience" and "11 - 15 years" are reported with 6% each. Figure 4.3 summarizes the distribution of the years of experience with agile methods. Lastly, 65% of the participants estimate their degree of experience as "Advanced" whereas 29% report their degree as "Beginner" and 6% state that they have no experience. The degree of experience is summarized in Figure 4.4.

Id	Role	Duration in hh:mm	Date
I1	Agile Coach	01:37	19.11.2019
I2	Software Developer	01:51	03.12.2019
I3	Team Manager Software Development	00:35	03.12.2019
I4	Software Developer	00:46	04.12.2019
I5	Senior UX Designer and Researcher	00:35	04.12.2019
I6	Product Owner	00:55	06.12.2019
I7	Software Developer	02:47	13.01.2020
I8	Software Developer	01:28	13.01.2020
I9	Senior Art Director	01:00	16.01.2020
I10	Online Campagne Manager	00:54	16.01.2020
I11	COO	01:54	17.01.2020
I12	Teamlead User Acquisition	01:05	16.01.2020
I13	Teamlead Campagne and Content	00:58	17.01.2020
I14	VP Engineering	01:04	21.01.2020
I15	Head of IT	01:29	21.01.2020
I16	Webbusiness Analyst	01:40	13.02.2020
I17	CEO	00:55	17.02.2020

Table 4.1.: Overview of conducted interviews

4.2. Case Description

The organization decided in 2013 to introduce agile methods. At this time, they have opened far too many projects, have done too much and consequently did not have the right focus anymore. They recognized that the long-term project planning phases were not working properly and decided to introduce agile, which was at this time felt as very exciting. Their objective was to increase their productivity, to focus and simply to move faster and more efficiently. Without setting up a plan, the COO and a new employee with already existing practical experience in agile, started introducing the agile method Scrum.

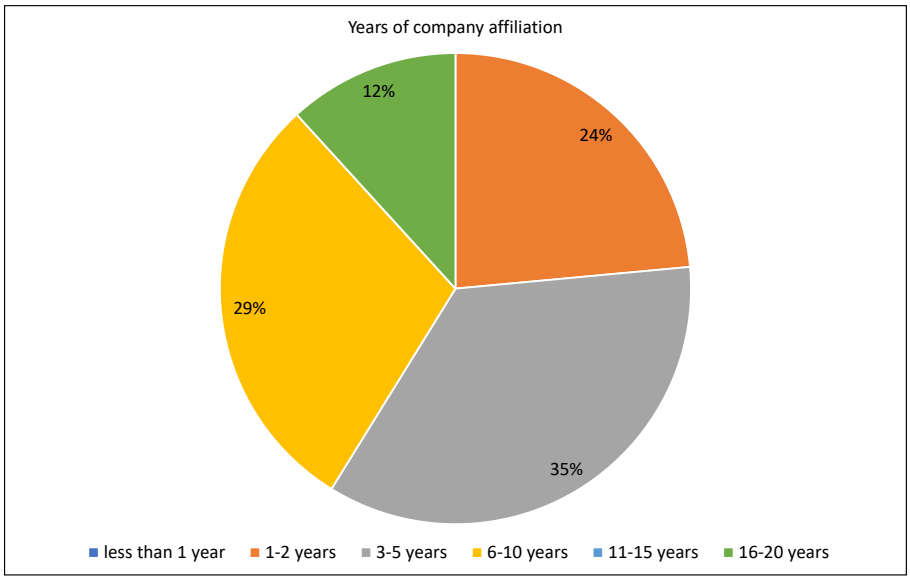


Figure 4.1.: Distribution of the years of company affiliation (n=17)

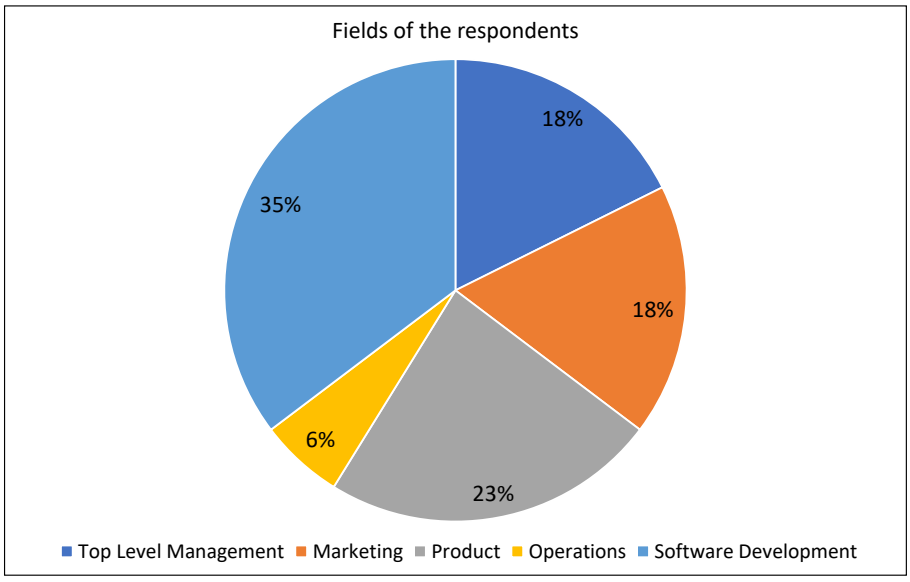


Figure 4.2.: Distribution of the fields of the respondents (n=17)

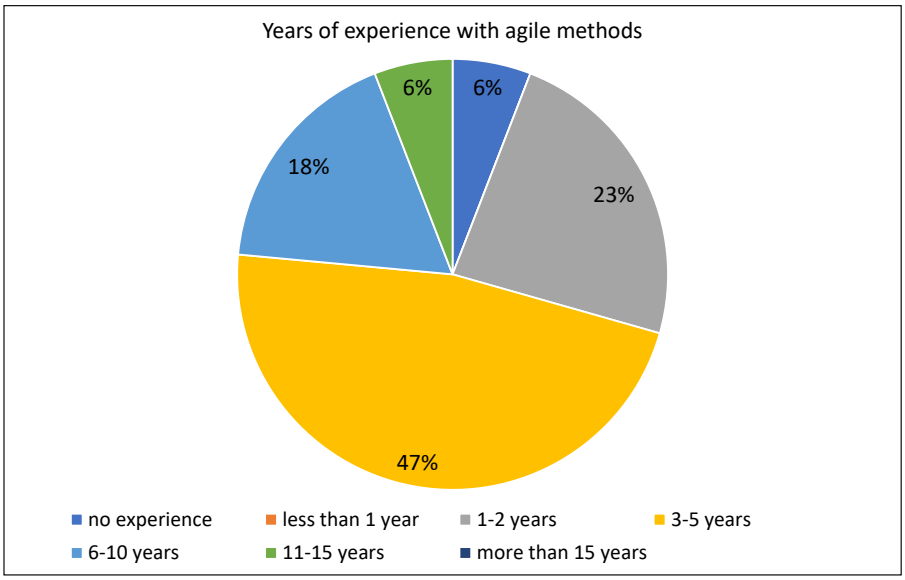


Figure 4.3.: Distribution of the years of experience with agile methods (n=17)

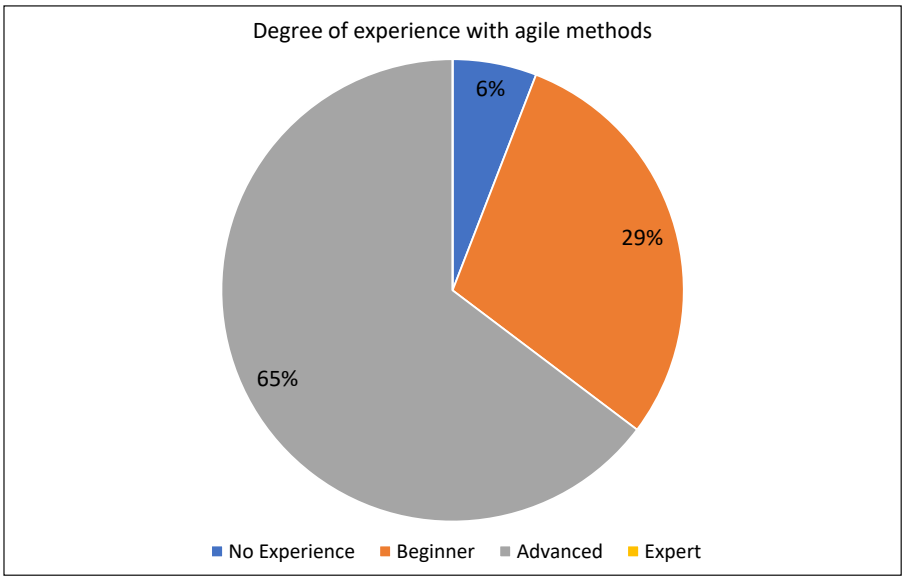


Figure 4.4.: Distribution of the degree of experience with agile methods (n=17)

They had an extra room they called Scrum where the management had isolated themselves regularly with the goal to develop a new digital product as fast as possible without involving the employees. The COO called their approach retrospectively "an electrifying personal ego thing". At this time, traditional project management structures were still existing in the organization which caused a discrepancy between management and employees.
The Agile Coach and VP of Engineering stated that they actually started introducing agile methods in 2015. The objectives were similar to the former ones: to react faster, to increase the flexibility, to have a customer-oriented product development process, to increase the intrinsic motivation of the employees, and to move away from the traditional project management process which was described as "this eternally lengthy planning with milestones and nothing is ever finished and in the end everything looks completely different anyway" by the agile coach. This time, the traditional project management team and the software development team initiated the introduction of agile methods with support from the top-management. Again they decided to introduce Scrum but with a different proceeding. They held workshops, formed appropriate teams and defined common goals to achieve, and were able to successfully develop a new digital product within two and a half years.
In addition, they conducted a short-term experiment in which they formed one Squad inspired by the Spotify Model with the goal to increase the conversion rates. They aborted this experiment due to major disagreements within the company.
Currently, they are migrating to the technological platform of their parent company and only have the software development teams working according to Kanban.

4.3. Results

The following section presents the results of the case study structured according to the research questions of this thesis.

4.3.1. Actual State and Target State regarding the Large-Scale Agile Transformation

This subsection presents the results on the analysis regarding the actual state of the large-scale agile transformation in the organization as well as the target state.
First, the agility maturity in the company was generally classified according to the statements "No initiative to achieve agility", "Considering an agile initiative", "Experimenting with agile practices", "Use of agile practices and techniques, but still maturing", and "Agile practices are used throughout the company" which are named in the following accordingly after the numbers 1-5. One person each voted for statement 1 and 2 (6% each), whereas 2 people voted for statement 3 (12%). A total of 13 people voted for statement 4, which is 76% of all participants. No one of the participants voted for statement 5. Consequently, the overall maturity of business agility in the organization can be defined as the use of

agile practices and techniques, but still maturing. Figure 4.5 gives an overview about the distribution of the maturity of business agility.

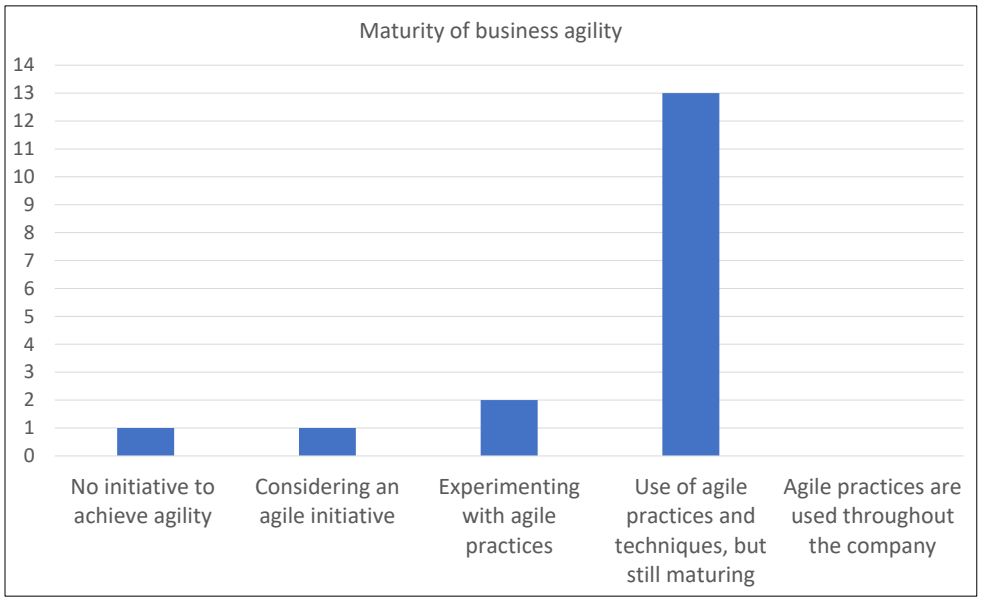


Figure 4.5.: Distribution of the maturity of business agility (n=17)

Thereafter every interviewee classified the agility maturity of their own department accordingly. CEO, COO, and the VP of Engineering were not considered in this part.
Every assessment of business agility was first examined by department by means of their mean (M) and standard deviation (SD). For this, each statement was given a number from 1 to 5 starting with "No initiative to achieve agility"=1 etc. Software development shop (M=4; SD=1) as well as software development production (M=3,667; SD=1,15) consider their agility maturity as "Use of agile practices and techniques, but still maturing". The product department shows the widest standard deviation and classifies their maturity of agility as "Experimenting with agile practices" (M=3,25; SD=1,5). The marketing department has the lowest agility maturity and classifies themselves as "No initiative to achieve agility" (M=1,33; SD=0,58). The agile coach, being in the operations department, claims that "Agile practices are used throughout the department" (M=5; SD=0). Figure 4.6 shows the comparison of the maturity of department agility of each department interviewed by their mean and standard deviation. It can be clearly seen here that departments like software development shop and production already use agile practices, whereas departments like marketing or a part of the product department, do not use agile practices or even do not consider using them.
This separation becomes also clear when looking at the absolute frequency of the selected

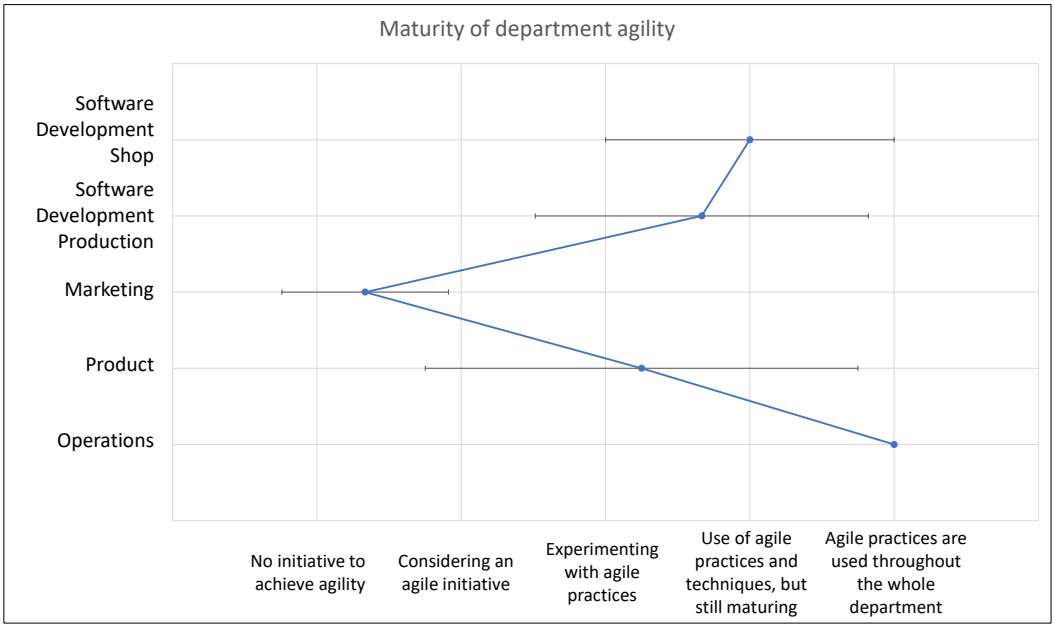


Figure 4.6.: Maturity of department agility of each department interviewed (n=14)

statements. Figure 4.7 represent the absolute frequency of the selected statements divided into departments.
While marketing and a part of the product team is classified under statement 1 and 2, the remaining is classified under at least experimenting with agile practices or even using them. Thereafter, the actual state and target state classification was made by means of categories which enable the distinction between the agile way of working and the plan-driven way of working. A total of 33 categories could be identified through the structured literature review. As the interviews proceeded, it became clear that different departments have different customer definitions. In this context, a distinction was made between the customer-oriented categories. Each customer-oriented category was divided into internal and external customer, whereas the internal customer represents another department within the organization and the external customer those who order the product online. In addition, the category "Customer external" was removed afterwards for the evaluation of the results, because none of the participants could make a statement about this category. This makes a total of 35 categories. Table 4.2 and 4.3 list all categories with their differentiation in agile and plan-driven.
The participants were asked to first classify the current state and then to name the target state according to the given scale both with an explanation. The scale is as follows: "Left applies strongly", "Left applies", "Rather left applies", "Rather right applies", "Right applies", and "Right applies strongly", whereas the left side represents the plan-driven way

Category	Traditional Development	Agile Development
Fundamental assumption	Systems are fully specifiable, predictable, and can be built trough meticulous and extensive planning	High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change
Control	Process centric	People centric
Management style	Command and control	Leadership and collaboration
Role assignment	Individual – favors specialization	Self-organizing teams – encourages role interchangeability
Communication	Formal and only when necessary	Informal and continuous
External customer's role	Important and low involvement	Critical and high involvement
Internal customer's role	Important and low involvement	Critical and high involvement
Knowledge management	Explicit	Tacit
Project cycle	Guided by tasks or activities	Guided by product features
Development model	Life cycle model (waterfall, spiral, etc.)	The evolutionary-delivery model
Organizational structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative, encouraging cooperative social action)
Technology	No restriction	Favors object-oriented technology
Quality control	Difficult planning and strict control. Difficult and late testing	Permanent control or requirements, design and solutions. Permanent testing.
Customer involvement external	Important usually only during project analysis	Critical and continuous
Customer involvement internal	Important usually only during project analysis	Critical and continuous
Team size	Often greater than 10	Usually fewer than 10
Team location	Predominantly distributed	Predominantly collocated
Team participation	Not compulsory	Necessary
Project planning	Up front	Continuous

Table 4.2.: Identified categories (1) [12, 15, 33, 41]

Category	Traditional Development	Agile Development
Feedback mechanism	Not easily obtainable	Usually numerous mechanism are available
Documentation	Substantial	Minimal
User requirements	Detailed and defined before coding/implementation	Interactive input
Cost of restart	high	low
Development direction	Fixed	Easily changeable
Testing	After coding is completed	Every iteration
Additional abilities required from developers	Nothing in particular	Interpersonal abilities and basic knowledge of the business
Appropriate scale of the project	Large scale	Low and medium scale
Developers	Oriented on plan, with adequate abilities, access to external knowledge	Agile, with advanced knowledge, co-located and cooperative
Customer external	With access to knowledge, cooperative, representative and empowered	Dedicated, knowledgeable, cooperative, representative and empowered
Customer internal	With access to knowledge, cooperative, representative and empowered	Dedicated, knowledgeable, cooperative, representative and empowered
Requirements	Very stable, known in advance	Emergent, with rapid changes
Architecture	Design for current and predictable requirements	Design for current requirements
Remodeling	Expensive	Not expensive
Size	Large teams and projects	Small teams and projects
Primary Objectives	High safety	Quick value

Table 4.3.: Identified categories (2) [12, 15, 33, 41]

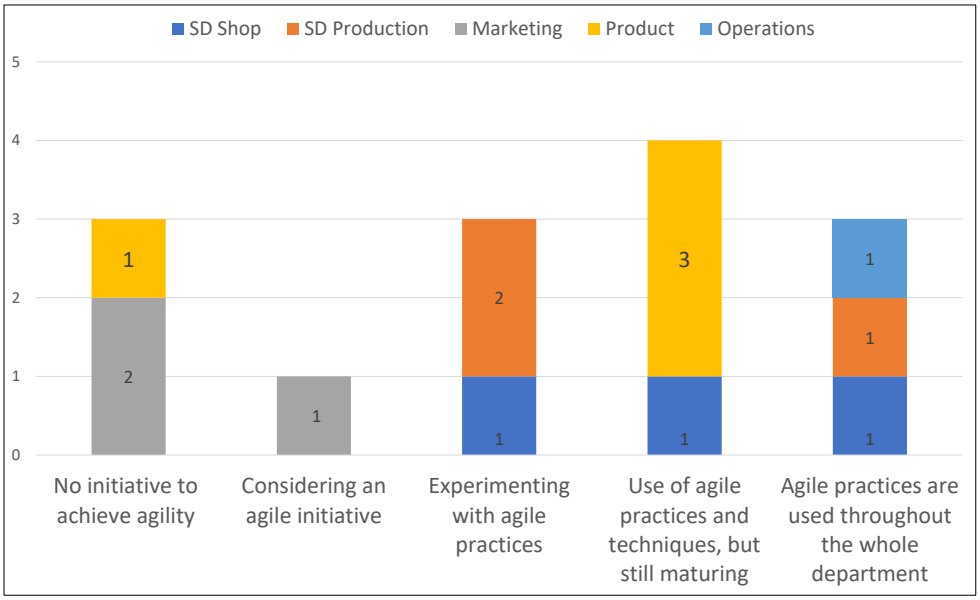


Figure 4.7.: Absolute frequency of the selected statements divided into departments (n=14)

and the right side the agile way. If the participants felt that a category did not apply to the organization, their area of responsibility etc., this category was left out. For simplification, the terms of the scale are abbreviated with "L1", "L2", "L3", "R3", "R2", and "R1", starting with "left applies strongly" = "L1".
The categories were evaluated by their mean value and classified according to the scale. Figure 4.8 represents the results for the actual state classification of each category by means of their mean value and standard deviation, whereas Figure 4.9 gives an overview about the results for the target state of each category accordingly. Figure 4.10 compares the results of the actual state and the target state by means of their mean value.

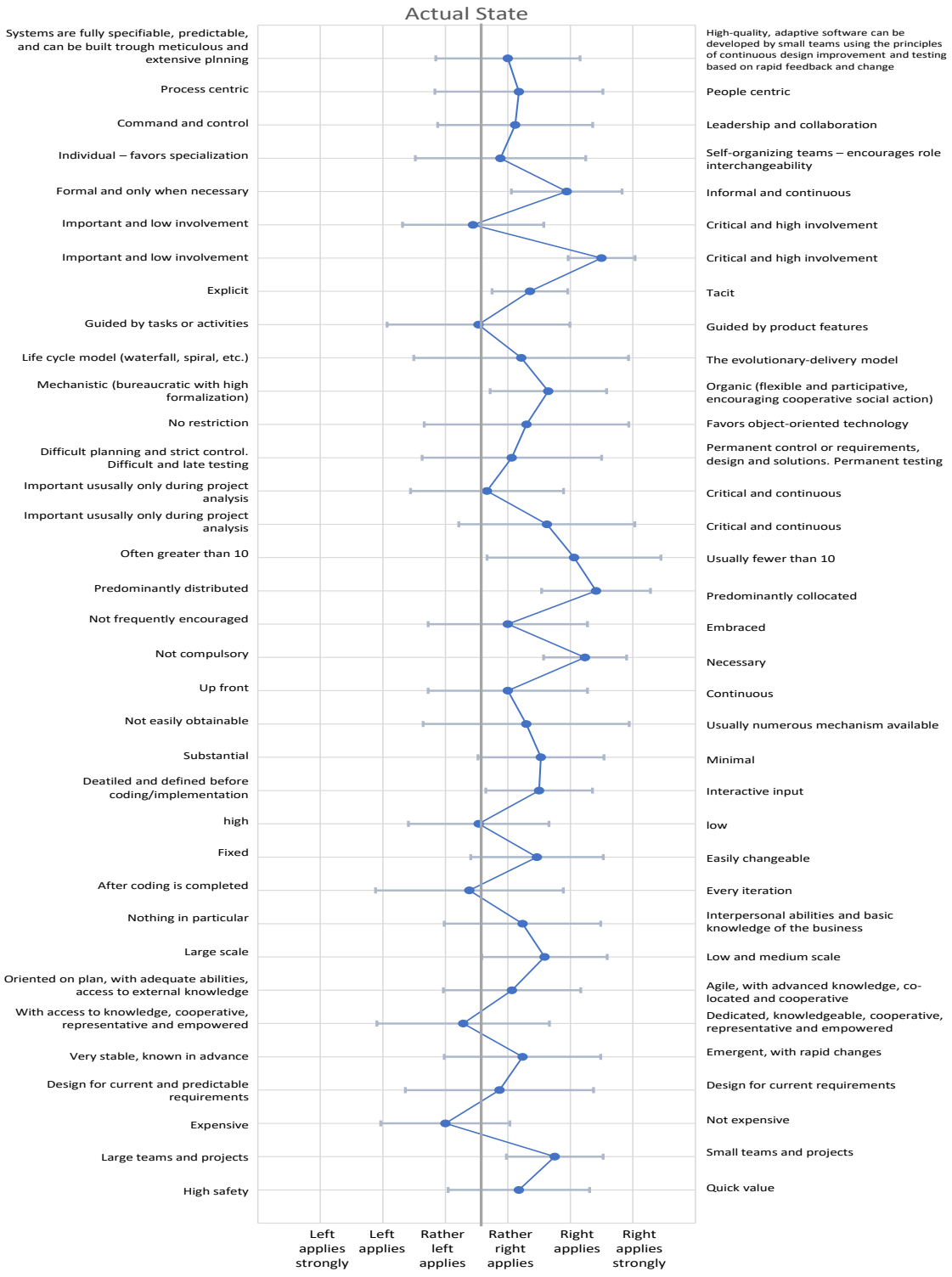


Figure 4.8.: Results of the actual state

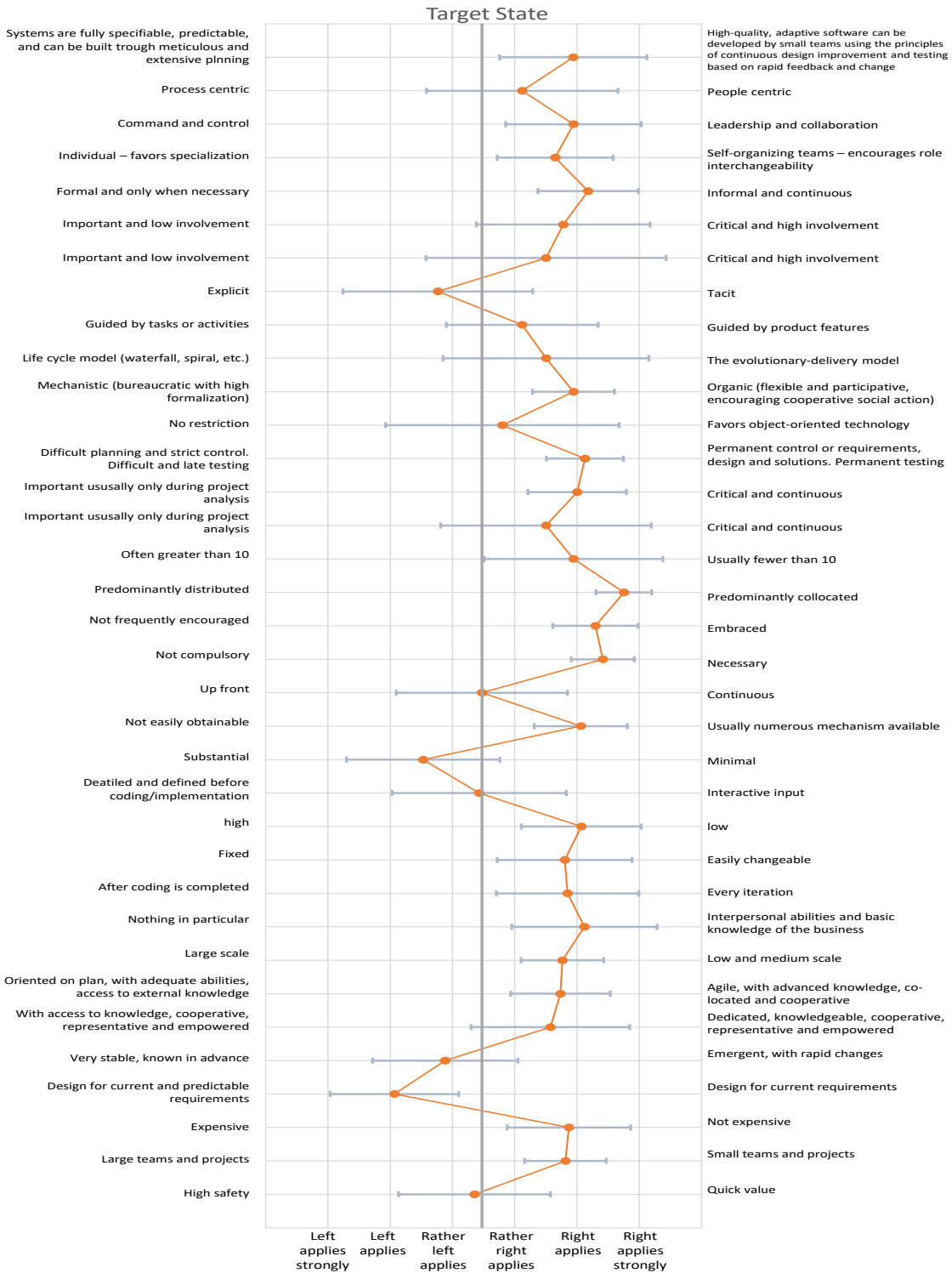


Figure 4.9.: Results of the target state

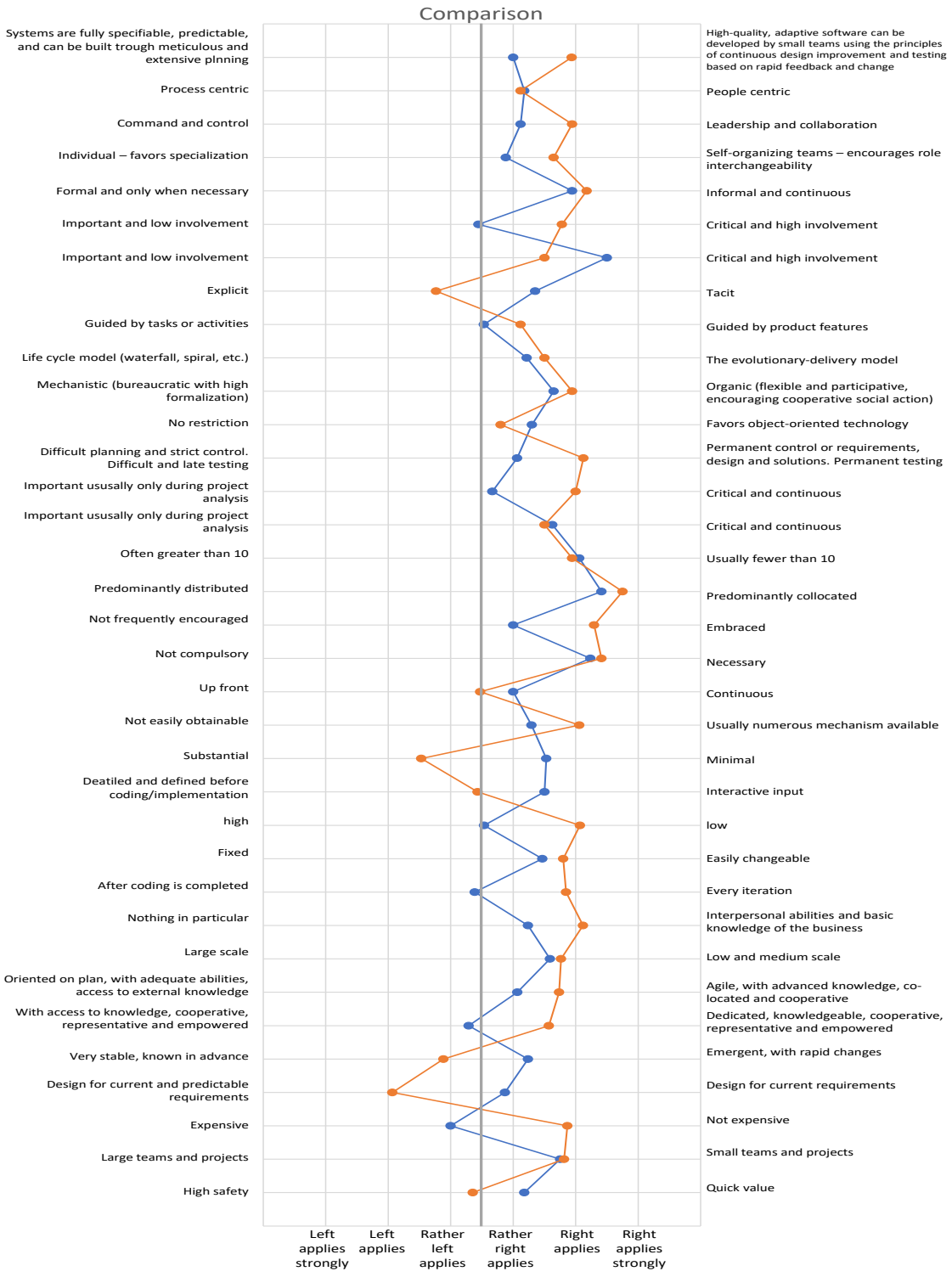


Figure 4.10.: Comparison of both results

When comparing the results of the actual state and the target state of each estimated category, it becomes clear which categories differ from each other to what extent e.g, the actual state and the target state of the category "communication" differ only minimally with a difference less than 0.5 of their mean values, whereas the actual state and the target state of "documentation" differ more than 1.5 in their mean values. This differences can be used to determine which categories the company should focus on more while proceeding with the large-scale agile transformation since categories with higher differences are further from the target state as categories that only differ minimally in their mean values. A total of 12 categories differ only minimally in their mean value from each other compared with the actual state and target state, 10 categories differ between a difference of 0.5 to 1.0, whereas 9 categories show a difference up to 1.5. The remaining 4 categories namely, documentation, cost of restart, architecture, and remodeling show the biggest difference in their mean value. However, the columns <0.5 and 0.5-1.0, which show a relatively small difference in their means, contain a total of 22 categories, which are 63% of the total categories. Consequently, 37% of the categories show a high difference to what the current state is and what the desired state is. Table 4.4 summarizes the categories ordered by the differences of their mean values.

<0.5	[0.5 - 1.0]	(1.0 - 1.5]	> 1.5
Control	Fundamental assumption	Customer's role external	Documentation
Communication	Management style	Customer's role internal	Cost of restart
Development model	Role Assignment	Quality control	Architecture
Organizational structure	Knowledge management	Customer involvement external	Remodeling
Customer involvement internal	Project cycle	Continuous learning	
Team size	Technology	User requirements	
Team location	Project planning	Testing	
Team participation	Feedback mechanism	Customer internal	
Development direction	Additional abilities required from developers	Requirements	
Appropriate scale of the project	Primary Objectives		
Developers			
Size			

Table 4.4.: Categories ordered by the difference of their mean value

Furthermore, not all categories, which differ in actual state and target state, should tend to be more agile. Although for 22 categories the objective is to achieve a more agile state, for 11 categories a traditional state is considered more appropriate. The remaining two categories have a difference of less than 0.1, it is assumed here that the current status corresponds to the target status. Overall 63% of the categories should be more agile, whereas 31% should be less agile and the remaining 6% correspond to the target state. Figure 4.11 provides an general overview of the tendency of the categories towards the agile or plan-driven way. Table 4.5 shows which categories, compared to the current status, should be

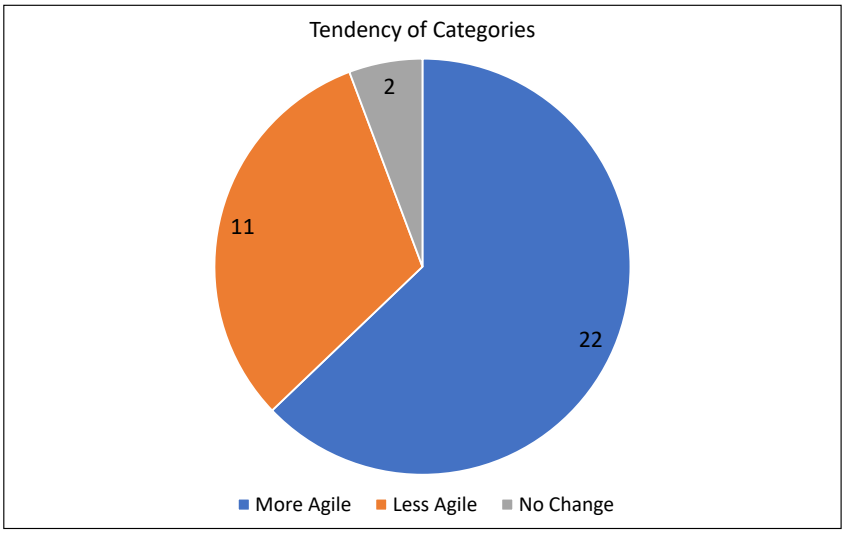


Figure 4.11.: Tendency of the categories

more agile and which less agile.
In the following, the absolute frequency of each scale point is shown below and each category is further explained. For each category, the entries for the actual status are specified in the first bracket whereas the entries for the target state are specified in the second bracket. The scale points are abbreviated as explained above e.g., L1 = "Left applies strongly", L2 = "Left applies" etc.
Fundamental assumption. [L1=0, L2=2, L3=3, R3=5, R2=5, R1=1, N/A=1][L1=0, L2=1, L3=1, R3=2, R2=6, R1=6, N/A=1] The basic assumption is rather that high-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change. I4, I11, and I15 indicate that the systems are not fully specifiable and also not predictable. Many requirements are fulfilled ad hoc (I4, I15, I16) and the software is continuously developed and optimized (I9). How-

More Agility	Less Agility
Fundamental assumption	Customer's role internal
Management style	Knowledge management
Role assignment	Technology
Communication	Customer involvement internal
Customer's role external	Team size
Project cycle	Project planning
Development model	Documentation
Organizational structure	User requirements
Quality control	Requirements
Customer involvement external	Architecture
Team location	Primary Objectives
Continuous learning	
Team participation	
Feedback mechanism	
Cost of restart	
Development direction	
Testing	
Additional abilities required from developers	
Appropriate scale of the project	
Developers	
Customer internal	
Remodeling	

Table 4.5.: Categories ordered by target agility

ever, some also indicate that there is no continuous design optimization and no continuous deployment (I2, I3, I14). The goal is to move further to R2.
Control. [L1=1, L2=3, L3=1, R3=5, R2=7, R1=1, N/A=0][L1=1, L2=3, L3=1, R3=2, R2=8, R1=2, N/A=0] The control is described as more person-oriented. Employees are relatively independent in what they do (I10), but it is also reported that Product Owners in particular are not completely independent of management and management still interferes (I1, I16). I14, and I11 state that the teams do not define their own goals, but distribute the tasks according to the given goal. I17, could not make a general statement, but stated that both occur. In this case 18 votes were cast. The status as it is, however, is also the goal. Full personal orientation is not desirable in the company, since processes provide the necessary structure (I6) and the maturity of agility in the teams is missing to allow them to work autonomously (I11).
Management style [L1=2, L2=0, L3=4, R3=3, R2=8, R1=0, N/A=0][L1=0, L2=1, L3=1, R3=2, R2=9, R1=5, N/A=0]. The management style in the organization tends to "Leadership and collaboration" rather than being a "Command and control" style. In some cases, manage-

ment is described as partly having the traditional mindset (I6, I9) which tends to be more controlling and commanding. This becomes particularly clear with important topics, the more important a topic is, the more the management tends to a controlling behaviour (I10). However, the style is described as being not controlling and it is usually an open cooperation, the teams work freely and the manager works together with the team (I2). In some cases the manager is a so called surfleader. Here is a relaxed cooperation very important and decisions by the manager are only made when requested by the team (I8, I15). The target state however is that the style should move further to the right. The aim is that the point "Leadership and collaboration" applies. Not fully, because there is always a part that requires control and command because the self-determined process cannot be fully trusted to the teams (I7, I11, I13, I14, I16, I17).
Role Assignment. [L1=2, L2=0, L3=4, R3=3, R2=8, R1=0, N/A=0][L1=0, L2=0, L3=3, R3=2, R2=10, R1=2, N/A=0] Role assignment is described as being rather depending on self-organizing teams. Some report that the team is fully responsible for the role assignment (I1, I4, I9, I14), other claim that they generally can decide within the team however, there is either a lead in the team who is trying to set something up (I2), or everyone already has a rough allocation of roles, so it is not possible to decide completely freely (I7, I8, I11). Some on the other hand report, that everyone in the team is specialized on one field and everyone has their predefined assigned tasks (I10, I12, I13). The target is to move slightly further to self-organizing teams, however it is reported, that sometimes a leadership position is needed to make decisions, in order to avoid unnecessary discussions (I8) or because a leadership position is more capable of identifying a person's strengths and weaknesses and, accordingly, the role (I15, I16, I17). In those areas where the roles are strictly defined, more openness is often expected, as this "being able to work in other areas" (I10) often enriches the level of knowledge (I10, I12).
Communication. [L1=0, L2=0, L3=1, R3=4, R2=7, R1=5, N/A=0][L1=0, L2=0, L3=1, R3=1, R2=9, R1=6, N/A=0] Communication within the organization is describes as already being "informally and continuous" due to being in constant exchange with team members or other departments (I13) or having regular meetings that promote communication e.g., Daily's, Retrospectives etc. (I2). Furthermore, communication is always possible if there is a need (I4, I19, I16). Nevertheless, some report that it depends very much on the department and the information itself (I14) and that the flow of information is sometimes very slow (I12). For the target state, communication only moves minimally further to the right.
Customer's role external. [L1=1, L2=0, L3=3, R3=4, R2=1, R1=0, N/A=0][L1=0, L2=1, L3=1, R3=0, R2=4, R1=3, N/A=0] Participants report that the external customer's role is already critical, that one looks and listens to customer numbers and customer feedback (I6, I1, I10). However, it is reported that there is not enough proximity to the customer (I17) and that decisions should be based more on the customer or customer opinions (I13, I12, I16) and have less of a commercial effect (I13, I11).
Customer's role internal. [L1=0, L2=0, L3=0, R3=0, R2=4, R1=4, N/A=1][L1=1, L2=1, L3=0, R3=0, R2=3, R1=3, N/A=1] The internal customer's role is described as being critical with a high involvement of the internal customer. Persons are in constant contact with the inter-

nal customer who is actively involved in the process, as he has to accept the product (I3, I7, I8, I9, I14, I15, I16). According to many, this should be the case, but some wish for less participation and more trust from internal customers (I9, I15).
Knowledge management. [L1=0, L2=1, L3=2, R3=8, R2=6, R1=0, N/A=0][L1=1, L2=6, L3=4, R3=2, R2=2, R1=2, N/A=0] Knowledge is partially communicated and shared (I6), but it is mainly owned by individuals or experts in the respective field (I3, I9, I11, I17) and is learned mainly by other people and not through existing documentation (I10). It is desirable to continue documentation, as this non-documented knowledge can often lead to loss of knowledge, e.g. in case of illness or loss of personnel, and the knowledge is therefore not secured (I6, I8, I15, I16, I17). In addition, familiarization with something new without documentation costs a lot of time (I7). A complete documentation of everything would not be feasible in this case, since there are always borderline cases, for example, that only individual persons know about, or some implicit knowledge is required (I14, I16). In this context knowledge management should therefore tend to be more explicit.
Project cycle. [L1=1, L2=3, L3=6, R3=2, R2=3, R1=2, N/A=0][L1=0, L2=2, L3=3, R3=5, R2=5, R1=2, N/A=0] The project cycle is reported being between "rather left applies" and "rather right applies". Many report that they are mainly guided by tasks and activities, because tasks are given and are just processed (I2, I9, I10, I11, I12, I13, I14). I16 reports, that this is again very department dependent e.g, marketing is guided by tasks and activities, whereas the engineering department focuses on the product (I16). Some state that they have only minimal predetermined activities or no activities at all (I3, I7, I15). The target state tends minimally to "rather right applies" due to the fact, while many report the target state should be guided by product features (I2, I3, I9, I10, I11, I12, I15), many are of the opinion that defined tasks and activities are necessary (I7, I13, I14, I16).
Development model. [L1=1, L2=3, L3=0, R3=1, R2=6, R1=3, N/A=3][L1=1, L2=2, L3=0, R3=1, R2=6, R1=4, N/A=3] Many claim to work with agile development models (I1, I4, I6, I7, I8, I15, I16). The focus here is on the methods Scrum and Kanban. Sometimes, however, there are projects that have fixed milestones and dates and which are then carried out using the waterfall model (I17). However, there are also areas that are strongly number-oriented (I13) and the implementation of agile methods in these sectors is not possible (I9, I12).
Organizational structure. [L1=0, L2=1, L3=1, R3=2, R2=12, R1=1, N/A=0][L1=0, L2=0, L3=1, R3=1, R2=13, R1=2, N/A=0] The reported organizational structure is really close to being organic. Some report the organization not being flexible due to a lack of personnel (I11, I13) and due to strict instructions (I12). Others report that the organization is neither bureaucratic nor highly formalized, but participant oriented and flexible (I2, I8, I10, I14). However, some claim that bureaucracy is slightly given and also necessary (I7, I15, I16). The goal is to move slightly further to being more organic.
Technology. [L1=1, L2=1, L3=0, R3=2, R2=4, R1=2, N/A=7][L1=2, L2=1, L3=0, R3=3, R2=2, R1=2, N/A=7] In general, there are restrictions to prevent "wild growth" (I3). There is a team defined base of technologies that everyone uses (I11, I2, I3, I4, I7, I8, I5). The advantage is that, no new programming language needs to be learned if staff leaves the company (I8). A little more in the direction of "no restriction" would be desirable to give new mod-

ern technologies a chance (I5, I7, I2).
Quality control. [L1=1, L2=2, L3=1, R3=5, R2=5, R1=2, N/A=1][L1=0, L2=0, L3=0, R3=2, R2=10, R1=4, N/A=1] Quality control is reported by several persons, being insufficient and happening only as much as necessary (I2, I5, I6, I7). Other reported that they continuously control requirements, design, and solutions however, in the end intensive testing still takes place but there is room for further improvements in this area (I8, I10, I11, I13, I15, I16).
Customer involvement external. [L1=0, L2=2, L3=2, R3=2, R2=3, R1=0, N/A=0][L1=0, L2=0, L3=1, R3=2, R2=2, R1=4, N/A=0] Involvement of the external customer is reported being too weak (I5), not continuous (I12, I17), and only happening at the beginning of a project (I11). Customer involvement is generally seen as in need of improvement, even if the customer is already involved (I1, I10). However, too much feedback and customer involvement is not considered efficient (I16).
Customer involvement internal. [L1=0, L2=1, L3=1, R3=0, R2=4, R1=2, N/A=1][L1=1, L2=0, L3=1, R3=0, R2=4, R1=2, N/A=1] The internal customer must accept the product and be satisfied with the outcome, therefore the involvement is reported being high (I2, I9, I8, I15, I16). On the other hand, many are satisfied with the fact that the customer only has a high level of participation at the beginning. This enables the developers to be independent of the internal customer (I7, I14).
Team size. [L1=1, L2=0, L3=1, R3=2, R2=4, R1=9, N/A=0][L1=1, L2=0, L3=2, R3=1, R2=5, R1=8, N/A=0] The team size is already small, it is mostly consisting of 5 - 8 persons. It is reported that this should be the case because the best experience has been made with small teams (I11), as well as the fact that more team members considerably complicate the distribution of tasks within the team (I2, I8). An area consists of more than 10 people, but this fits the number of tasks that exist (I10). In two cases, more personnel would be desirable, on the one hand because there are too many tasks (I15), and on the other hand to be able to set up agile teams (I12).
Team location. [L1=0, L2=0, L3=1, R3=1, R2=5, R1=10, N/A=0][L1=0, L2=0, L3=0, R3=0, R2=4, R1=12, N/A=1] Nearly all employees are at one location, in some cases they have to communicate with each other across sites (I7, I9, I15, I16). The actual state represents the target state.
Continuous learning. [L1=0, L2=3, L3=2, R3=6, R2=4, R1=2, N/A=0][L1=0, L2=0, L3=0, R3=2, R2=8, R1=7, N/A=0] Continuous learning is not really embraced within the organization. It is reported being rather frequently encouraged, than not. A lot of participants state that they have the respective tools (I3) which every one can use at any time (I11). A variety of training courses are also available to employees (I8, I9, I13). In some cases, however, these are considered to be too few (I12) or the impression is created that one has to "fight" for support in specific topics (I10). Many also report that it is not demanded by management and that one has to take care of it oneself and no time is released for it (I2, I6, I14, I17). However, I11 says that no company or management can be held responsible for the training of personnel, but that everyone must want it themselves. The goal of every participant is to have continuous learning.

Team participation. [L1=0, L2=0, L3=0, R3=2, R2=9, R1=6, N/A=0][L1=0, L2=0, L3=0, R3=0, R2=10, R1=7, N/A=0] Two participants indicate that the team is participating, but that too little attention is paid to the team (I10, I12). I13 reports that a team participation is not always necessary e.g., for topics that are number-oriented. I17, the CEO, reports that the team participation is fully necessary and takes places, but which is also reported as being exhausting, because the management team has to " deal" with the team to some extent. He would like a little less participation in some decisions here. The remaining participants are either satisfied with the situation or would like to increase the team participation even more.
Project planning. [L1=0, L2=2, L3=5, R3=2, R2=5, R1=3, N/A=0][L1=0, L2=6, L3=3, R3=3, R2=4, R1=1, N/A=0] Project planning is described by most as, rather continuously than in advance. A rough planning in advance always takes place at the beginning of the year, but this planning is continuously adjusted within the time frame (I3, I4, I5, I7, I8, I11, I12, I16). Others state that there is rough planning, but that the continuous ad hoc planning outweighs this planning (I14, I15, I17). Some report that up front planning is not even possible because of the frequent ad hoc topics. They wish for a rough up front planning just to know which "direction to go" (I2, I6). Only one participant reports the project planing being up front (I13). This, she says, is how it is demanded by the other departments. The target is to move slightly further to up front planning.
Feedback mechanism. [L1=1, L2=3, L3=1, R3=1, R2=7, R1=4, N/A=0][L1=0, L2=0, L3=0, R3=4, R2=8, R1=5, N/A=0] Some report that only one or too less mechanisms for receiving feedback exist (I4, I6, I9, I10) Those are mentioned being "too superficial" (I4) or "too profane" (I9). More than the majority, however, state that several mechanisms exist (I3, I5, I16, I17) such as retrospective meetings (I1, I2, I14), monthly feedback meetings (I7, I8, I15), individual meetings (I11, I14), annual meetings (I15), or team meetings (I11, I13). Participants who are dissatisfied with the number of mechanisms still see potential for improvement in the feedback mechanisms. The remaining participants are satisfied with the actual state.
Documentation. [L1=0, L2=1, L3=1, R3=5, R2=8, R1=2, N/A=0][L1=2, L2=9, L3=3, R3=2, R2=0, R1=1, N/A=0] Documentation is classified as being minimal in general. Only I13 can state that their documentation within the team is appropriate. It is reported that good documentation is available in some places, but much is still missing (I7, I8, I15, I17). For others, the existing documentation is often too little (I2, I8, I9, I12, I14). The desired target state is to have a substantial documentation. A fully comprehensive documentation is rejected because it becomes outdated too quickly (I14, I17) and partly serves as a job creation rather than fulfilling its purpose (I16).
User requirements. [L1=0, L2=0, L3=2, R3=4, R2=7, R1=1, N/A=3][L1=0, L2=5, L3=3, R3=2, R2=3, R1=1, N/A=3] Two respondents indicate that user requirements are specified in detail in advance (13, I17), but that there may be exchanges in individual cases. The others report that there is more interactive input. Some of them are satisfied with this, saying that this way current requirements can always be met (I11) and that an interactive input or exchange is less prone to errors than a definition given in advance (I13). However, many people wish to have detailed requirements in advance, since most of them are not detailed

enough, which only slows down the developers' work (I2, I7, I8, I15).
Cost of restart. [L1=0, L2=2, L3=7, R3=3, R2=2, R1=1, N/A=2][L1=0, L2=0, L3=1, R3=3, R2=5, R1=6, N/A=2] More than half indicate that the restart costs would be high. The reasons for this would be that such a thing is automatically associated with costs (I3, I10), there is too little documentation and specialist knowledge may have been lost, which massively increases the familiarization period (I4, I7, I9, I11, I17), or that the high familiarization period is due to the complexity of the systems (I15, I17). However, six participants claim the costs of restart being already low, but without any appropriate explanation.
Development direction. [L1=0, L2=1, L3=2, R3=2, R2=9, R1=1, N/A=2][L1=0, L2=1, L3=1, R3=1, R2=9, R1=3, N/A=2] The direction of development is for the most part described as easily changeable and flexible. There is no predetermined direction that each team must follow (I14). Often the developers get stuck and have the opportunity to discard work (I8, I15). Some people still think that the development direction is too rigid. The reason in this case is attributed to the prevailing lack of personnel in the company (I11. I13). More flexibility in this respect would be desirable (I11, I13, I17). However, two people report that they would feel more comfortable with a fixed direction of development in order to have a rough guideline and be able to stick to the work they have started rather than constantly discarding work (I2, I7).
Testing. [L1=1, L2=3, L3=4, R3=1, R2=3, R1=1, N/A=4][L1=0, L2=1, L3=0, R3=3, R2=5, R1=4, N/A=4] Testing is associated with a great deal of time (I3, I11), resulting in little or no testing (I5, I6). Many would also not understand the long-term thinking behind it (I3). Others report that testing is done, but not iterative, but rather as soon as the developer has completed his complete coding (I2, I7, I8, I14). However, a large number also report that testing happens iterative (I1, I4, I9, I15, I16).
Additional abilities required from developers. [L1=0, L2=2, L3=3, R3=3, R2=7, R1=2, N/A=0][L1=0, L2=0, L3=3, R3=1, R2=4, R1=9, N/A=0] For the most part, interpersonal skills and basic knowledge of the company are already somewhat present. This is due to the fact that the developers are often in contact with the internal customer or work a lot together as a team (I2, I8, I10, I15). Nevertheless, there are people in the company who do not master these skills or who do not feel comfortable working with other people (I4, I12, I13, I14, I15, I16). Here the aim is to take a step to the right.
Appropriate scale of the project. [L1=0, L2=1, L3=0, R3=7, R2=6, R1=3, N/A=0][L1=0, L2=0, L3=0, R3=6, R2=9, R1=2, N/A=0] The appropriate scope of a project in the company is described as rather medium or small. Spread over the year there are one or two large projects (I9, I11), but these are divided into the smallest possible steps (I2, I4, I7, I8, I11, I15, I16, I17). This makes it possible to react more flexibly and faster (I3, I14), and also gives you a concrete goal to work towards (I2). All in all this also reflects the desired state.
Developers. [L1=0, L2=2, L3=1, R3=7, R2=4, R1=1, N/A=2][L1=0, L2=0, L3=1, R3=4, R2=8, R1=2, N/A=2] Developers are described as rather agile. There are many developers who like working with agile methods, but also many who do not have the agile mindset (I8, I15). It is reported that some developers are more oriented to plan than to be agile (I2, I5, I13). Achieving complete agility is not possible because a minimal plan must always

be created (I15). The goal would still be to have developers with an agile mindset in the company.
Customer internal. [L1=0, L2=3, L3=1, R3=1, R2=2, R1=0, N/A=2][L1=0, L2=1, L3=0, R3=1, R2=4, R1=1, N/A=2] The internal customer is more likely to be referred to as dedicated and knowledgeable, because he either does not participate in the process with commitment (I15), or has insufficient knowledge to assess the requirements (I14). A close and knowledgeable customer is desired, who should have some knowledge to assess whether the implementation is possible (I2, I14) and to lead a cooperative relationship (I15).
Requirements. [L1=0, L2=2, L3=2, R3=6, R2=4, R1=3, N/A=0][L1=2, L2=5, L3=4, R3=5, R2=1, R1=0, N/A=0] Requirements do rather emerge and change rapidly, than being stable and known in advance. Participants report that they are roughly given and known, but do change. This is due to the rapid changes that occur and to which it responds (I14, I11). Some report that requirements are generally known in advance and that the overall goal is clear (I13, I17). In any case, stable requirements that are known in advance are desirable, e.g., as they allow better planning (I8).
Architecture. [L1=1, L2=2, L3=3, R3=3, R2=4, R1=2, N/A=2][L1=4, L2=8, L3=2, R3=0, R2=1, R1=0, N/A=2] The architecture is designed to meet current requirements rather than predictable ones. Some report that the architecture is already designed for current and foreseeable requirements (I17, I2, I3, I1, I14), but sometimes this cannot be implemented as well, because the fast value has priority (I2), or because one should not plan so much in advance, because this means that the more general the architecture has to be designed (I14). However, a large part reports that architecture is generally designed only for current requirements. Looking into the future is difficult to implement (I9), and thus only those things are designed that are certain to be viable in the future (I7). I11, is of the opinion that one should not look into the future, but only deal with rapidly changing requirements. The remaining participants see the goal as designing architecture with current and predictable requirements.
Remodeling. [L1=0, L2=6, L3=5, R3=2, R2=2, R1=0, N/A=2][L1=0, L2=0, L3=1, R3=5, R2=4, R1=5, N/A=2] Due to the high complexity of the systems (I7, I8, I14, I15, I17), the gone employees with sufficient knowledge (I4) and insufficient documentation (I7), the redesign is perceived as rather expensive. A clear goal is here to arrange a reorganization inexpensive.
Size. [L1=0, L2=0, L3=1, R3=4, R2=9, R1=2, N/A=1][L1=0, L2=0, L3=0, R3=5, R2=9, R1=2, N/A=1] The size of the teams and the projects are small according to each participant. This also corresponds to the target status, because with small teams the overview can be kept better (I8) and they work more efficiently and faster (I13).
Primary Objectives. [L1=0, L2=2, L3=2, R3=5, R2=7, R1=1, N/A=0][L1=1, L2=3, L3=6, R3=3, R2=4, R1=0, N/A=0] The primary goal in the company is rather the fast value than the high security. Security should not be confused here with IT security, on which high value is placed (I6, I14). However, the fast value here is the driver (I14) to increase sales (I12). Everything has to happen so quickly that in some cases security is not a major priority, reports I10. A few state that the primary goal is security, e.g., because they cannot afford to make mistakes in their area (I7, I13). The common primary goal should rather be high

security and one should be able to test a little longer (I2).

4.3.2. Impact of the Large-Scale Agile Transformation on the Organization

This subsection presents the results on the impacts of the agile transformation on the organization.To study these, the sociotechnical systems theory has been used a guide. In this context, the impacts on the organization are described in the following by means of the four components, namely structure, task, actors, and technology. Figure 4.12 provides an overview of all identified impacts on the organization.

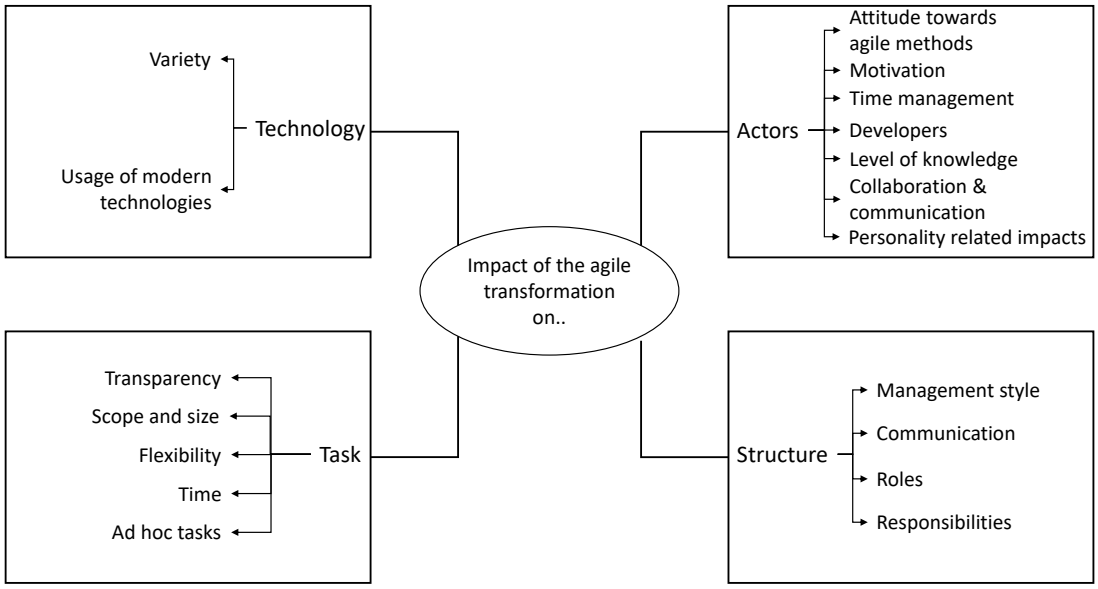


Figure 4.12.: Impacts of the large-scale agile transformation on the four components

Impacts on the tasks
The term refers to every step within the organization to produce goods or services including all sub tasks [29].
Transparency. With the introduction of agile boards tasks got more transparent (I1, I6, I7). Everyone is able to look at any time at the board and see exactly e.g., which task is currently being processed or is already finished (I4, I15).
Scope and size. Introducing agile methods had an impact on the tasks in terms of size and scope. Tasks need to be defined more precisely and shortened (I4, I14, I15, I16) to enable estimation of the effort and to fit them into the short iterations (I2, I6, I7, I8).
Flexibility. Using agile methods enables faster reorientation in the implementation process if the team notices after two or three iterations that the chosen approach does not work or

if problems arise (I3).
Time. On the other hand, better definition and precision of tasks requires more time to be invested in defining the tasks (I15). Furthermore, attending to all meetings which are part of the process of agile methods is felt to be a waste of time (I16).
Ad hoc tasks. Ad hoc tasks are difficult to implement since everything is planned ahead for two weeks and everyone needs to fully commit to the plan (I8, I14).

Impacts on the structure
Structure includes the terms "systems of communication, systems of authority, and systems of work flow" [29].
Improvement of communication. Communication as well as cooperation within teams, with other areas, or people improved by the fact that agile methods require a high level of communication. (I2, I4, I6).
Communication. This impact can be subdivided into three categories which are explained in the following.
Single-point of contact. With the introduction of new roles, a single-point of contact has been created to whom everyone outside the team can turn in case of questions or problems (I8, I7, I14, I15). This allows team members to work undisturbed and more efficiently since in case of problems the single-point of contact is contacted and not a team member.
Redundant communication. Due the fact that processes got more transparent and are portrayed on a agile board to which everyone has access, redundant communication e.g., questions referring the state of a task, could be eliminated (I7, I15).
Roles. With introducing agile methods, new roles have emerged in the company e.g., agile coaches or team leads who encourage the agile transformation and support teams in the execution of agile practices (I1, I3 I7, I8), and product owner who are responsible for the product (I14).
Responsibilities. There were no longer clear leadership positions, which led to confusion for some, but also to the fact that management is not involved in every decision making process and responsibility has shifted from the management and lies now more with the teams (I17). Decisions are made within the team or by cross-functional teams and are no longer dependent on management's approval (I17) which has an additional impact on the speed (I9).
Management style. There has been a change of management style from giving orders to leading and managing the team (I14). In addition, more responsibility in the teams (I14) goes hand in hand with a loss of control by managers which partly led to an even stronger rejection of agile methods (I16, I17).

Impacts on the actors
Actor represent every employee in the organization with the attitude, culture, mindset etc.
Attitude towards agile methods. After introducing agile methods, two contradicting opinions emerged on agile within the organization. Wanting to promote and defending agile methods lead to tension and conflicts with several non-agile fields or persons (I11, I14). On

one side, people who encouraged working agile, who enjoy the benefits of working that way (I14) and on the other side, people rejecting the introduction of agile methods and not supporting them (I6, I7, I8, I16). Some people even left the company because they could not work according agile methods (I11). However, some report that there has also been a change of attitude towards agile methods. Some people's attitude changed from being skeptical to being satisfied with the way of working because of fast and good results (I17) and the openness to such methods has improved (I5).
Motivation. Motivation of people increased due the fact that tasks got more shortened and success can be achieved earlier (I3). Also, the increased personal responsibility leads to being more satisfied with one's work (I1, I9) and thus also more committed to it (I16).
Time management. All introduced meetings e.g., daily's and retrospectives as well as the precise definition of tasks are perceived as annoying and a waste of time (I3, I8, I15).
Developers. Due to the introduction of a single-point of contact, especially developers only receive important, relevant tasks and irrelevant ones drop away (I7, I8). This allows them to work more quietly (I15).
Level of Knowledge. With the introduction of agile methods in the company, many people got involved in the topic and dealt with it which led them to expand their knowledge on agile methods in general (I5, I10). Also, knowledge from other departments could be gained by working together with people from other departments (I6).
Collaboration & communication. Collaboration and communication improved both within the own team (I2, I3) and with people from other teams (I10, I15) due the fact that the management is no longer responsible for coordination, instead people need to communicate and work together towards the goal.
Personality related impacts. Working in teams with no hierarchies consisting of people of different areas evokes a feeling of not belonging in the team (I13) and also triggers a feeling of missing perspectives (I17).

Impacts on the technology
Technologies are described being "problem solving inventions" [29].
The least impact was on the technologies used in the company. I3 and I11 report that the variety of technologies increased due the fact that processes got more flexible and projects smaller. Also, short iterations and communication improvement have the effect that more modern technologies can be used (I6, I15). The remaining participants could not notice an impact on the used technologies.

4.3.3. Challenges

Data was coded using a deductive approach, as described by [13] resulting in a list of a total of 37 codes. On the basis of these codes, 7 categories have been identified which are explained in detail in the following. Table 4.6 gives an overview about all categories.

Challenges	Reported by
Change process	I1, I3, I5, I10, I11, I13, I14, I15, I16, I17
Cooperation of agile areas with non-agile areas	I2, I7, I8, I10, I11, I14, I15, I16
Management	I1, I6, I9, I10, I13, I17
Lack of personnel	I9, I10, I12, I13, I16
Team constellation	I6, I10, I12, I13, I17
Personal challenges	I2, I4, I9, I13
Hierarchical challenges	I6, I14

Table 4.6.: Identified challenges

Change process
A challenge is the transformation of the mindsets of people (I8, I13, I14, I15, I16) and breaking work habits (22). Since everyone needs to fully commit to the agile methods, the rejection of agile methods or change in general is a major challenge for the organization because working with people that have a traditional mindset can inhibit the agile transformation since those people do not fully enable agile processes. Some people rejected change in general regardless of whether agile or non-agile (I1, I15, I17). Others refused the change to agile methods because e.g., they made bad experiences with agile methods in the past or they don't even want to do it on principle (I1, I3, I5, I10, I11). On the other hand, some people were not happy about the speed and claimed the transformation is proceeding too slowly (I11, I14).
Cooperation of agile areas with non-agile areas
Another reported challenge is the collision of agile areas with non-agile areas made cooperation difficult and caused conflicts between people due to completely different mindsets, processes, and often no understanding of agile methods (I2, I7, I8, I10, I11, I14, I15, I16).
Management
This category can be split up into three subcategories which are explained in the following.
Management against agile. There has been an a priori rejection of agile by the management. I1, I10, and I17 report that the topic was not taken seriously enough by them and that the management did not support it.
Unwillingness to give up control. One of the most important aspects of agile are the self-organizing teams. However, management was not fully able to give up control and trust the team (I1, I9).
Management still thinking in waterfall model. Although claiming to be agile, the management has difficulties to approach agile projects because of their traditional mindset. I6 and I13 report that despite these claims, management still regularly excepted fixed dates and an effort estimations which they were not able to deliver. The challenge in this context is to change the managements attitude to agile and to develop a agile mindset (I3, I9, I13).
Lack of personnel
The lack of personnel has been reported as challenging during the squad experiment be-

cause personnel has been missing in important positions and daily business remained undone (I9, I19, I12, I13, I16). Consequently, more personnel is needed in order to work in a agile setting (I8, I14, I12).

Team constellation

Since working in teams requires a high level of communication, it is necessary that the team members harmonize with each other. Consequently, to find a suitable team composition is considered challenging (I10). In addition, the organization's cooperation with their parent company is increasing. In this context some reported, that a challenge will be to find a appropriate cross-located team constellation with their parent company (I6, I12, I13, I17) who are at the same time autonomous and can take on a high level of responsibility.

Personal challenges

Agile requires a high level of team participation which can be difficult for those with a weak communication ability (I4). Some find it difficult to work with people from different areas of expertise, in addition, the team constellation of members of different areas can lead to the feeling of no clear team identity (I6). Lastly, the fact that there are no clearly defined goals can lead to demotivating people (I2).

Hierarchical challenges

When the organization's goal is to perform an company-wide agile transformation, it is necessary to break up linear company structures (I9, I11). Some report the problems of team leaders who have been assigned to autonomous teams.

4.3.4. Barriers

After analyzing the barriers, a list of 18 codes could be identified resulting in four categories. Table 4.7 gives an overview about all barriers. The definition of each category is provided in the following.

Barriers	Reported by
External influences	I1, I4, I5, I6, I10, I11, I14, I17
People unwilling for change	I2, I8, I12, I13
Lack of personnel	I8, I11
Traditional management	I2, I15

Table 4.7.: Identified barriers

External influences

Many report the migrating project to the technological platform of their parent company as a barrier, which is executed by them (I5, I6, I10, I11). For now, they have no clear perspective on what will happen after the connection of both systems (I4, I14), which does not allow to really get on with the agile transformation (I1).

People unwilling for change

If especially the majority of people from the teams are not willing for change or reject agile

methods, the agile transformation will not be successful since the execution of agile processes relies on them (I2, I8, I12, I13).

Lack of personnel

With a lack of personnel, regardless of whether they support agile methods or not, the organization is only able to do basic work or daily business (11). Another barrier is the lack of appropriate personnel e.g., Scrum Masters that encourage the agile transformation and coach teams (I8).

Traditional management

Having a management that does not have an agile mindset or does not support the agile way of working allows agility only up to a certain point (I2) or rather inhibits agile processes (I15) since e.g., the change of corporate structure decision lies with those.

4.3.5. Success Factors

To observe factors to successfully implement the agile transformation, data was coded using a deductive approach. From a list of 40 codes, a total of six categories could have been identified. Table 4.8 summarizes the categories. The categories and their definitions are provided in the following.

Success Factors	Reported by
Personnel	I3, I4, I5, I6, I7, I10, I11, I12, I13, I14, I15
Coaching & training	I1, I5, I9, I14, I16, I17
Introduction of agile methods	I2, I3, I8, I10, I12, I15
Management support	I6, I12, I14, I15
Level of knowledge	I7, I10, I17
Increasing motivation	I1, I5

Table 4.8.: Identified success factors

Personnel

The most mentioned success factor refers to the appropriate personnel. First of all, to succeed at the agile transformation, enough personnel is needed so that work does not remain undone and suitable teams can be formed (I10, I13, I15). Secondly, the personnel needs to have an agile mindset (I3, I6, I10, I13, I15) or at least needs to be open for change in order to enable the agile transformation (I4, I5, I6, I10, I11, I12, I14).

Coaching & training

Another much mentioned success factor is providing coaching and training by agile coaches. This will more likely increase the openness to the topic and consequently lead to acceptance by people (I1, I5, I9, I14, I16, I17). Furthermore, the supervision of agile coaches ensures the correct implementation of the agile methods.

Introduction of agile methods
This category is composed of how to introduce agile methods and where to introduce them. When introducing agile methods, these should be customized according the teams and ideally implemented step by step instead of all at once (I3). However, essential tasks should not be neglected (I15). Additionally, great attention should be paid to consistently enforcing the agile methods despite the problems that have arisen (I2, I3, I10). Furthermore, when planning to introduce agile methods, these should be introduced in several teams and scaled throughout the organization (I2, I8, I12).
Management support
Despite the management being part of the personnel, these were pointed out specifically. Management support needs to be assured (I6, I12, I14, I15).
Level of knowledge
In order to be able to work together efficiently, the level of knowledge needs to be the same (I7) or at least having some people in teams with already existing knowledge in the topic will guide inexperienced team members (I10, I17).
Increasing motivation Increasing the motivation of employees by means of conducting experiments and letting people have positive experiences with agile will increase the probability of acceptance (I1, I5).

4.3.6. Lessons Learned

A total of eight lessons learned could be identified by means of 42 codes. Table 4.9 provides an overview of all lessons learned. The definitions are explained in the following.

Lessons learned	Reported by
Appropriate people needed	I5, I6, I8, I10, I11, I13, I14, I16
Agile transformation requires time	I11, I13, I14, I16, I17
Management support	I1, I6, I9, I14, I17
Application area	I13, I17
Coaching & supervising	I2, I14
Customization of agile methods	I2, I3
Enough personnel	I12, I13
Strongly personality dependent	I1, I5

Table 4.9.: Identified lessons learned

Appropriate people needed
In order to successfully perform an agile transformation, appropriate people are needed. These involved in the process need to have an agile mindset (I5, I10, I13, I14), need to be fully committed to the process (I6, I8, I11, I16), and need to be willing to work with different people of different fields (I10).

Agile transformation requires time
The agile transformation process is describes as complex and tedious (I16, I17) which can not happen overnight (I13). If people are not given time to accept the change and deal with the issue, it is more likely to lead to conflict and rejection (I13, I14).
Management support
Management support is again mentioned in this context. The support by the management is reported es crucial because contradicting mindsets are likely to cause conflicts between people (I14, I17), performing agile will be only possible in specific departments but scaling the methods will not be possible, and the management influences the mindset of the team (I6) which makes the change process even more difficult and will slow down the agile transformation process (I9).
Application area
Another lessons learned is that the agile way of working does not always fit in specific departments or projects. I13 reports that agile methods are very dependent on the respective area and that departments with daily business or defined processes can not adapt these methods. Also, projects that do have a fix deadline and given processes can also not be performed using agile methods (I17).
Coaching & supervising
Coaching and supervising teams is necessarily needed because it serves to introduce people to the topic and prepares them for the change (I2). In addition, having agile coaches will encourage the agile transformation (I14).
Customization of agile methods
I2 and I3 report that adapting Scrum as it is described is not likely to work and that agile methods should not be enforced if they do not work well for the team, instead they recommend to customize the chosen approach. Therefore, each team need to customize their method in order to establish a well working setup with which every team member is satisfied.
Enough personnel
In order to be able to build appropriate teams with each one or two people of each field, enough personnel is required (I12, I13).
Strongly personality dependent
Whether someone likes working in an agile environment is strongly personality dependent. One one side, working in an agile environment motivates the team members and leads to fast results (I1), whereas on the other hand people can get demotivated as they are constantly throwing work away (I5).

5. Discussion

This chapter presents and describes the key findings of this thesis in Section 5.1 and its limitations in Section 5.2. The limitation section discusses the validity of this thesis by using a recommended scheme [37].

5.1. Key Findings

Achieving full agility in every category is not the organizational goal

For the actual state, 23 categories of a total of 35 categories can be classified in the range between "Rather right applies" and "Right applies". Whereas 2 categories can be classified between "right applies" and "Right applies strongly" and 7 categories between the range of "Rather right applies" and "Rather left applies". For the target state, 21 categories of 35 categories are classified in the range of "Rather right applies" and "Right applies" whereas 6 categories are classified in the range of "Right applies" and "Right applies strongly". For 4 categories in fact it would be more appropriate being in the range of "Rather left applies" and "Rather right applies", as well as for the remaining 4 categories that are classified between "Left applies" and "Rather left applies". After comparing the results for the actual state of agility and the target state of agility, it can be derived that the organizational goal is not to achieve full agility or agility at all in every category.

Large-scale agile transformation has impact on the whole organization

As in the case study of Gerster et al. [20], by using the socio-technical systems theory as a guide to investigate the impacts of the large-scale agile transformation on the organization, in this thesis it could be derived that the agile transformation not only has impacts on the units using agile methods, but on every working system and therefore on the whole organization itself. The large-scale agile transformation has the greatest impact on the people in the company followed by the tasks. However, the least impact of the agile transformation could be identified having on the used technologies and not on the structure as in the case study of Gerster et al. [20]. Also, while only some impacts identified in this thesis can be found in literature, many new impacts were identified in this thesis. Furthermore, identical impacts, e.g., Role, are classified in different components. Gerster et al. [20] classify the impact on the role in the actors components, whereas the same impact is classified in the component structure in this thesis.

Continuous emergence of new challenges

Due to the available quantity of literature on the challenges and success factors of agile transformation, duplicates were initially being expected between the challenges and success factors of the existing literature and those mentioned in the interviews. Although this applies fully to the identified success factors, literature is missing the challenges of the team constellation, personal challenges, and the lack of personnel which were identified in this thesis. From this it can be concluded that the success factors and challenges already identified by this thesis have been confirmed, but that new challenges are constantly emerging that need to be addressed by organizations.

Success factors and lessons learned can be derived from emerged challenges and barriers

Emerging challenges and barriers within the organization due the large-scale agile transformation form the success factors for performing a large-scale agile transformation successfully as well as the lessons learned of a large-scale agile transformation and often demonstrate a possibility to overcome those challenges and barriers. The challenge "change process" and barrier "people unwilling for change" describe the challenge of having people in the organization with traditional mindsets and being unwilling for change. The success factors "personnel" as well as the lesson learned "appropriate people needed" addresses this challenge and provides with the success factor "coaching & training" a solution for building an agile mindset of people. The challenge "management" and barrier "traditional management" are addressed by the success factor and lessons learned "management support".

Success of the large-scale agile transformation depends on the people in the organization

In order to successfully perform a large-scale agile transformation, all people involved in this process have to fully commit and support this transformation. Four of the seven identified challenges, namely "change process", "management", "personal challenges", and "team constellation", as well as two of the four identified barriers, "people unwilling for change", and "traditional management" are people related. This challenges and barriers emerge due to people, who are either not willing for change, do not support the introduction of agile methods, or are not able to adapt to these methods. This makes 57% of the challenges and 50% of the identified barriers. Since success factors and lessons learned can be derived from the emerged challenges and barriers, they show similarities. Four of a total of six success factors, namely "personnel", "coaching & training", "management support", and "increasing motivation" as well as four of the in total eight identified lessons learned, "appropriate people needed", "management support", "coaching & supervising", "strongly personality dependent". This makes 57% of the identified success factors and

50% of the lessons learned.

5.2. Limitations

This section represents threats to the validity of this thesis. The validity of a thesis indicates the trustworthiness of a study which are discussed in the following according to the recommended scheme by Runeson & Höst [37]. Accordingly, four distinctions of validity can be made.

Construct validity. This validity aspect addresses the degree to which the operational measures examined actually reflect what the researcher intended and what is being examined according to the research questions [37]. To address this validity, several people of different fields were interviewed, a introduction to the topic before the interview took place, and the interviewees were reminded frequently to ask questions in case of ambiguity. Nevertheless, because of the restricted number of interviews and the limited time of some interviewees, there is a possibility that not all occurring impacts, challenges, barriers, and success factors within the organization are included in this thesis.

Internal validity. This aspect should be addressed when examining causal relations. This can be a threat to the thesis' validity when the investigated factor is also affected by a third factor, which is not known or the researches is not aware of [37]. In order to address this aspect, interviewees were asked explicitly about the factor mentioned in this thesis, namely the agile transformation, and the participants were also explicitly reminded of this factor. In addition, they were reminded of the fact that only impacts which took place in this company should be mentioned.

External validity. External validity concerns the degree to which the findings can be generalized and to which the findings are of interest to people not included in the investigated case [37]. As this thesis analyzes occurring success factors, challenges, barriers, lessons learned, and the impacts on the organization, which can emerge during an agile transformation, the results can be generalized for any organizational member intending to initiate or currently undergoing an agile transformation within their organization. The analysis of the current state and the target state can be used as a framework and applied to any organization currently undergoing an agile transformation.

Reliability. This aspect addresses the degree to which the analyzed data depends on a researcher. Accordingly, the results of another researcher conducting the same study should be the same [37]. Since this study relies on interviews and therefore subjective opinions, conducting this study at a different organization can not guarantee same results. This is because another organization can have a different maturity of business agility, different employees with different cultures and mindsets, and a different approach for initiating the

agile transformation.

6. Conclusion

This last chapter of this thesis provides in the first section a summary of the thesis based on the described research objectives in Section 1.2 and presents in the second section an outlook for future work.

6.1. Summary

In the following, the findings are summarized by answering the four research questions of this thesis.

Research Question 1: What is the actual state regarding the large-scale agile transformation?
In order to answer the first research question, categories were identified that enable the distinction between the plan-driven way of working and agile way of working by conducting a structured literature review. Thereafter, participant were asked to classify the actual state of all 35 categories according to the scale "left applies strongly", "left applies", "rather left applies", "rather right applies", "right applies", and "right applies strongly" with the left side representing the plan-driven way. Subsequently, the mean of each category was determined in order to be able to provide a general statement. In addition, each participant was asked to determine the general maturity of the business agility and also the agility maturity of their departments. The majority of the participants report that agile practices and techniques are used within the organization but are still maturing. However, a clear difference in the agility maturity becomes apparent when observing the individual departments. While departments like the software engineering and product state that they already use agile methods and practices, marketing reports that they do not even consider introducing agile methods. After analyzing all 35 categories and classifying them by means of their mean, a total of 26 categories were reported being agile whereas 9 categories tend to be rather plan-driven.

Research Question 2: Which impact has the large-scale agile transformation on the organization?
To answer the second research question participants were asked to describe the impact of the large-scale agile transformation on each working system as described in the socio-technical systems theory, namely actor, task, structure, and technology. Subsequently, categories for each working systems were identified. By using the socio-technical systems

theory as a guide to investigate which impact the large-scale agile transformation has on the organization, it could be derived, that the agile transformation not only has a impact on the units using agile methods but on the whole organization itself. The agile transformation had the greatest impact on the people in the company followed by the tasks. The introduction of agile methods caused two contradicting opinions within the organization towards agile methods. Tasks got more transparent by using agile boards and shrink in size so that they can be fitted in one short iteration. The most mentioned impact on the tasks is that these got more detailed in order to estimate them correctly and the flexibility of how to deal with tasks increased due to being enable to reorient faster. Furthermore, the most mentioned impact on the structure is the increase of communication within the team and across different teams or departments. The least impact could be identified on the used technologies within the organization. Technologies are described being more modern and vary widely.

Research Question 3: What are success factors and barriers of the large-scale agile transformation?
To answer this research question, participants were asked to list the emerged challenges, barriers, and success factors due the large-scale agile transformation. Thereafter, data was coded using a deductive approach in order to identify categories for challenges, barriers, and success factors. This thesis identified a total of seven challenges, four barriers, and five success factors which are listed in Table 4.10., 4.11., and 4.12. respectively. The two most mentioned challenges are "change process" which describes the people within the organization that are either not willing for change or do not have an agile mindset and "cooperation of agile areas with non-agile areas" which describes the challenge that the confrontation of agile units and non-agile units makes collaboration difficult. The two most mentioned barriers are "external influences" which describes the ongoing migration project of the organization and "people unwilling to change" which included people who are not willing to change. Success factors that were most mentioned are "personnel", "coaching training", and "introduction to agile methods".

Research Question 4: What are the lessons learned of the large-scale agile transformation?
In order to answer the last research question, interviewees were asked to list their lessons learned of the large-scale agile transformation. The data was analyzed using a deductive approach in order to identify appropriate categories. A total of 8 lessons learned were identified in this thesis which are listed in Table 4.13. The most mentioned lessons learned are "need for appropriate people" which refers to the fact that people within the organization need to have an agile mindset in order to perform the large-scale agile transformation successfully, "agile transformation requires time" which points out that the large-scale agile transformation is a complex process that needs time, and "management support" which addresses the fact that the management needs to have an agile mindset as well as to fully support the large-scale agile transformation so that it can be performed.

6.2. Outlook

By conducting this case study, four topics were identified that demand for further examination.
First, since the large-scale agile transformation of the German online retailer is still ongoing, further investigation of this transformation process might be of great importance. One should analyze how their transformation process proceeded in the future and whether they have reached their reported target state or whether the actual target state represents the target state of the future. Second, more research should be conducted on the impact of the large-scale agile transformation since this thesis identified impacts not mentioned in other literature. Third, more research should be conducted on challenges and success factors that emerge due to large-scale agile transformation and whether the organization was able to overcome their reported challenges and what additional challenges have arisen that the organization needed to deal with in the future, since new challenges are emerging continuously although there is already a great amount of literature on this topic. Furthermore, research on how to overcome this challenges might be valuable. Fourth, since this thesis provides a framework to measure the agility maturity by means of categories, it would be of great value if more research is conducted on this topic in order to make the large-scale agile transformation measurable and comparable. Although the agility maturity of each category was examined, no general statement could be made as to whether the company can now be described as an agile organization.

A. Appendix

A.1. Structured Interviews

Structured interview questions about the large scale agile transformation.

1. Section: General questions about your company and your role
    a) What is your role description in the company?
    b) How many years have you been working for the company?
    c) What department do you work in?
    d) How long have you had experience with agility?
    e) How would you rate your experience in agility?
2. Section: Questions about agility in the company
    a) How would you assess the maturity of your business agility
    b) How would you assess the agility maturity in your department?
    c) Can you evaluate the categories listed below in terms of their current status? Please also evaluate which condition you think should be achieved.
3. Section: Retrospective
    a) What impact does agile transformation have on the people in the company?
    b) What impact does agile transformation have on the technologies used?
    c) What impact does agile transformation have on tasks?
    d) What impact does agile transformation have on the organization's structure?
    e) Which "lessons learned" have you learned from the agile transformation project in your company so far?
    f) What would you like to change in the future?
    g) What challenges have you encountered so far?
    h) In your opinion, what are the success factors for successfully implementing agile transformation?
4. Section: Outlook

a) What do you see as the challenge of transformation for the organization in the future?

b) Are there barriers that need to be removed before agile transformation can continue?

5. Section: Feedback

a) What wishes/concepts do you have for us?

b) Do you have any comments or open points?

A.2. Structured Interviews Additional Questions

Additional structured interview questions about the large scale agile transformation. This questions were only asked the COO and Agile Coach.

1. Questions about agile transformation
    a) When did the agile transformation start in your company?
    b) In your opinion, what are the goals of agile transformation?
    c) In your opinion, what is the background of agile transformation in the company?
    d) Who initiated the agile transformation?
    e) How was agile transformation initiated?
    f) Which transformation approach was chosen for the introduction?
    g) What are the advantages and disadvantages of the chosen approach?

Bibliography

[1] Pekka Abrahamsson, Kieran Conboy, and Xiaofeng Wang. "'Lots done, more to do': the current state of agile systems development research". In: *European Journal of Information Systems* 18.4 (2009), pp. 281–284. URL: https://link.springer.com/content/pdf/10.1057%2Fejis.2009.27.pdf.

[2] Muhammad Faisal Abrar, Muhammad Sohail Khan, Sikandar Ali, Umar Ali, Muhammad Faran Majeed, Amjad Ali, Bahrul Amin, and Nasir Rasheed. "Motivators for Large-Scale Agile Adoption From Management Perspective: A Systematic Literature Review". In: *IEEE Access* 7 (2019), pp. 22660–22674.

[3] *Agil mit Scrum*. 2018. URL: https://www.hwzdigital.ch/agil-mit-scrum/.

[4] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. *History: The Agile Manifesto*. 9.09.2019. URL: https://agilemanifesto.org/history.html.

[5] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. *Manifesto for Agile Software Development*. 9.09.2019. URL: https://agilemanifesto.org/.

[6] Ilya Bibik. *How to Kill the Scrum Monster*. Berkeley, CA: Apress, 2018.

[7] B. Boehm and R. Turner. "Management Challenges to Implementing Agile Processes in Traditional Development Organizations". In: *IEEE Software* 22.5 (2005), pp. 30–39.

[8] Robert P. Bostrom and J. Stephen Heinen. "MIS Problems and Failures: A Socio-Technical Perpective: Part I: The Causes". In: *MIS Quarterly* 09/1977.Vol.1(3) (1977), pp. 17–32.

[9] Robert P. Bostrom and J. Stephen Heinen. "MIS Problems and Failures: A Socio-Technical Perspective: Part II: The Application of Socio-Technical Theory". In: *MIS Quarterly* 12/1977.Vol.1(4) (1977), pp. 11–28.

[10] Amadeu Silveira Campanelli, Florindo Silote Neto, and Fernando Silva Parreiras. *Assessing Agile Transformation Success Factors*. URL: https://arxiv.org/pdf/1711.04188.pdf.

[11] COLLABNET and VERSIONONE, eds. *13th Annual State of Agile Report*. 5/2019. URL: https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508.

[12] Kieran Conboy, Sharon Coyle, Xiaofeng Wang, and Minna Pikkarainen. "People over Process: Key Challenges in Agile Development". In: *IEEE Software* 28.4 (2011), pp. 48–57.

[13] D. S. Cruzes and T. Dyba. "Recommended Steps for Thematic Synthesis in Software Engineering". In: *2011 International Symposium on Empirical Software Engineering and Measurement*. IEEE, 22.09.2011 - 23.09.2011, pp. 275–284.

[14] Kim Dikert, Maria Paasivaara, and Casper Lassenius. "Challenges and success factors for large-scale agile transformations: A systematic literature review". In: *Journal of Systems and Software* 119 (2016), pp. 87–108.

[15] Torgeir Dingsøyr, Nils Brede Moe, and Eva Amdahl Seim. "Coordinating Knowledge Work in Multiteam Programs". In: *Project Management Journal* 49.6 (2018), pp. 64–77. URL: https://sintef.brage.unit.no/sintef-xmlui/bitstream/handle/11250/2574214/8756972818798980.pdf?sequence=4&isAllowed=y.

[16] Torgeir Dingsøyr, Nils Brede Moe, and Roberto Tonelli, eds. *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers*. Lecture Notes in Business Information Processing. 2014. URL: http://dx.doi.org/10.1007/978-3-319-14358-3.

[17] Torgeir Dingsøyr, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe. "A decade of agile methodologies: Towards explaining agile software development". In: *Journal of Systems and Software* 85.6 (2012), pp. 1213–1221.

[18] T. Dyba and T. Dingsoyr. "What Do We Know about Agile Software Development?" In: *IEEE Software* 26.5 (2009), pp. 6–9.

[19] Christoph Fuchs and Thomas Hess. "Becoming Agile in the Digital Transformation: The Process of a Large-Scale Agile Transformation". In: *Thirty Ninth International Conference on Information Systems*, (2018).

[20] Daniel Gerster, Christian Dremel, and Prashant Kelker. ""Agile Meets Non-Agile": Implications of Adopting Agile Practices at Enterprises". In: (2018).

[21] Riitta Hekkala, Mari-Klara Stein, Matti Rossi, and Kari Smolander. *Hawaii International Conference on System Sciences 2017 (HICSS-50): Hilton Waikoloa Village, Hawaii, January 4-7, 2017*. URL: https://research-api.cbs.dk/ws/portalfiles/portal/57389459/mari_klara_stein_et_al_challenges_in_transitioning_to_an_agile_way_of_working_publishersversion.pdf.

[22] Jan Jöhnk, Maximilian Röglinger, Markus Thimmel, and Nils Urbach. "How to Implement Agile IT Setups: A Taxonomy of Design Options". In: (). URL: https://www.fim-rc.de/Paperbibliothek/Veroeffentlicht/640/wi-640.pdf.

[23] Henrik Kniberg. *No, I didn't invent the Spotify model*. Crisp's Blog, 2015. URL: https://blog.crisp.se/2015/06/07/henrikkniberg/no-i-didnt-invent-the-spotify-model.

[24] Henrik Kniberg. *Spotify engineering culture (part 1)*. 2014. URL: https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/.

[25] Henrik Kniberg. *Spotify engineering culture (part 2)*. 2014. URL: https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/.

[26] Henrik Kniberg and Anders Ivarsson. "Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds". In: (2012). URL: https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf.

[27] Maarit Laanti, Outi Salo, and Pekka Abrahamsson. "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation". In: *Information and Software Technology* 53.3 (2011), pp. 276–290.

[28] Craig Larman and Bas Vodde. "Scaling Agile Development: Large and Multisite Product Development with Large-Scale Scrum". In: *CrossTalk* (2013). URL: http://static1.1.sqspcdn.com/static/f/702523/22609354/1367558447003/201305-Larman.pdf.

[29] Harold J. Leavitt. "Applied Organizational Change in Industry: Structural, Technological and Humanistic Approaches". In: *Handbook of Organizations* (1965), pp. 1144–1170.

[30] Kalle Lyytinen and Mike Newman. "Explaining information systems change: a punctuated socio-technical change model". In: *European Journal of Information Systems* 17.6 (2008), pp. 589–613. URL: https://link-springer-com.eaccess.ub.tum.de/content/pdf/10.1057%2Fejis.2008.50.pdf.

[31] Charles C. Manz and Greg L. Stewart. "Attaining Flexible Stability by Integrating Total Quality Management and Socio-Technical Systems Theory". In: *Organization Science* Vol.8(1) (1997), pp. 59–70.

[32] Peter Measey, Chris Berridge, Alex Gray, Richard Levy, Les Oliver, Barbara Roberts, Michael Short, Darren Wilmshurst, and Lazaro Wolf. *Agile Foundations: Principles, practices and frameworks*. Swindon: BCS Learning & Development Limited, 2015. URL: http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&AN=821933.

[33] Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. "Challenges of migrating to agile methodologies". In: *Communications of the ACM* 48.5 (2005), pp. 72–78.

[34] Maria Paasivaara, Benjamin Behm, Casper Lassenius, and Minna Hallikainen. "Large-scale agile transformation at Ericsson: a case study". In: *Empirical Software Engineering* 23.5 (2018), pp. 2550–2596. URL: https://link-springer-com.eaccess.ub.tum.de/content/pdf/10.1007%2Fs10664-017-9555-8.pdf.

[35] Ken Power. "A Model for Understanding When Scaling Agile Is Appropriate in Large Organizations". In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*. Ed. by Torgeir Dingsøyr, Nils Brede Moe, and Roberto Tonelli. Lecture Notes in Business Information Processing. 2014, pp. 83–92.

[36] Emils Pulmanis. "Project Management Development – Practice and Perspectives". In: (). URL: https://www.balticpmconference.eu/sites/default/files/image-uploads/2016_001-006_Conf_proceeding_2016_ar_vaku_final_22.04.2016%20%281%29.pdf.

[37] Per Runeson and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering". In: *Empirical Software Engineering* 14.2 (2008), pp. 131–164. URL: https://link-springer-com.eaccess.ub.tum.de/content/pdf/10.1007/s10664-008-9102-8.pdf.

[38] Roman Sauter, Werner Sauter, and Roland Wolfig. *Agile Werte- und Kompetenzentwicklung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. URL: https://link-springer-com.eaccess.ub.tum.de/content/pdf/10.1007%2F978-3-662-57305-1.pdf.

[39] Inc. Scaled Agile. *Achieving Business Agility with SAFe® 5.0*. 2019. URL: https://www.scaledagile.com/resources/safe-whitepaper/?_ga=2.10817396.1366921162.1582458174-2123538590.1582458174.

[40] Ken Schwaber and Jeff Sutherland. "The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game". In: (2017). URL: https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100.

[41] Marian STOICA, Marinela MIRCEA, and Bogdan GHILIC-MICU. "Software Development: Agile vs. Traditional". In: *Informatica Economica* 17.4/2013 (2013), pp. 64–76. URL: http://www.revistaie.ase.ro/content/68/06%20-%20Stoica,%20Mircea,%20Ghilic.pdf.

[42] Jeff Sutherland. *Jeff Sutherland's answer to Were Agile and Scrum developed separately, and do they overlap or complement each other? - Quora*. 15.3.2020. URL: https://www.quora.com/Were-Agile-and-Scrum-developed-separately-and-do-they-overlap-or-complement-each-other/answer/Jeff-Sutherland-13?srid=uAHih.

[43] Jan Vom Brocke, Alexander Simons, Bjoern Niehaves, Kai Reimer, Ralf Plattfaut, and Anne Cleven. "Reconstructuring the giant: On the Impotance of rigour in documenting the literature search process". In: *ECIS 2009 Proceedings* 161 (2009). URL: https://aisel.aisnet.org/ecis2009/161.

[44] James P. Womack, Daniel T. Jones, and Daniel Roos. *The machine that changed the world*. 1990.

[45] Robert K. Yin. *Case study research and applications: Design and methods*. Sixth edition. Los Angeles et al.: SAGE, 2018.