CHALMERS

Agile into Traditional Organizations and Hardware Development

Master of Science Thesis in the Programs Production Engineering and Quality and Operations Management

SOFIA HAGLUND
CAROLINE SVENSSON

Department of Technology Management and Economics
Division of Innovation and R&D Management
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sverige 2016
Report No. E 2016:059

MASTER'S THESIS E 2016:059

Agile into Traditional Organizations and Hardware Development

SOFIA HAGLUND
CAROLINE SVENSSON

Tutor, Chalmers: Lars Trygg
Tutor, Company: Eric Dutt

Department of Technology Management and Economics
Division of Innovation and R&D Management
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

Agile into Traditional Organizations and Hardware Development

SOFIA. HAGLUND
CAROLINE SVENSSON

Master's Thesis E 2016: 059

Department of Technology Management and Economics
Division of Innovation and R&D Management
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone: + 46 (0)31-772 1000

Cover:
[The picture describe the many different phrases that circulate around when working with agile software development (Vizteams, 2013)]

Chalmers Reproservice
Göteborg, Sweden 2016

Abstract

This master thesis presents a qualitative study conducted at Volvo Group Trucks Technology (GTT). GTT is a global leading automotive company, which is the second largest producer of trucks and transportation solutions within the commercial vehicle industry. This study has been done at Driver Electronics (DE) development department within Electrical & Electronics Engineering.

The field of product development is growing more competitive because of an increasing speed of changing customer requirements and technologies. In order to stay competitive it is important to have a development process that can adjust to those changing requirements in an effective and efficient way. Consequently, Driver Electronics (DE) introduced the agile methodology about two years ago, which is a short iterative development process allowing high flexibility and fast feedback-loops.

Almost no investigations regarding the new way of working have been done since agile was introduced at the department. A small investigation was made and found that the main unexplored area was working with agile in a small part of the company whilst the rest of the organization followed traditional development and waterfall model.

The purpose of this thesis is to study what challenges emerge when agile is introduced in a traditional organization and how they can be managed. The study perspective is on processes at DE where the focus is on products that are inside the cabin of the truck and close to the end user. The products consist of both hardware and software components, which have significantly different development cycles.

This created an interest to investigate in possibilities to work with agile in hardware development as well, which is the second purpose in this thesis.

The research is a qualitative study, consisting of interviews at GTT and benchmarking studies at Saab Surveillance and Ericsson. Both the benchmarking companies have a similar organizational structure and products of both software and hardware, which provided a good comparison. As part of this study, two workshops were also conducted. An internal workshop was conducted with employees from GTT where the attendees had different perspectives and a second workshop with managers from Saab Surveillance, Ericsson and GTT.

The result is presented as eight main challenges that emerge when agile is implemented in a traditional organization together with suggested solutions. One of the main things is to get supportive managers to get an agile distribution to other departments and get consisting teams in order to create self-organized teams. A main fact in an agile implementation is to understand the change management process, which is not negligible in this case either, for example a mindset change takes time. This thesis also presents several long-term recommendations as well as a specific “improvement tool”, which is used at Ericsson today.

Keywords: agile, lean product development, agile in hardware development, agile V.S traditional organization, lean production, organizational structure, traditional product development, resource and flow efficiency, cross-functional teams, agile practices, agile principles, scrum, tools of lean.

Acknowledgements

This thesis work is our final step in completing our respective master's programs at Chalmers University for Technology. It is the first research done in connection with the agile implementation at Driver Electronics at Volvo Group Trucks Technology. We also hope that it will contribute to the research in agile methodology and provide inspiration for continued work at Chalmers University of Technology.

A special thanks to our company Volvo Group Trucks Technology for letting us conduct this thesis and to the group manager Jenny Olsson and our company supervisor Eric Dutt for giving us continuous support and consideration throughout this thesis.

At Chalmers University of Technology we would like to extend special and grateful thanks to our supervisor Lars Trygg who has been a warm and kindly support, believing in us throughout the whole study.

Additionally, we would like to thank to our benchmarking companies; Saab Surveillance and Ericsson that have given us great insight and inspiration from their agile and lean transformation work. We want to send the warmest thanks for the friendly reception at the respective companies by each company, Thomas Riddarstråle, Madelene Lundqvist and Petter Björk.

Also, a dedicated thanks to all the people we have been interviewed during the thesis time and overall the people that have given us their time during the whole project time.

Last but not least, we would like to say that we are grateful to all the friendly and inspiring people that we have met along the way, and thank them for making this an enjoyable time for us.

Sofia Haglund & Caroline Svensson, June 2016

Table of Contents

1 Introduction 1
1.1 Background 1
1.2 Purpose 3
1.3 Company description 3
1.4 The agile introduction at DE 4
1.5 Problem description and research questions 5
1.6 Delimitations 6

2 Theoretical framework 7
2.1 Traditional Product Development 7
2.1.1 Managing Product Development 7
2.1.2 Organizational structures 9
2.1.3 Development approaches 12
2.2 Lean 16
2.2.1 Lean Production 16
2.2.2 Lean product development 16
2.2.3 Resource and flow efficiency 17
2.2.4 Tools of Lean 18
2.3 Agile 19
2.3.1 History of Agile 19
2.3.2 What is Agile? 19
2.3.3 The Agile Manifesto 21
2.3.4 Agile Principles 22
2.3.5 Agile Practices 24
2.3.6 Agile vs. traditional 28

3 Methodology 35
3.1 Methodology description 35
3.2 The case study 37
3.3 Literature study 38
3.4 Interviews 39
3.4.1 Different interview structures 39
3.4.2 Face to face interview and phone interviews 40
3.4.3 Documenting the interviews 41
3.5 Observation 41
3.6 Benchmarking 42
3.6.1 Saab Surveillance 42
3.6.2 Ericsson 44
3.7 Workshop 46
3.8 Data analysis 47
3.9 Reliability and validity 47
3.9.1 Reliability 47
3.9.2 Validity 48
3.10 Ethical consideration 48

4 Empirical data 51
4.1 Analysis of the empirical data 51
4.1.1 Concluding findings from the interviews 69
5 Analysis 71
5.1 Analyzing research question one 71
5.2 Analyzing research question two 78
5.2.1 Concluding analysis of research question two 84
6 Conclusion 85
6.1 Concluding research question one 85
6.2 Concluding research question two 87
7 Discussion 90
7.1 Recommendations 91
7.2 Suggestions for future research 92
8 References 93
Appendix A - Interviewees 101
A:1 Internal interviewed people 101
A:2 Benchmarking companies attendants 101
A:3 External Workshop attendants 101
A: 4 Internal Workshop attendants 102
Appendix B - Interview template for GTT 103
Appendix C - Interview template for Ericsson 105
Appendix D - Internal Workshop 108

List of Figures

2.1 The waterfall model with four phases 14
2.2 Stage-gate model 15
2.3 The effectiveness matrix 18
2.4 Four sprint iterations 20

1 Introduction

In this chapter, the aim is to frame the research area by presenting the case background, to define and clarify the thesis content by stating the purpose, to pose the research questions that will be addressed, and to state the delimitations of the work.

1.1 Background

The field of product development has been growing more competitive and at an increasingly rate for the past half a decade. The dramatically changing technologies and market requirements demand an ever-increasing faster time to market (Highsmith and Cockburn, 2001; Cooper, 2016). For a product development company this means that the customers have increased our demands and they actually have higher requirements of the products that they buy today. Technology may change so quickly that what I want today may be different than what I might want tomorrow. This is something that the developers are constantly dealing with, i.e. changing customer requirements.

When talk about the development of products such as trucks, where the industry norm is a 4-6 years development cycle and a much shorter technology life cycles for certain areas, it is a challenge to develop products with relatively long lead time to market. In order to stay competitive and to achieve organizational growth, organizations must stay close to the market trends and constantly meet the changing requirements and a development process that can manage the uncertainty and allow adjustments for unexpected and late changes. The ability to live with uncertainty and manage it, is central in the agile mindset. This is one big reason why agile development has been adopted by several organizations.

In contrary, the majorities of the traditional development organizations today develop and design according to a traditional sequential model properly defined and phases are elaborated by specifying the needed input and output parameters in an early phase. Common traditional models are the waterfall and the stage-gate model, both of which call for the completion of a development or design stage before moving on to the next stage in the process. This approach has a major drawback in that it does not allow developers and designers to easily go back to a previous stage or to adapt to unexpected changes without compromising project timing and / or causing issues with project resource allocations.

The traditional model also demands that the product requirements are established at the initiation of the project or at least at the initiation of a specific phase of the project. Since the development time for a product can range up to a few years, it would mean that the product requirements have to be frozen many *years* before the product is released to the market. By then the customer demands and the market trends may have changed. It goes without saying that a traditional development model allows limited space for dynamic adaptation to new or

changing customer requirements or to counteract the competition. Another dilemma is that this situation causes difficulties in planning for possible challenges and efforts that might appear along the way. This is perceived as a complicated and uncertain process, which puts a lot of pressure on companies, and today many companies are dealing with this dilemma. Verganti (1999) discusses the importance of planned flexibility, *"the capability to build flexibility into the development process"* (Verganti, 1999, p. 365), linking reaction and anticipation capabilities into product development. In the automotive industry, flexibility is highly desired, since it is required that the products have a long duration in the market -- in order to correspond to the market, for example, questions asked is "what gadgets are asked for in five years?". (this sentence is too long)

In the 1970's the traditional sequential model was criticized by Dr. Winston Royce who claimed that software is not aimed to be developed on an assembly line (like automobile manufacturing), where one piece at the time is developed sequentially (Agile Methodology, 2008). Problems emerged in the IT development that the waterfall and stage-gate processes could not handle because traditional processes focused on big and long-term goals where a final product, including all features, should be decided years before launch. This is not appropriate for IT development since the features and criteria typically changed within a 12-18 month period (Cooper, 2016).

In the late 1990s and early 2000s agile software development methodologies were introduced, methodologies based on different software methods (Cooper, 2016). The agile methodology is an iterative and incremental process, consisting of several principles and practices/methods, where for instance flexibility and short feedback loops are important principles, in order to satisfy the customer. One of the popular agile methods is scrum.

Scrum software development is a definition of dedicated full-time, cross-functional teams, focusing on developing functional software in a short time period to get rapidly feedback (Cooper and Sommer, 2016). Nowadays, the agile methodology development methodology is gaining in popularity, especially for software development. More and more traditional organizations are shifting from the traditional models for the purpose of achieving increased quality, new feature adaptation, as well as cost and time savings (Popli and Chauhan. 2013). Companies have also started to integrate agile methods into their stage-gate process for the purpose of developing hardware products (Cooper, 2016). The trend began when it was found out that stage-gate and agile methods was a complement to each other.

Challenges however, have emerged when implementing agile methods into traditional development organizations due to the different ways of working – this is the case at GTT.

One of the departments that adopted agile methods was Driver Electronics (DE), part of Electrical & Electronics Engineering (EE & E). Their group develops hardware and software components placed mainly in the cab. DE adopted parts of the scrum methodology in 2012 in the software development for one of their projects -- scrum teams are now being used for their second project.

Since the agile introduction at DE, almost no investigation has been conducted to find out what results the agile way of working have provided. Currently, several existing questions are unanswered at DE, which are for instance: what perceptions do the employees have regarding the agile way of working? What has been going well and what challenges have emerged? How can we improve our "agile way of working" before initiating a new agile project? This thesis was commissioned to try to find answers to those questions. The focus is on investigating what happens when parts of the agile methodology are introduced into a traditional organization? The main problem with the new way of working tried to be identified in the beginning of the thesis in order to create a scope, but quite soon it was realized that the challenges are many.

1.2 Purpose

When agile development is introduced into small parts of a traditional led organization, challenges emerge due to the radically different ways of working in the different types of development compared to traditional development. Agile development focuses on development in short iterations and to plan in short-term perspective, which is just opposite to that of traditional development. This thesis work, will help to identify what happens when agile development is introduced into a small part of an organization whilst the rest of the organization continues following traditional development approaches. Worth mentioning is that the outcome of using agile methods has provided positive results in both the development process and in the "way of working" for software based development. Therefore, it will be investigated if similar agile methods that are used for software development are feasible to adapt into certain hardware development processes as well.

1.3 Company description

GTT (Group Trucks Technology) belongs to the Volvo Group AB concern and is a traditional matrix organization, where projects are going in parallel with different line organizations. GTT consists of several line organizations and departments where Driver Electronics (DE) is one department within Electrical & Electronics Engineering (EE&E).

At E&EE electronic platforms are developed, with the capability to satisfy both high feature demands from their brand customers as well as utilizing their scale effects to reach high component volumes. Further, DE is responsible for component development in the cabin and maintenance, in close co-operation with E&EE, product sites, segment architects, and cross-functional technology strategies in the areas of Security, Drive Time Management, Infotainment and Driver Interaction (this sentence is way to long). For these areas DE is also responsible to secure that there are technical solutions available that fit the whole range or target products, from emerging markets to premium high-end market segments. The components consist of both software and hardware and are mainly placed inside the cabin of the truck within the area of HMI (Human Machine Interface).

In the DE group the employees are split into groups with one (group) manager each. These managers belong to the line organization, which is more or less a chain of command organization that is built in a pyramid structure. The Chief Executive Officer (CEO) is at the top cascading down to the group or team managers. The project organization is also built based on a pyramid structure with the complete project level project manager (PM) at the top cascading down to the department's' PMs, sub system PMs and component level PMs.
Basically, it is the not the project but the line organization that owns the product, the technology connected to the products and the competences. When a project is set up, it is the line managers that have to provide the required resources to the projects (e.g. engineers, designers, programmers, project managers etc.) in order to achieve the project goals.

The entire GTT project plan is broken up into a time line that is split into periods of eight weeks, each period is called a baseline. At the DE department, the baselines are split up into four sections to match the two-weeks-sprints used by the agile teams. The definition of project time plan gives the lines management, the projects and the GTT organization as a whole, a common view of the time line in terms of project planning.

1.4 The agile introduction at DE

Before agile development was introduced into DE, the actual development of the software was outsourced to external suppliers. Only specifications were developed in-house. This not only created problems in communication but also long lead-times from specification to error identification and correction. Specifications had no conflict support, so as the project progressed and the amount of specification documents increased, the harder it was to spot errors during the specification phase. DE took over in-house software development to alleviate these problems. Alone the way they introduced agile practices and methods into their traditional stage-gate process, adopted forms of the scrum methodology to suite their specific software development needs as well as the organization. Scrum was combined with lean tools such as a kanban board, which is a visualized tool for planning and prioritization. So far, the agile introduction has been provided positive results according to employees at DE. For instance, the development speed; the possibilities to adapt to changing requirements; and the motivation among the developers have been increased.

The introduction to agile started by training a few permanent (and motivated) employees become scrum masters/coaches, though education mentioned little about the agile methodology itself.
The introduction was managed by an external agile coach for about two years initially. The coach´s main task was to teach the employees about the agile methodology, as well as to identify where in the software development process they could adapt parts of the agile principles. The coach found out that using the scrum methodology was the most suitable way of working at DE. Therefore it was identified how scrum teams could be created and what competences that were needed within the teams.

Today year 2016 there are three scrum teams with six to ten members in each of the team, sitting close to each other in an open landscape area. One of the three teams has designers, programmers, specifications performer and validators. The teams work in two weeks sprints with the tasks prioritized by the product owner in JIRA as well as with a kanban board.

In hardware development the traditional waterfall methodology combined with a gating process with different gates/deadlines is followed and has been used for many years. This work is outsourced based on Volvo's requirements.

1.5 Problem description and research questions

Since introducing agile there have been many challenges in coordinating this new methodology with the pre-existing stage-gate processes. This implies that the agile teams must adapt to this gating process with its fixed time constraints etc. and this affects their ability to effectively use the agile methodology. The agile introduction at DE has resulted in difficulties in cooperation among different departments at GTT. The difficulties are for instance due to the different ways of planning, communicating, documenting and getting from customer feedback. At present, the origin and nature of the challenges are not yet fully identified nor researched--, this leads to the first research question:

- *RQ1: How to manage the challenges that emerge when agile development is introduced into a traditional development organization?*

The pre-study executed at the beginning of this thesis indicated that the agile teams are influenced by a part of the organization that was still working according to the traditional development and due to limited understanding of the agile methodology itself. One problem for instance was that the specification of the product requirements in hardware development are defined early, limiting the software developers to fully make use of the agile methodology.

As previously mentioned, the components developed at DE consisted of both hardware and software-- collaboration between the two units was becoming increasingly important. One thought from the authors was that the collaboration might increase if the hardware development adopts parts of the agile methodology as well.

According to Cooper (2016), companies have already started to integrate agile methods into their traditional gating processes for the purpose of developing physical products, which has provided good results: faster product releases, improved response to changing requirements, and improved communication and moral between employees. Also, due to increased competition in product development. The need for faster time-to-market and flexibility in the hardware development process was necessary. The arguments mentioned above, indicate that

investigation about agile and hardware development is an interesting research area, but unfortunately, this research area is currently not well covered.

The agile methodology is closely related to *lean*, which is a well-known philosophy based on the Toyota's Production System (TPS). The focus in TPS is the customer, continuous improvement and quality through value adding activities and elimination of waste, reducing the lead-time, cutting costs, and increase the quality in production (Liker and Morgan, 2006). The lean approach has been applied into many other areas, such as health care, insurance industry, as well as product development. In order to benefit from lean in product development it is important to adapt the lean principles into the specific area and not just copy the *Toyota Way*. To adapt these principles it is important to have the right mindset when applying the agile methodology into areas other than software development. This leads to the second research question:

- *RQ2: What parts of the agile methodology can beneficially be used in the hardware development process?*

1.6 Delimitations

This thesis is conducted at Volvo Group Trucks Technology and is located at a development department (DE) in Gothenburg, Sweden. The data is thereby collected mainly at the department, where the agile teams are observed. The findings are not compared with other agile teams from the organization or agile teams in other countries.

During the thesis work, the components developed at DE and the ongoing projects are studied in order to relate the interviewees opinions about agile to reality. However, due to confidentiality issues at GTT, the components and projects will not be presented in this report.

The goal of this report is to lift the discussion of the research area in a more generalized level, therefore the execution of this thesis at DE is aimed to get knowledge about real situations in order to answer the research questions. The findings are also generalized in order to hopefully contribute to knowledge in other organizations similar to DE. Though, the founded answers for the research questions might differ depending on the organization.

The theory regarding agile in hardware development is limited. Therefore the focus in this thesis is only to look into the possibilities of adapting agile and hardware development, based on the theory that was found, and combine this with what the interviewers believe are possible.

2 Theoretical framework

In this chapter the theory presented aims to give prerequisite knowledge in the areas that the analysis of the research questions will touch. The presented theory is traditional product development, lean and agile.

2.1 Traditional Product Development

The goal of all product and process development projects is to take an idea and make it a real product to meet the needs of the market in an economical and manufacturerable form (Wheelwright and Clark, 1992). The general development process is a set of activities that runs over a period of time. The process starts with several ideas that are successively refined and selected, thereafter different development projects are created which can later be pushed to rapid completion and introduction (Wheelwright and Clark, 1992). In traditional product and process development the process is normally executed in traditional ways. This has brought some problems and challenges during the years, which are highlighted in this section. Some traditional organizational structures and development approaches are also presented.

2.1.1 Managing Product Development

Managing a product development process in an integrated perspective has been a huge challenge for managers (Wheelwright and Clark, 1992; Verganti, 1999). The competition is increasing and the product requirements and the technology is changing in a dramatically increasing pace, which in turn require a shorter time-to market and the competition is constantly increasing (Cao and Ramesh, 2007). In order to deal with turbulent environment a reactive and proactive flexibility is needed (Brown and Bessant, 2003).

Many researchers have emphasized different problems and uncertainties when managing product development. According to Wheelwright and Clark (1992) challenges emerge due to different reasons:, One reason is mismatches between the organization and the external environment. This means that the changing technology is often missed and decisions were made about the technology before it is stable on the market. Another reason is mismatches among functions (functions are units where people with related skills are grouped together) due to lack of communication between the functions or by only adopting the sequential process approach *"throwing over the wall"*, which is a typical phenomenon in functional organizational structures. The last reason is that making decisions early in the process lead to companies getting looked into one concept too early as well as leaving little or no room in the time plan for unexpected requirement changes.

A major part of traditional product development documentation bringing both positive and negative effects. According to Polpi et al. (2013) it provides knowledge sharing, but Boehm (1988) discusses the difficulties of using document driven development approaches. The

primary source of its difficulty is the emphasis on *completely* elaborated documentation, which is a criterion for setting the early requirements. He claims that this is not the most appropriate way in all cases because requirements decided early might have to change at the product launch.

The early phase in product and process development however, is what many researchers are concerned about and see as the most crucial phase (Gupta, and Wilemon, 1990; Rosenthal; 1992; Wheelwright and Clark, 1992; Clark and Wheelwright, 1993; Hart, 1993; Bacon et al. 1994; Cooper, 1994; Cooper and Kleinschmidt, 1994; Brown and Eisenhardt, 1995; Hayes et al. 1988; Khurana and Rosenthal 1988), Verganti (1999) did research in how to manage the early phase in order to reach a high performance product development process (Verganti, 1999). In the early phase several tasks are included such as requirement analysis, concept generation, product planning and product definition. This means that the company first develops a product concept based on requirements, defines the specification and architecture of the product and finally a plan is made up. Early decisions had the highest influence of the project outcome since they cannot be changed in downstream phases. Nerur et al., (2005) emphasize the importance of having close collaborartion with customers in the early phase when the specifications are development, but he also stated that this is considered to be less prioritized in organizations.

Verganti (1999) suggests that the early phase is complicated because the decisions consider possible constraints and opportunities that might emerge in downstream phases, which is usual in fast changing industries. The problems emerge since information about constraints and opportunities are not shared with the downstream phases before the implementation phase (Iansiti, 1993; Eisenhardt and Tabrizi, 1996; Iansiti, 1996; Verganti and Iansiti, 1998). The consequences of undiscovered constraints and opportunities result in late engineering changes, decreased product quality, high costs and time-delays (Verganti, 1999). For instance, product specifications must be consistent with the manufacturing capability, and new upcoming technologies must be able to be incorporated to the product architecture during the design phase. By the way, the “new technologies” may not have been invented in the early phases of the project.

Verganti (1999) also conducted a study to investigate different approaches to manage the early phase in product development. He analyzed how anticipation and reaction are interrelated in the different approaches and the link between these two terms. This is called planned flexibility, i.e., building flexibility into the product development process by making early analyses of the product. According to Verganti (1999) the only way to manage product development is through *anticipation,* feed-forward planning (Verganti, 1977; Verganti, 1977) or front loading (Fujimoto, 1977). Basically this means that companies using their capabilities of anticipating information as early as possible, focusing on a plan for reaction or feedback planning in order to be responsive to late changes. The early decisions cover market requirements, design feasibility, usability, reliability, manufacturability etc. In order to get the information needed from downstream phases to anticipate for future upcoming constraints and opportunities it requires great effort and management support as well as early

involvement of manufacturing (Trygg, 1991; Jürgens, 1995; Ettlie, 1996; Haddad, 1997) or suppliers (Dyer and Ouchi, 1993; Liker et al., 1996).

Dealing with early decisions is what many researchers have stated as the "knowledge-gap", indicating that decisions are made before having the required information and knowledge. This is one reason why Nyman (2010) talks about the importance of taking decisions when the level of knowledge has increased systematically during the project. This is product development uncertainty.

According to Galbraith (1977, p. 21) *"the greater the uncertainty of the task, the greater the amount of information that has to be processed between decision makers during the execution of the task"* -- as long as the tasks are well understood before execution, the planning becomes much easier and feasible. Alternatively, if the task is not well understood, more knowledge is needed during the actual execution and changes might be needed in e.g. resource allocation, plans and priorities. This means that these changes require information processing during the task execution. Galbraith basically says to limit the ability of the company to plan or to make decisions about tasks before the actual execution. Nyman (2010) also discusses *planning at a higher level* since the amount of knowledge is low at the *beginning* of the project as well as possible changes emerging *during* the project-- planning in detail is not an effective way of working. Galbraith (1977) further states that either the company must increase its ability to plan before execution and the ability to adapt to this, or to decrease the level of performance continued viability. Nyman (2010) also discusses the high project uncertainty that makes it difficult to state clear goals and clear limitations before the project has even started. He further states that projects that ignore this, tend to be late and spend more resources than what was initially planned.

Actually, the above mentioned is partly what Engwall (2002) discussed as well, but rather than the *challenges of managing projects* and he pointed out three reasons behind project failures: 1) projects fail due to management deficiencies such as incomplete planning, lack of coordination, non-appropriate technical solutions; 2) projects fail due to environmental factors, i.e. if not enough resources, being opposed with stakeholders or if other projects have higher priority; and 3) due to the project goals, i.e. deficient contracts, poorly defined assignments and specifications, or goals that change frequently throughout the project. The argument is that projects should not be initiated without the sufficient preparation, research or planning, which is usually the case. This criticism is nothing new, but still managers have difficulties of managing these projects. Furthermore, he states that one must accept that project goals cannot be more than just a qualified guess about future happenings. (Engwall, 2002).

2.1.2 Organizational structures

A Product development organization is a layout of people with different competences being linked together into groups where the links are formal or informal (Ulrich and Eppinger, 2008). They can be linked by a physical layout or linked by only communication and

remotely located. According to Ulrich and Eppinger (2008) the strongest organization structure are those who integrate performance evaluation, budgets, and other resource allocations. Some alternative organizational structures are presented next, commonly used within traditional organizations. Those that are presented are the following: *functional organization, project organization, lightweight organization, heavyweight organization, and cross-functional organization.*

2.1.2.1 Functional Organization

Functional structures are often found in larger, traditional organizations (Wheelwright and Clark, 1992). In this structure the link is between people with related skills and grouped together in specialized units called functions (lines). Each function works under direction to a sub function manager who does all the administrative work (such as evaluate the team members performance, pay their salaries, handle the budget related questions etc.) (Wheelwright and Clark, 1992; Ulrich and Eppinger, 2008). Generally the functions in product development companies are marketing, design and manufacturing etc. (Ulrich and Eppinger, 2008). The activities within each function are: the engineers develop the hardware and software and execute tests, the marketing keeps contact with the customer and the manufacturing producing the product (Wheelwright and Clark, 1992). The meaning of the functional structure is to put all resources needed for one single activity close together to maximize knowledge sharing (Palhan, 2007). Within each function, only the work related to the specific function is done and when the work is finished the responsibility is passed to the next function. This responsibility transfer is commonly known *"throwing it over the wall"* (Wheelwright and Clark, 1992, p 192). This organizational structure brings both advantages and disadvantages. The advantages are that all knowledge required for executing the work is available in a short distance and the group easily benefits from prior experiences. Also the managers can easily control both the resources and the performance in the project (Wheelwright and Clark, 1992; Maylor, 2010). However, as Maylor (2010) and Wheelwright and Clark (1992) state, this structure is relatively slow because of limited communication and coordination across the different functions which is, according to Wheelwright and Clark (1992), because of the different functions not usually sitting co-located. Allen (1970) stated that physical proximity is a key element to improving communications and interactions between functions.

2.1.2.2 Project Organization

In *project organization* the link is instead among the people who are working on the same project where a project is seen as a set of activities (Ulrich and Eppinger, 2008). Compared to a functional organization where people with related skills are sitting together, a project organization consisting of a group of people from different functions. Each project team reports to a project manager instead of functional managers, and the team members should preferably sit in the same building in order to easily share information (Ulrich and Eppinger, 2008). Usually in traditional organizations, everyone in the different functions belong to the project in some way or another.

2.1.2.3 Matrix organization

When project and functional organizations are combined you get the *matrix organization*. Here the link between people are both to those within the same function as well as those who are working on the same project, which means that the team members report typically to both the functional managers as well as to the project manager (Ulrich and Eppinger, 2008). According to the authors, either the functional manager or the project manager takes more or less responsibility over the team members, since both the managers cannot take responsibility for the administrative work. However, three different matrix structures are identified: the heavyweight project organization, the lightweight organization structure (Clark and Fujimoto, 1991; Wheelwright and Clark, 1992; Ulrich and Eppinger, 2008; Maylor, 2010), and the autonomous project organization (Wheelwright and Clark, 1992), all are various forms of cross-functional teams.

In the *heavyweight project organization* the strongest link is to the project, where the project manager takes all authority over the project team and the functional manager has less authority. This means that the project manager does all the administration work for the project team members even though they belong to different functions. Contrary, the *lightweight project organization*, the strongest link is to the function and less to the project, which means that the functional manager has the strongest authority over the team members. The project managers acts as a facilitator and execute other tasks such as organizing meetings and set up plans etc. (Ulrich and Eppinger, 2008; Maylor, 2010). The last structure is the *autonomous project organization,* where a dedicated heavyweight project team is moved from the function in order to be co-located and only keep focus for a single project (Wheelwright and Clark, 1992; Ulrich and Eppinger, 2008).

2.1.2.4 Cross-functional integration

In projects and matrix organizations working in teams are more important than in pure functional organizations (Ulrich and Eppinger, 2008). Also, in order to stay competitive at the market Wheelwright and Clark (1992) emphasize the importance of creating integration across functions, at the right time and in an effective way. This is what Wheelwright and Clark (1992) call *effective cross-functional integration* and Clark and Fujimoto (1991) call *integrated problem solvin*g – this is the trend in the latest years instead of working in pure functional groups. Outstanding products that provide both high quality and value to the customer is required, which rapidly and efficiently should be developed to the marketplace (Wheelwright and Clark, 1992). It has been discussioned that quality and delivery speed is a trade-off, but Clark and Fujimoto (1991) have argued that this is not the case, since evidence shows that improved quality and faster development process can be achieved by adopting integrated problem solving.

However, cross-functional integration is not required in all types of projects (Wheelwright and Clark, 1992). For example when the design is stable, the requirements may not change so much, the interfaces are clear between the functions, they know what activities are within their scope and the lead times are long, then functional groups might develop effectively without integration across the functions (Wheelwright and Clark, 1992). Instead, cross-functional teams are required when the market is more uncertain - when the market requirements and the technologies are changing faster and the time-to-market is getting more crucial (Wheelwright and Clark, 1992).

Integration among functions brings lots of advantages, especially if some functions are integrated in an early phase; e.g. an integrated design can be achieved earlier if developers integrate with manufacturing since information is easier shared about what is feasible or not. Also, early supplier involvement provides knowledge and insight of the product, helping to avoid eventually upcoming problems. Integration with marketing provides better customer relationsas the customer will influence the direction of where to go in the project. (Wheelwright and Clark, 1992).

2.1.3 Development approaches

In the traditional development approaches, the general development process is broken down into phases from idea to launch (Cooper, 2011), and executed in sequence. The approaches can be referred to an informal (Nerur et al., 2005) “plan-driven” or “task based” approach, where the goals are to achieve *predictability, stability* and *high assurance* (Boehm and Turner, 2004) and the focus is on realizing optimized and repeatable processes (Nerur et al, 2005). The goals are supported by plans, documentation, working products and verification and validation strategies. In the planning, the focus is on up-front planning for big and long-term goals (Cooper, 2016). The purpose with up-front planning is for predicting, measuring and controlling possible problems and variations that might occur during the product life

cycle and further able to identify these (Highsmith and Cockburn, 2001). The traditional approaches brings some drawbacks: For instance, when late changes are discovered it is difficult or even impossible to go back to earlier phases; estimation of time and budget is done early, which is a difficult task to execute before the project has even started (Balaji & Murugaiyan 2012; Stoica et al. 2013); and also that the final product including all its features and requirements, should be decided early-- according to Cooper (2016) and Boehm (1988) -- not the most appropriate way for all products due to product requirements are changing rapidly, especially in IT related products. A consequence of this is that the features and requirements decided when the project initially was planned will probably have changed when the product is going to be released. Reagan (2012) in Cooper (2016, p. 22) states that *" it's hard to alter course when you're being swept down a large waterfall...Too much up-front planning means too much change management downstream"*. Reagan (2012) means that making early decisions about product features and plans leads to later compromises, which actually symbolizes the traditional approaches: e*arly decisions to major features, long time-plans, long feedback-loops and much re-planning which results in inefficiencies* and *slow development processes* (Cooper, 2016). Two models, *waterfall* and *stage gate*, are presented below.

2.1.3.1 The waterfall model

The waterfall model is a linear sequential model that was first proposed by W. W. Royce who recommended an iterative development process in the 1970s, in which became the basis of framework for software development (Popli et al., 2013; Stoica et al., 2013), but the framework is commonly used in many traditional development organizations. The model is easy to understand and provides knowledge sharing, each phase must be completed and frozen before going on to the next; each phase proceeds without any overlapping; and finally, each phase must be completed within the specific time period. In the end of each phase the process is documented, since it helps maintaining the quality of the project (Balaji and Murugaiyan, 2012). The waterfall model includes phases of requirements analysis, design implementation, test, integration and maintenance (Popli et al., 2013; Brassil, 2012; Petersen and Baca, 2009). Royce (1970) who was the founder of the model stated that the waterfall model is a risky process, and invites failure because of the high extent of early work that must be done. In Figure 2.1, the product requirements are decided early in the process -- the requirements cannot easily be changed in the next phase. However, the waterfall model is recommended for small projects with requirements that are clearly understood in an early phase, but not recommended for complex projects with high uncertainties (Balaji & Murugaiyan 2012; Stoica et al. 2013). Four phases of a simplified waterfall model are described next:

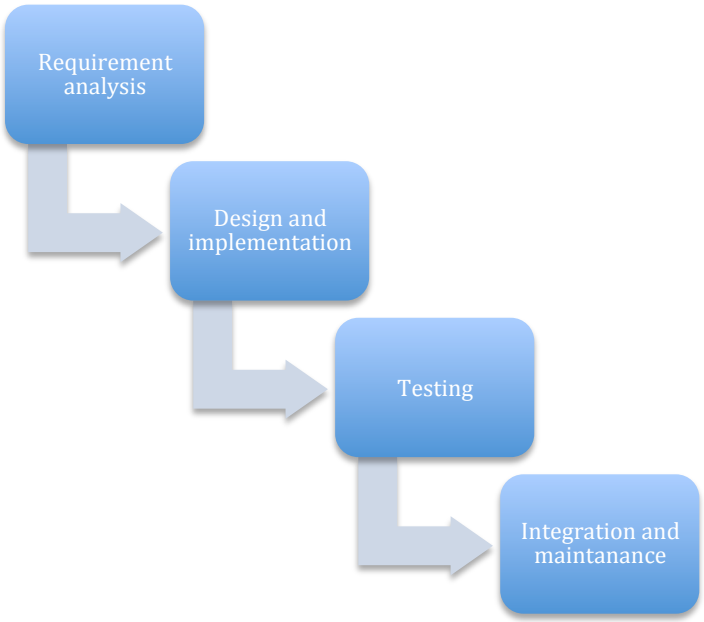

FIGURE 2.1: THE WATERFALL MODEL WITH FOUR PHASES

Requirement analysis: A comprehensive description of what the product should include based on the customer needs. Here the functional and nonfunctional requirements are defined and analyzed for use as inputs to the design and implementation phase. (Brassil, 2012; Petersen and Baca, 2009).

Design and Implementation: In this phase, the focus is on planning and problem solving for the product solution. Both designers and software developers are involved in this phase to develop a functional product that includes the customer needs and requirements. The implementation phase includes realization of the requirements and design specification e.g. in a visual program. (Brassil, 2012)

Testing: The testing phase is also known as the verification and validation phase. Here the product is tested regarding quality and functionality to make sure it meets the earlier defined requirements and specifications. (Brassil, 2012; Petersen and Baca, 2009)

Integration and Maintenance: In the last phase, a product solution is modified based on delivery and deployment to refine output, errors and to improve performance as well as quality (Brassil, 2012). The product must be maintained after it has been delivered to the customer, e.g. if the customer detects problems, packages for updating the product must be delivered (Petersen & Baca, 2009).

2.1.3.2 The stage-gate model

The stage-gate model was first illustrated by Robert G. Cooper with the goal of developing an effective model to manage, direct and control the development process, basically to reduce lead-times (see Figure 2.2). Cooper (1990) describes the stage-gate system as a conceptual and operational model for developing a product from idea to launch (Boehm, 1988; Cooper, 2016; Karlström and Runeson, 2005) as well as for improving the effectiveness and efficiency of managing development projects. The model, he states, is quite easy to follow. The stage-gate model consists of several stages and gates in sequence and each stage-gate seeks to increase the work efficiency (Veryzer, 1998) and to create the best practice (Cooper, 2016). Depending on the business the number of stages and gates differ. However, within each stage different tasks must be executed and each stage is followed with a gate, where decisions are made (Karlström & Runeson, 2005; Cooper, 1990; Nyman, 2010; Cooper, 2016). The gates function as a quality control checkpoint where the top management and project sponsor controls the projects and decide whether or not to continue funding the project (Karlström & Runeson, 2005; Cooper, 1990; Cooper, 2016). The gates facilitate the decisions of what projects to do and what activities to do within each project (Cooper, 2016). The gates are characterized with a set of inputs, criteria, and outputs: The inputs are deliverables that must be delivered by the project leader; the criteria are the aspects of how the project will be judged; and finally, the outputs are the decisions made at the gate. These are usually Go/Kill/Hold/Recycle decisions (Cooper, 1990; Cooper, 2016). The highly structured development process helps to structure the work and to make sure that the key questions decided at the beginning will be addressed in the project (Veryzer, 1998). Based on experiences, this process also provides improved quality, since the focus is only on the most important activities within each stage where the aim is to systematically build knowledge throughout the process in order to make good decisions (Nyman, 2010). The stages included in a stage-gate process are: scoping, building the business case, development, testing & validation, and launch (Cooper, 2001).

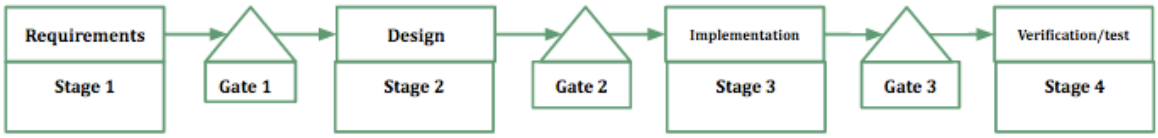

FIGURE 2.2: STAGE-GATE MODEL

2.2 Lean

During the 1959s Taiichi Ohno at Toyota Motor Corporation (Hines et al. 2004) founded the Toyota Production System (TPS) with a focus of eliminating waste through various kinds of improvement activities in production flows (Hines et al.. 2004; Monden, 2011). Essentially, the system includes Just in Time (JIT), Kanban system (see description of the systems below in the section lean tools) and respect for human (Hines et al. 2004) and could be described as a method to produce products in an effective way, primarily through cost reduction or productivity improvement by waste elimination (Monden, 2011). With TPS, Toyota was able to deliver high quality products at low cost and in time, whilst they did get paid without having a high inventory level (Liker and Meier, 2006). Additionally, even during slower periods Toyota was able to make a profit by using the system.

2.2.1 Lean Production

Many of the principles at Toyota were later applied to other manufacturing fields. It is important to understand that Toyota's strategy might not be successful for everyone, it is also hard to "implement"; Liker and Meier (2006) discuss the importance of developing your own strategy that corresponds to your organization and not just copy what Toyota did, it is in fact almost impossible to copy what Toyota is doing today because they are improving all the time. However, the extent of Toyota's system was still limited in the western countries until the book *"The machine that changed the world"* by Womack et al. (1999) stated that there was a gap between Toyota's development system compared to other car developers (Hines, 2004). In this book the terms "lean production/lean manufacturing" were introduced. The definition of lean production is *"doing more of everything with less of everything"* (Morgan and Liker, 2006, p.3), and this result in a much faster, better and cheaper production where less resources are required according to Morgan and Liker (2006). Furthermore, lean production is also describe as *"a set of tools (e.g. Kanban, Andon, Poka yoke) that eliminates waste and creates flow of material through a transformation process"* (Morgan and Liker, 2006 p.5).

2.2.2 Lean product development

In order to entirely become a lean enterprise it is not sufficient with only having a lean manufacturing. Morgan and Liker, (2006) states that the lean transformation has to move upstream, to a level of product and process development. Therefore, a framework for the Lean Product Development System (LPDS) was developed by the authors based on the identified principles: process, people and technology (Morgan and Liker, 2006). The LPDS describes as *"the importance of appropriately integrating people, processes, tools, and technology to add value to the customer and society"* (Morgan and Liker, 2006, p.5). However, companies have faced challenges of becoming a lean enterprise, partly due to that they have overlooked the need of integrating cooperation among different functions such as

sales and marketing design, purchasing, manufacturing, engineers and suppliers (Morgan and Liker, 2006). However, integration among functions is central to LPD, especially in the product design phase (Morgan and Liker, 2006). Finally, according to the authors, LPD provides great competitive advantages to all consumer-driven organizations, which daily are dealing with unpredictable environmental changes.

2.2.3 Resource and flow efficiency

In lean there are two different ways of how to look at effectiveness within a process or organization, mainly *flow efficiency* and *resource efficiency*. Many of today's traditional organization are more focused on resource efficiency rather than flow efficiency (Modig and Ahlström, 2012). Basically, too much focus on resources leads to much negative effects such as increased amount of work and longer lead-times. All of the effects have bad influence on the flow of the process. *"Focusing on flow efficiency is a way of overcoming these negative effects"* (Modig and Ahlström, 2012, p.67). This means that the focus should be on the flow of the processes rather than resources, which further will eliminate waste in a development process (Modig and Ahlström, 2012). In order to achieve a flow effectiveness, integration among functions are going to be needed in order to speed up the transformation of knowledge and information between functions (Modig and Ahlström, 2012). The authors states that if the flow effectiveness is within the focus, the resource effectiveness will be improved in the same way. However, to put flow and resources in practice, resource optimization basically means that resources (e.g. people) are being allocated to projects, contrary to the flow optimization where projects are being allocated to the resources.

Modig and Ahlström (2012) introduced a framework called "*The effectiveness matrix"* explaining the relation between resource and flow efficiency, see Figure 2.3. The condition to strive for is to get both high flow efficiency and resources efficiency. This is when a company fully utilize their resources at the same time as they meet their customer demands (Modig and Ahlström, 2012). Because of the changing product requirements, variations occurs in product development, which are important to manage (Morgan and Liker, 2006) and to understand, since the variation will affect the organization's possibilities of combining high resource efficiency and flow efficiency. In order to achieve this condition; it requires knowledge about the customer needs of today and in the future, and the utilization of resources must be flexible (Modig and Ahlström, 2012). With this saying, the resources must be flexible in order to adapt to changing demands without giving impact on the lead-times.

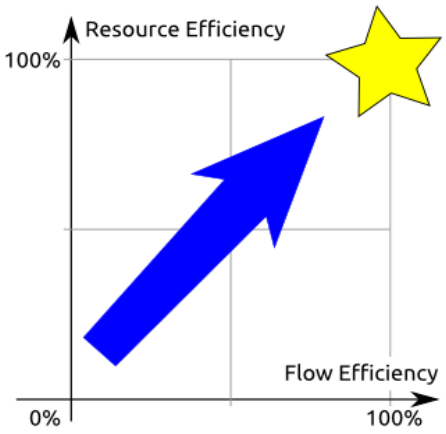

FIGURE 2.3: THE EFFECTIVENESS MATRIX (MODIG AND AHLSTRÖM, 2012)

2.2.4 Tools of Lean

In lean production there are many different tools, one is *Kanban* which is a product control system that controls the production or development in a *pull-system* through signals (Morgan and Liker, 2006) and the system helps to measure the effectiveness of the process (Liker and Meier, 2006). Another one is a pull-system, which means that overproduction is eliminated by giving signals from downstream to upstream of their needs, such as material. Contrary, in product development, knowledge and information are the "materials" that are demanded from downstream activity and the authors emphasize that "*Knowledge is the fundamental element (material) in product development"* (Morgan and Liker, 2006, p. 96). In lean product development the pull-system helps to sort out the right information to the right person at the right time (Morgan and Liker, 2006), this concept is called "*just-in-time"*.

2.3 Agile

One question to answer before going into this chapter is, what is the link between lean and agile? The link can be described in different ways, but according to Björkholm and Brattberg (2008), lean is a culture with focus on resource efficiency focusing to maximize value to the customer and several methods and tools are available for different purposes. On the other hand, agile is a collection of values, which mainly comprises the ability to adapt to changes. Within agile there are methods that supports the lean culture, e.g. methods that provide attributes such as mobile development as well as methods to increase the work effectiveness, which at the same time increases the quality of the product. The combination of lean and agile methods works fine together, where for instance, lean can be used for analyzing the process while agile methods are used to execute the work effectively. The authors' also state that the common between lean and agile is that both have evolved based on experiences rather than through research.

2.3.1 History of Agile

In the middle of the 1990s many found the plan and document-driven approaches to be frustrating and hopeless (Highsmith, 2002). The technology and business requirements did fluctuate since never before, resulted in out of date requirements and plans within a very short period, and besides this the customer became unable to express their needs (Williams and Cockburn, 2003). In the traditional approaches it was believed that all requirements could be anticipated in the early phase and further be able to eliminate late changes and reduce the cost. According to Highsmith and Cockburn (2003) early elimination of changes implies limited or non-responsiveness to changing business conditions. This led to the actual question of how organizations could improve themselves of managing unexpected changes throughout the product's entire lifecycle. However, gradually, the plan began to fizzle out and was no longer the main focus. Instead the focus was to satisfy the customer at the deliveries of the product and not at the project initiation as before (Highsmith and Cockburn, 2001). As a result from those above mentioned issues, practitioners from different IT development areas met in order to find methods and practices with the aim of higher the rates of change and out from this the *Agile Manifesto* evolved, see chapter 3.3.3 (Williams and Cockburn, 2003; Davis, 2012). However, the agile methodologies have been criticized by the practitioners and academics, partly due to lack of scientific support for the stated values that are included in the methodology (Moe et al., 2008).

2.3.2 What is Agile?

Agile can be described as a set of development methods and is based on a flexible, iterative and incremental process (Beck et al, 2001; Deemer et al, 2010, Nyman, 2010). The purpose of the agile methods is to create adaptive planning, evolutionary delivery, a time-boxed iterative approach as well as a flexible response to change (Cooper, 2016). Through

collaboration in self-organized and cross-functional teams, requirements and solutions are created (Beck et al. 2001). In order to adapt to changes, short iterative cycles with feature planning and dynamic prioritization is utilized. Dealing with unexpected changes requires planning in short and iterative cycles that enables dynamic prioritization (Cockburn and Highsmith, 2001). What they mean is that plans usually go out of date and generally too much effort is put on executing long-term goals, which is waste of time and resources. Nerur et al., (2005, p. 77) state "*agile methodologies rely on speculation, or planning with the understanding that everything is uncertain, to guide the rapid development of flexible and adaptive systems of high value".*

Agile relies on change-driven approach rather than a plan-driven approach as in the traditional way (Moe et al., 2008). Cooper (2016) states that agile is about *"plan and build on the fly"* (p.22). Basically, what he means is adaptive planning in short-term goals rather than long-term goals, partly due to the fast changing requirements that makes it unsure of what the customer are asking for in the future. Other researchers agreed upon that agile development is focusing on short-term planning due to the reason that everything is uncertain (Cooper, 2016). This enables flexibility during the development process in order to rapidly develop adaptable products to the customer that provides high value (Highsmith, 2002; Highsmith, 2003).

Important in agile is to strive for team performance, which is achieved by the two factors: communication and feedback. Agile methods help the developers to keep focus on developing functionalities rapidly in order to get feedback. The agile process is divided into small sessions, called iterations and also sprints, see Figure 2.4. The short-term planning is executed for each iteration, which increase the responsiveness to change product requirements within the development process (Cooper, 2016). The benefits of implementing agile into the IT world identified by Begel and Nagappan (2007) are: *improved communication and coordination*, *faster deliveries* as well as *faster response to changing customer demands and technical challenges*.

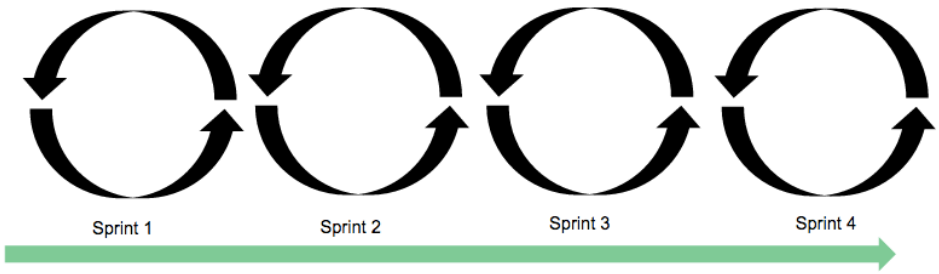

FIGURE 2.4: FOUR SPRINT INTERATIONS.

Chao and Ramesh (2007) are talking about agile in a management and organization setting and explain why agile is important in different development environments. They base their arguments on three valuable areas: *Dynamic capabilities, Coordination and Learning.* The

conclusions based on their study are threefold: agile development suits in high velocity environments; as long as the task uncertainty and interdependencies increase as well as the work group size is getting bigger, the need for collaboration becomes more important; short feedback loops provide a learning environment and improve the communication within the team. However, they ascertain that agile methods are more appropriated in flat organizations than in deep hierarchical organizations.

However, to find only one definition of agility is quite difficult, but according to Cockburn and Highsmith they define agile as: *"Agility is dynamic, context-specific, aggressively change embracing, and growth-oriented. It is not about improving efficiency, cutting costs, or battening down the business hatches to ride out fearsome competitive "storm". It is about succeeding and about winning: about succeeding in emerging competitive areas, and about winning profits, market share, and customers in very center of the competitive storms many companies now fear"* (Cockburn and Highsmith, 2001, p. 122).

2.3.3 The Agile Manifesto

In the beginning of 2001, seventeen practitioners were met for a ski semester in the Wasatch Mountains of Utah in order to associate and to find a common ground of the above mentioned problems (Beck et al. 2001; Williams and Cockburn, 2003). The representatives had their backgrounds from different software development methods, e.g. extreme programming (XP), feature-driven development, crystal clear method, scrum, dynamic systems development and adaptive software development (Nerur et al., 2005), as well as other compassionate who felt the need of changing from the document driven software development approaches. They found that all of them wanted to achieve customer satisfaction and high quality (Williams and Cockburn, 2003). Their method was categorized as agile, which was a term earlier used in flexible manufacturing firms (Williams and Cockburn, 2003). Out from this *The Agile Manifesto* emerged, signed of all the seventeen participants (Beck et al. 2001). After this, agile received lots of attention in development (Cooper, 2016). Below, the agile values that were concluded by the representatives are stated, which are directly quoted from Beck et al. 2001:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value": (Beck et al. 2001 X; Highsmith and Cockburn, 2001, p. 121)

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

"That is, while there is value in the items on the right, we value the items on the left more" (Highsmith and Cockburn, 2001, p. 121)

The twelve agile principles were also founded for the agile manifesto, which is directly quoted, from Beck et al. 2001.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity-the art of maximizing the amount of work not done-is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2.3.4 Agile Principles

The agile methodology consists of a set of guiding principles rather than a rigid methodology. Overall the principles are based on the agile manifesto, but a few researchers has summarized the principles and selected those that are the most urgent according to them. Therefore the core principles have vary to some extent depending on the researcher. Two different summaries of the manifesto will be presented, executed by Nyman (2010) and Brattberg and Björkholm (2008):

Nyman (2010) summarized nine agile principles out of twelve from the *Agile Manifesto.* Number nine, ten and eleven were excluded from the agile manifesto. However, the

principles are reformulated in such a way so they can be applied in non-system development, meaning that the principles can be used in other fields than only software development. Below the principles are stated, with a short explanation and other researchers that confirm to the principles stated by Nyman (2010).

The most essential is to satisfy the customer through early and frequent deliveries of valuable result, the customer is of the highest priority. Highsmith and Cockburn (2011) said that customer satisfaction at the deliveries are the most important and not at the project initiation as in traditional approaches. Highsmith and Cockburn (2001) further states that customer satisfaction is achieved by having close collaboration since the customer needs and demands will easier be outspoken. According to Collier (2011) the focus is to deliver early and continuously deliver business value throughout the product's entire lifecycle.

Adapt the development to changing requirements, openness for changing requirements is natural in agile development, even though the changes appear in late phases, make use of the changes as an advantage for either the organization or the company. Highsmith and Cockburn (2001) say that in agile development changes are promoted rather than rejected.

Deliver useful results frequently, rather within a few weeks, will provide faster feedback-loops between the developers and the stakeholder in order to guide the development into the right direction. Immediately feedback is required in a turbulent changing environment, said Cockburn and Highsmith, (2001).

Business people and developers are working close together, collaboration and communication provide possibilities for collective action (Cockburn and Highsmith, 2001; Cockburn, 2002; Highsmith, 2003). Highsmith and Cockburn (2001) states that the importance of collaboration is emphasized by many researcher, since it provides increased speed and cost savings as well as improved communication by face-to-face communication.

Create self-organized and responsibility-taken individuals is a success factor, meaning teams that are self-driven and capable to take decisions in the team. Cao and Ramesh (2007) point out the importance of focusing on coordinating people into teams. Self-organizing teams is a contributing factor for successful innovative projects is stated by Hoegl and Parboteeah (2006); Takeuchi and Nonaka (1986). However, this point in the agile manifesto also promotes that the team members should be motivated. As a result of having self-organized teams the power can be shift from the management to the teams, which is hard to accept for many organizations (Nerur et al., 2005).

Face-to-face communication to share information, this is seen as the most efficient way of communicating, both within the team and with people outside the team. Informal face-to-face interaction is promoted rather than formal document-driven interaction (Cao and Ramesh, 2007).

Useful results is the primary measure for progress, it provides immediate implementation and fast feedback-loops to guide the development in the right direction.

Simplicity, do the right thing, neither more or less, more effort than needed is not given, which will open up for possibilities to achieve an even development pace. Chao and Ramesh (2007, p 46) state, *"Rather than trying to do things right, the agile approach focuses on doing the right thing to enhance the customer's business value. Both customer and developers frequently adjust their strategies and action by directly monitoring the feedback that result from their decisions"*.

Regularly evaluate and adapt the way of working in order to increase the effectiveness, continuous improvement within the team is essential to always become better and find more effective ways of working. Retrospective is an event in the scrum methodology that is aimed to evaluate the previous sprint and recommended improvements in the team (Cooper and Sommer, 2016).

2.3.5 Agile Practices

In order to achieve the mentioned principles the agile development is supported with a number of practices or methods. First, the scrum methodology is presented and the most common elements within the method. Secondly, agile practices are presented in such a way so they can be adapted into different phases in a project context. To make sure, the practices must whatsoever be adapted specifically for each organization. Therefore, the different practices presented below are only briefly described.

2.3.5.1 Scrum

Scrum is a framework developed in the 1980-1990s with the aim of managing complex product development and consists of a number of *artifacts, tools* and *roles* and is a way of how to organize teams. Basically, scrum is a handbook that tells you exactly how to work in a scrum team such as roles, activities and rules. The scrum methodology provides the possibility to higher the effectiveness in the development and process strategies through empirical directions, prioritized work, in which make that selected tasks can be more efficiently completed. The results of introducing the scrum approach can be seen in a short time perspective and further, it provides tangible progress in the development (Layton, 2015). The common elements in the scrum methodology are:

Artifacts

The development is executed throughout short iterations, called time-boxed sprints or only sprints, which are shorter time periods from normally two to four weeks. Within each sprint the work is focused in order to reach the decided goals for the period. In agile the concept of "definition of done" is really important in order to estimate the work to be done and it means

that the team should together decide of what should be done during this sprint and "what is really done". Usually it is when something is potential for release (Björkholm and Brattberg, 2008). Only after each sprint there are possibilities to reprioritize the work tasks (Cockburn and Highsmith, 2001). In the sprint there are several activities such as; sprint planning, daily scrum, retrospectives, demos etc. (Schwaber & Sutherland, 2013). In the beginning of each sprint a planning session is hold, where they define the goals for the coming sprint. The backlog is developed and prioritized of what product features that should be done during the sprint. The daily scrum is a meeting session that is held every day of 15 min, where three questions are answered by all members in the team, *what did you do since previous meeting, what will you do the coming 24 hours, do you have any constraints?* In the end of the sprint the team has a retrospective meeting, where the previous sprint is evaluated and suggestions for team improvements are discussed. After one sprint is completed, a new one is started and then it continues until the goal is achieved. (Cooper and Sommer, 2016). The usage of sprints contributes to an even pace, "a project heartbeat", which provides to a productive work that helps the team to meet its assignments (Wells, 2009).

Tools

Visual tools provide effective work such as visual management, better prioritization and easier adaption to changes. The most used one is the product backlog is a common tools, which is a prioritized list of tasks or features that should be done within one sprint. The backlog is prioritized by the product owner with support from internal stakeholders and customers. Though, only the known and well-understood requirements are included in the backlog and the requirements will gradually increase as long the customer needs and insights become clearer as well as the product evolves. As the sprint start the backlog will be broken down into smaller tasks and these will be put up at the scrum board in order to visual the progress and to show who is doing the different tasks. The scrum board provides burndown charts, which measure and compare the actual progress with the expected progress. (Cooper and Sommer, 2016)

Roles

The scrum team consists of a product owner, a development team consisting of different competences and finally a scrum master. The scrum team is a self-organized team, which means that the team decides how to accomplish their tasks themselves, instead of being directed. All the knowledge required is also within the team in order to manage the tasks without any help from the outside. The members have no individual titles, and the responsibility of the work is shared between the team members, even the special competences on the members (Schwaber & Sutherland, 2013). Projects are believed to be more successful when the responsibility is shared among the team (Cooper and Sommer, 2016). However, the product owner is responsible for maximizing the value of the product and the team's work. This is done mainly through prioritizing the product backlog. The product owner also needs to listen to new requests from the stakeholders regarding opinions on the product. The scrum master's role is to see that rules, applications and theory are followed. For example, it is to help the product owner to understand the empirical environment. The scrum master is also

acting as a coach for the development team in order to help them create a high value product. (Schwaber & Sutherland, 2013)

2.3.5.2 Agile practices in a project context

Identification of stakeholders

Since two of the principles focus on producing useful results it is recommended to make clear of what results are asked for in an early phase. This is done by identifying stakeholders, which have important opinions about the product as well as providing knowledge of what requirements and needs they have. This gives the team a common understanding what the stakeholders want. The product owner (a role in an agile development team) will take the responsibility to clarify and transfer the stakeholders' needs to the development team and also, prioritize the order of what the team should focus on in the development process. (Nyman, 2010). Highsmith and Cockburn (2001) claim that importance of customer collaboration in the agile methodology, meaning early involvement of sponsors, customer, users and development expertise. This combination of experts lead to increased possibilities to quickly change directions, which implies that contract with the customer is necessary to create.

Goal setting

In agile development the aim is to define the goals at a higher level due to the knowledge required to set up all goals in early phases is not available. If working agile one should take decisions according as knowledge is built during the development process. The aim is to be able to adapt the new knowledge in order to change direction and to set up new goals to match the actual market needs and requirements, in which results in beneficial deliveries. (Nyman, 2010)

Visualization of the work

Transparency is important in agile and one way to achieve this is to visualize the work progress and the problems that emerge in the project. Most commonly a "task board" is used where the aim is to show; what do we have to do, what are we doing right now, what is done etc. It is important to adapt the visualize method so it adapts to the specific project. (Nyman, 2010)

Organizing teams and roles

Create a clear picture over the individuals that are working within the project in order to make clear what roles are important in the organization/project. In agile projects it is really important to have a "product owner", which represents what needs and requirements are needed and has a great understanding of what results that are expected and how to utilize the result. The product owner must be known to the others within the organization and also to have defined responsibilities and authorities. (Nyman, 2010)

Another important aspect of agile is to enable teams, e.g. scrum teams, that can work undisturbed and to get focused on what they are doing. This is done by striving for persistent teams, consisting of 5-9 members where all of them have high availability. The people within the team are preferably working full-time together and where no one is forced to go between different projects or organizations. Based on experiences this way of working has proven to be most effective and provides most beneficial outcome. In those teams it is important to have a team leader that makes sure to prevent that nothing can disturb the teams from their daily work. Another aspect of the agile teams is to make them self-organized, meaning that they plan and estimate their own work rather than having a team leader who tells them what to do. The team should plan in such a way so they can deliver functionalities in the end of each increment. (Nyman, 2010).

Project planning

One of the most essential part of the agile principles is to plan, not the actual plan itself. This means that planning should be an ongoing activity throughout the entire project. Usually, the definition of project success is the ability to deliver in time and within budget with all functionalities included in the result. This is according to Nyman (2010) risky, since the functionality that seemed to be good at the project initiation might not be good in the end. Therefore, the planning should be reconsidered so the new knowledge created during the project is utilized. Nyman (2010) means that detailed planning is ineffective and instead he states that the planning should be at a higher level so the plan easily can be changed and also to plan constantly instead of doing the entire planning in the beginning of the project. Traditional planning tools can be used, such as a Gantt chart and through that keep focus on a higher level planning. It is recommended to plan in the beginning of each increment, where the product owner together with the team discuss the prioritized requirements to achieve for the upcoming increment. The product owner prioritize what to focus on and the team estimate the time (usually in story points) needed to execute the prioritized work, in order to make sure about what will be achieved within the increment (Nyman, 2010)

Managing the uncertainty

The ability to manage uncertainty is one big reason why agile work is adopted by organizations. The ability to live with uncertainty and to manage it, is central in the agile mindset. What Nyman (2010) states that information and knowledge increase constantly during the project, and decisions should therefore not be taken before getting the right information and knowledge. In the beginning of a project a small risk analysis can be conducted just to be aware of critical success factors. Another well-used activity used for managing risks is *daily meetings,* which has proven to be very useful. The purpose with those meetings is to update the project status and to discuss eventually upcoming problems. The meetings should not last more than 15 min and each team member should answer three questions: what have I done since last meeting, what should I do until the next meeting, and do I have any obstacles? (Nyman, 2010)

Project execution
Agile projects are divided into small increments where the aim is to develop a small part of the entire project in one short increment in order to regularly deliver beneficial functionalities to the stakeholders, who immediately can give feedback. This lower the risk of developing something that is not asked by the customer. Further, the developers continue to build on the last functionalities. In this way the stakeholder get beneficial results regularly that can be utilized. Moreover, the team continuously improve themselves through collective reflection upon their work, which is done after each increment. (Nyman, 2010)

2.3.6 Agile vs. traditional

Below, the combination of agile methods and stage-gate processes will be presented based on a study by Karlström and Runeson (2005). After this, agile in non-software development is brought up and finally, management challenges when implementing agile methods into traditional organizations will be presented.

2.3.6.1 Agile practices and stage-gate processes

Additional companies have started to integrate agile methods into their traditional stage-gate processes, which has shown to bring lots of benefits (Karlström and Runeson, 2005; Cooper, 2016). The agile-stage-gate hybrid model became a trend and nowadays, leading firms are integrating it into their processes with the aim of developing hardware products (Cooper, 2016). However, the research of integrating agile methods into stage-gate processes is limited. Therefore, Karlström and Runeson (2005) conducted a study to investigate the feasibility of applying agile methods into a context of large scaled software development projects managed by traditional stage-gate management models. More specifically, they investigated the integration of agile teams into stage-gate software product development.

Generally, companies are managing product-driven development by using traditional stage-gate processes, but instead agile methods are originated from small-scale, contract-driven software development projects (Karlström and Runeson, 2005). Managing product-driven development differs a lot from managing single-iteration contracts. As product-driven development all stages such as requirements, design and testing are sequential developed or iterated in cycles (Karlström and Runeson, 2005; Boehm, 1988) where the aim is to limit the requirements changes in order to deliver the product in time, but instead in agile the effects of changes try to be minimized at any time in the life cycle (Karlström and Runeson, 2005). With this saying, they mean that the *scope is the fixed variable in traditional* development and *time is the fixed variable in agile development*. Furthermore, the differences between agile and traditional development was also defined by Peter Fürst, who developed a similar concept: *"In project management, there are three variables:* ***scope of work, budget and time****. In traditional methods, scope of work is fixed (the product requirements), and budget and*

time are flexible. But in a time-boxed system, for each sprint, time and budget are fixed, and scope of work flexible" (Cooper, 2016 p. 22).

However, Karlström and Runeson (2005) studied three companies, ABB, Ericsson and Vodafone where ABB and Ericsson integrated small parts of the Extreme Programming into their stage-gate processes. The result was that both companies did successfully integrate agile methods without changing their stage-gate processes. The third company was Vodafone, where the stage-gate process itself was studied from a program and product management perspective. The aim was to identify problems that could benefit from applying agile methods into the stage-gate process. However, the key characteristics of the companies were kind of the same, global companies with thousands of employees, customers all around the world and at that moment, all three companies adopted traditional development.

The case at ABB was performed successfully; the current stage-gate process was not changed for the agile team, but they instead got management support and coaching. The result of the agile team was delivering ahead of schedule and under budget. In comparison, at Ericsson the agile team did not get any management support in their agile teams. The problem in this case was the management who faced the agile teams as a threat to the rest of the organization but actually the use of the agile methods increased the quality of the work compared to the other parts of the project. However, in the Vodafone case agile was tried in another manner, product management, which indicated difficulties of planning one day at the time. (Karlström and Runeson, 2005)

In the study by Karlström and Runesson (2005, 2006) the outcomes identified of the agile-stage-gate model were following (also stated in Cooper 2016):

- Integrating agile with stage-gate processes made the planning more efficient. The inflexible and fixed plans were avoided so that the risk of project delays minimized.
- The continuous integration of functionalities provided faster and earlier feedback loops and it eliminated the final testing of the entire product in the end of the project.
- Resistance may have emerged among the managers because of losing control over the agile teams.
- Tasks were split down into smaller tasks, which facilitated the estimation of how much work can be done within one iteration and also the time it takes to estimate the work was speeded up.
- The teams did perceive improved internal team communication and they felt that they had control over their work, which increased the motivation of the developers' daily work. The downside of the agile teams was that the other employees perceived them as isolated from the rest of the organization since they are sitting by themselves. However, the positive was that they were undisturbed in

order to finish their work but less good if they lose contact with their other colleagues.

- The face-to-face communication was seen as more effective than communication through documentation. Although, since the agile teams deliver functional software instead of documentations the management thought the project status became more visualized and clear than in the traditional process. The team planned for all activities, even the documentation work, which was showed as effective. Though, conflicts between agile and stage-gate emerged due to the different ways of documenting.
- The customer feedback was improved, in agile it is important to find customers who want to have close cooperation in order to get continuously feedback on the work that has been performed.
- The combination of stage-gate and agile methods resulted in better response to changing customer needs, more proactive and effective way to build in customer needs, improved ways of dealing with resource issues and the cycle time has been reduced and is now more productive (Cooper and Sommer, 2016).

2.3.6.2 Agile in non-software development

At the very beginning, agile was developed based on software methods and aimed for software development. Nowadays, even manufacturing companies have embraced agile methods, in order to be able to quickly react to changing customer requirements (Cao and Ramesh, 2007). Naylor et al. (1999) conducted a study to indicate that agile should be adopt into the manufacturing in order to best satisfy fluctuating demands. Further, Brown and Bessant (2003) state the importance and need of integrating agility in manufacturing to achieve proactive and reactive flexibility. However, still the research is dearth in the field of agile and manufacturing firms as well as for physical products. According to Cooper and Sommer (2016) agile and stage-gate processes for physical products are seemed to provide positive results, such as improved productivity, increased time-to-market and success rates and Cooper (2016) states that it makes sure that the right product is developed to the customer, the uncertainty is accommodated, the development speed accelerates, the team gets focused and the teamwork improves.

For physical product development the scrum methodology is seemed to be the most applicable agile method. According to the existing theory it is also the only method that has been used in the industry for hardware development so far. For instance, short time-boxed sprints, daily scrum meetings, burndown charts, sprint backlog list and scrum masters are parts of scrum that can be used. However, scrum is well adapted in between the stages, but it has only been used for selected phases, most commonly in the technical development phases and testing (Cooper and Sommer, 2016). Cooper and Sommer (2016) states that scrum is not only used for those mentioned phases, it can be used for earlier phases as well such as the concept phase, feasibility, launch or when building the business etc.

Though, Cooper and Sommer (2016) mean that there are more challenges when adopting scrum into the earlier phases and adjustments are needed. For instance that the definition of done must be adjusted, which is mainly due to the hardware is not divisible like software. But what the definition of done is for hardware, varies depending on what company it is. Instead of deliver functional software Cooper and Sommer (2916) suggest to use for instance pictures and words or by using prototypes and CAD drawings and Cooper (2016) adds that 3D printing and computer simulations can be used. However, the most important is to have something that can immediately be given feedback on. The agile approach has made the development more similar to the software development due to the short and fast iterations. The most important is to have something that can immediately be given feedback on. Also the length of the sprints can vary and be decided by the organization. Also, definition what is meant with a done sprint is needed as well as what the fixed and variable elements are. Furthermore, the planning become different, instead of detailed planning, it should be at a higher level and this should only be tentative. Cooper and Sommer (2016) say that the sprint planning is planning on a higher level for the entire development stage and not only two weeks like in software. Cooper (2016) also said that the design becomes more flexible, meaning that faster response to changes.

Moreover, Cooper and Sommer ascertain that adjustments is needed for the teams that actually should be dedicated and co-located, which is not that typical for hardware development due to much waiting time in the process. Instead it is stated that cross-functional teams should be adopted for cooperation among involved functions, but also for cross-functional teams it will be challenging to achieve 100% dedicated they say. Therefore it is suggested to only have the functions in different phases when the specific competences are needed. For instance, to have marketing when the customer requirements are collected.

However, Cooper and Sommer (2016) conducted a case study where the firm adopted the scrum methodology. The outcome was: increased productivity, improved team communication as well as a decrease in misunderstanding, improved work-flow.

To conclude, they said that agile methods should mainly be used for large projects including high uncertainty and when there is a need for much experimenting since it provides incremental product versions, fast learning and much customer involvement (Cooper and Sommer, 2016).

2.3.6.3 Challenges when implementing agile practices and principles into traditional organizations

Since the agile trend have been used in several organizations already, they have faced different challenges when integrating agile methods into traditional organizations. Nerur et al. (2005) stated *"Most organizations cannot ignore the agile wave, but for those steeped in traditional systems development, adoption of agile methodologies will likely pose several challenges since the two software development methodologies are grounded in opposing concepts"* (p 74). Researchers are dealing with questions of how to manage the challenges

and Boehm and Turner (2005) asked *"How do you merge agile lightweight processes with standard industrial processes without either killing agility or undermining the years you've spent defining and refining your systems and software engineering processes?"*

When companies moving from traditional development to agile development a case study conducted by challenges occur due to the different ways of working the different approaches (Olsson et al., 2012). In order to achieve a successfully transformation, the organization must be open for those changes it is going to bring. Nerur et al (2005) claims that organizations must for instance reformulate their business goals and change the mindset of their employees as well as the management. An organization that usually adapts the traditional plan-driven approach changing to a change-driven approach will notice that the organizational culture, structure and management are going to be influenced. Boehm and Turner (2003) state that people's mindset and habits is not easily change and therefore the transformation might probably be slow.

Adapting agile methodologies implies a change from a command-and-controlled management to a leadership-and-collaboration management, meaning that project leaders must release the control and authority they are used to have, which is a huge challenge (Nerur et al., 2005). However, the decision making process will be a huge challenge as well since in traditional environment the project leader takes all decisions, while in the agile environment the developers and the customer take the decisions. It will take an organization an enormous effort, time and patience to build a culture of trust and respect among other employees to facilitate such a collaboration (Nerur et al., 2005).

Other challenges have been faced as well and Boehm and Turner (2005) have identified few of them. For instance, if both a traditional team and an agile team are developing software for the same product the integration might be a challenge due to the different ways of developing software. It is suggested that the management put lots of attention and effort to synchronize the teams, since it is not easily done. Also, due to different life cycles in traditional approaches compared to agile, complications occur. E.g. agile teams deliver functionalities frequently while traditional approaches develop over a longer time period. This is causing that the agile teams must adjust to the traditional approaches in order to get synchronized. Moreover, they have identified human resource issues due to the differences between agile and traditional, which organizations must learn to accommodate. These are for instance timekeeping of individuals, position descriptions, team and individual rewards and what skills that are required in different teams.

Abrahamsson (2000) state that a first step to overcome challenges that emerge, is to introduce agile working practices at first while taking support from the management, which was indicated in the study, conducted by Olsson et al., (2012), to be an advantage since the agile working teams provided inspiration to the rest of the organization which further provided overall positive effects. Olsson et al. (2012) says that the key focus is to integrate cross-functional teams, which will help to overcome challenges of cooperation and information exchange, which is according to Larman and Vodde (2009) even crucial in order to overcome

issues dealing with cooperation and communication. The study by Olsson et al. (2012) further showed that teams focusing on features rather than components could shorten the lead-times and therefore release functionalities more frequently.

However, in order to get the benefits of introducing agile methodology, the management must adapt the right amount of autonomy and cooperation, in which provides increased flexibility and responsiveness in the development (Cavaleri and Obloj, 1993). *"Organizations must carefully evolve toward the best balance of agile and plan-driven methods that fits their situation" (Nerur et al., 2005, p.74)*. Even though the agile methodology provide lots of benefits Boehm (2002) ascertains that organizations should be careful when integrating agile practices and principles into existing practices.

Organizations that have distributed development have also faced challenges. In order to overcome the communication challenges is suggested to have communication anywhere and at any time, which increase the efficiency and the effectiveness of informal communication. In order to achieve a good communication, only a vague structure is needed, which is also good for coordinating the work within the team. Also, it is important to have a rich knowledge sharing between the partners, which will increase the understanding about the particular business as well as the quality of the product. Moreover, it is important to create trust among the partners since it will help to decrease the formalities e.g. contracts.

However, in order to achieve a flexible process it is essential to clearly document the critical parameters in the development. Through continuously process changes it helps the developers to only focus on the necessary requirements and adjust to the current needs, which will minimize the risk of doing something unnecessary. This is facilitated by the continuously planning, in which also guides the development to the right direction. Moreover, the documentation and verification help the managers and the organization to still keep control over the development, which is in their nature. To conclude, these suggestions given will help to create an agile development teamwork with suppliers.

Nerur et al, (2005) states the success of agile development is to find customers who will actively participate with them in the development process where the customer is expected to be collaborative, representative authorized committed and knowledgeable, which it is not easily found, especially not in complex systems.

Even though many challenges are faced when integrating agile methods into traditional approached, companies are still doing it. The question could be, why are they doing so? Organizational changes are absolutely crucial in the competitive and changing world we are living in and by performing changes it creates competitive advantages. Though, as mentioned before it is challenging and many researchers state it is due to insufficiency in the internal communication (Johansson and Heide, 2008). The consequence of this is that people makes resistance and it obstructs the process (Nevis, 1987). Johansson and Heide (2008) stress out the importance of effective communication in change processes as well as having a change

agent. The change agent should possess skills such as coaching, communicating, involving others, motivating, rewarding and promoting teamwork (Gilley et al., 2009).

3 Methodology

In this chapter the research method used for this thesis will be motivated and explained by going into more details in how the literature and the data collection have been conducted and what techniques that have been used, followed with a motivation why all of them were chosen. Lastly the validity and reliability of the thesis will be presented.

3.1 Methodology description

The goal of doing research is to generate new knowledge based on scientific theory, which in turn creates understanding of the reality. To achieve the goal, the decision of what research method to use needs to be discussed. When taking this decision it is helpful to reflect upon the relationship between theory and research. This relationship is usually described in terms of a deductive or inductive approach (Bryman and Bell, 2011).

According to Bryman and Bell (2011) the most common view of the relationship between theory and research is the deductive theory. Deductive theory means that hypothesis is deduced from existing theory. With other words, this means that data are collected and analyzed based on real situations, which hopefully generates in findings that later on will help to reject or confirm the hypothesis. Therefore, deductive is basically about verifying existing theory with empirical data. In contrary, the inductive theory is about generating new theory based on empirical data.

However, due to the strong linear approach of both approaches a more iterative approach was developed, called abductive. This provides a more weaving back and forth approach between empirical data and theory in order to develop new theory (Bryman and Bell, 2011). Furthermore, Timmermans and Tavory (2012) states that abduction refers to an inferential creative process where new hypothesis and theories are created based on surprising research evidence.

According to Bryman and Bell (2011) the abductive approach is similar to grounded theory, which is defined as *"theory that was derived from data, systematically gathered and analyzed through the research process. In this method, data collection, analysis and eventual theory stand in close relationship to one another"* (Strauss and Corbin, 1990, p.12). Several drawbacks are stated for grounded theory, e.g. the time it takes to transcribe all interviews, and whether or not grounded theory results in theory. Certainly, this approach probably provides the most influential strategy for a qualitative research (Bryman and Bell, 2010). This is also according to Locke (2001) a well-suited theory when conducting research in organizational settings. In addition, the positive outcome of grounded theory is for instance; capturing complexity, linking with practice, facilitating theoretical substantive areas that have not been well researched by others, putting life into well-established fields (Bryman and Bell, 2011, p.584). To conclude, abductive theory aims to widen the scope of the research,

compared to the deductive and inductive theory and instead develops a more general and broad theory.

Moreover, research can be either a qualitative or a quantitative study. According to Bryman and Bell (2011) these two differs in many ways, but the most obvious difference is that a qualitative research puts more emphasis on words rather than the quantification of collected data. The qualitative approach emphasis mainly on interviews and literature reviews. In order to answer the research questions for this thesis, a qualitative research was adopted since the collected data was mainly based on interviews.

This thesis was conducted mainly through a number of research steps: First of all, before the scope was decided of this thesis, several unstructured interviews at the DE department were conducted in order to find a relevant research area. In parallel with this a first literature reviewed was done in order to acquire knowledge about the research area and to find out new and relevant research questions. Later on, a more comprehensive literature study was done in order to find theories directed towards the research questions, which became the theoretical framework.

In the next step, the empirical data was collected in forms of semi-structured interviews, observations workshops and benchmarking. The data collection was to a large extent collected at DE in forms of semi-structured interviews with key-persons in order to find out what challenges that have emerged since they introduced agile teams and also to find out ideas of how agile development can be applied into hardware development. Observations have been done continuously since the researchers have been sitting at the company during the thesis period. In so, they could participate in different kinds of meetings and observe the agile teams doing their daily work. Sitting at the company also implied that many unstructured interviews could be conducted, for example at the lunch breaks and in the corridors. This enabled that the authors could easily ask follow-up questions after the semi-structured interview in order to increase the understanding of unappreciated questions. This also provided valuable information for learning and for increasing the understanding about the scope of this research. Furthermore, two workshops were conducted: The first was the “external workshop”, when participants from GTT, Saab Surveillance and Ericsson were met for a seminar session with the purpose of share experiences of one another's different agile initiatives. The second one was the “internal workshop” where the Affinity-Interrelationship Method (AIM) was used, which is a problem solving method. The aim of this workshop was to brainstorm ideas of how the hardware development could be more similar to agile development and all the attendances were here from GTT.

The analysis for this thesis was later to find patterns of common challenges that emerged when agile development is introduced into a traditional organization based on the interviews. This was later compared with the literature in order to increase the trustworthiness of the founded challenges. For the second research questions the received ideas from the interviews of what agile parts can be used in the hardware development process were also compared with the existing literature to find relevance in the answer to the questions. These reasons

increase its relevance that this thesis is a qualitative and deductive research, since existing theory will be compared with the data gathered from interviews.

3.2 The case study

Research design is helpful though it provides a beneficial framework for the collection and analysis of the data. A range of dimensions regarding the research process is the reason to which research design is being chosen. Five different types of research designs are available and these are: cross-sectional or social survey design; longitudinal design; case study design; and comparative design. In this chapter, it is explained why a case study design is decided for this thesis.

Bryman and Bell (2011) explain a basic case study as a detailed and intensive analysis of one single case. They further state that it is the most popular and well-used approach for cases at business level such as workplace or organization. What distinguishes a case study from the other types is the focus on a specific area; where the goal is to concentrate on the uniqueness and further to develop deep understanding of the complexity of the case (Bryman and Bell, 2011), which is known as an idiographic approach. To add, case studies are good for the purpose of seeking answer of specific questions in a short time-perspective (Hays, 2004). With this saying, a case study is appropriated for this master thesis since it last only for a few months. Cases that are appropriated for this approach are for instance: a single organization, location, person or an event (Bryman and Bell, 2011) and for this thesis a single organization will be analyzed and observed and employees will be interviewed in order to get deeper information about the situation.

Case studies are sometimes associated with qualitative studies and not quantitative studies, but this is not completely the truth. The reason for the association is that case study design often favors qualitative methods such as observations and unstructured interviews since these methods provide useful input to the research (Bryman and Bell, 2011). However, Stake (1995) suggests that the selection of cases should be based on the opportunity to acquire good knowledge; therefore it is required to execute cases where you expect the most learning outcome. As stated at the beginning of this section, a case study is an analysis of one single case, but nowadays, it is more common to execute more than only one single case in business research. It allows the researcher to compare the findings between the different cases in order to consider of what is common or unique and based on that be able to provide new theories and answer the research questions.

Since the agile introduction was executed at DE, no investigation has been don since then. Therefore this case study is conducted at this department in order to be able to make a detailed and intensive analysis of their situation of using parts of the agile methodology in their traditional waterfall organization and to investigate of what have happened. Moreover, two other cases were observed as well in order to get inspiration and input to DE, but these

were not as deeply analyzed as at DE since only a few interviews were done. The two other cases will be further descried under the chapter named workshop.

3.3 Literature study

The first focus of the literature study was to create an overall understanding before collecting the empirical data, this was done through researching in Chalmers library database and Google Scholar, where the keywords were; traditional development, lean product development, stage-gate model, the agile manifesto and agile software development. The search engine was focused on books, articles, journals and conference papers. Although, to create the literature chapters same keywords were used from the first searching and new added keywords were; product development, development approaches, agile practices, agile in hardware, waterfall model, cross-functional, organizational structure, challenges when introducing agile into traditional organizations, lean production, lean philosophy, lean tools, agile history, scrum. One entire book was read by both authors, "This is Lean" with Niklas Modig and Pär Åhlström where flow efficiency and resource efficiency were in focus.

The aim of the literature study was to create a sequential flow from traditional product development to agile development. In the traditional development chapter the main problem with traditional development were studied in order to highlight the need for a more agile way of working and also to describe the common development models and organizational structures in traditional development. The second theory chapter that was studied was lean due to the fact that lean principles are close related to agile principles. However, lean is not within the main focus area for this thesis, therefore only the most essential related to the thesis is presented. The main area of the thesis is agile, which resulted in much literature about it in order to understand the agile way of working and different ways of how to approach it.

Furthermore, theory direct linked to the first research question in this master thesis were: "positive outcomes of changing from traditional development to agile development", "challenges that have emerged when agile development has been introduced to traditional organizations", "mismatches between agile development and traditional development" and also "the outcome of combining agile development and traditional development processes". The literature study for the second research question, agile in hardware development, was hard to find, since this area is not currently well researched. Instead, the focus has been to investigate more in the traditional development methods and agile principles/element separately in order to find possible opportunities to answer the question. However, some earlier research was found where they have introduced agile development into the hardware development process.

3.4 Interviews

In a qualitative study, interviews are commonly used in order to collect data even though it is time-consuming to conduct. However, interviews is to prefer since it provides evidence to the real situation and the fact that many of the areas for this thesis is not well researched, made them to a central role for this thesis as well. In this case the interviews were conducted in order to collect qualitative data only, due to the fact that in-depth understanding was desired. This was also one of the main methods and therefore much effort has been put on conducting interviews for the data collection.

3.4.1 Different interview structures

The three most common structures of interviews are: structured, semi-structured and unstructured. For this thesis unstructured and semi-structured interviews were conducted, which is presented below followed with a motivation of why these were chosen.

An unstructured interview is more like an open conversation with a general topic and agenda with no specific questions that control the interview. The advantage of an unstructured interview is that it gives great in-depth data in the focus area of the interview, but it also put some pressure on the interviewee to be able to communicate without clear directions and leading questions. Moreover, it is good to have a checklist that can be as a memory card for the interviewers of what topics that have been taken up in the discussions. The interviewer must plan the time set up for the interview, when new, unexpected topics come up. (Wilson, 2014). The unstructured interviews for this thesis were aimed to help defining the scope of this master thesis. The authors walked around at DE the two first weeks of the project time and talked with employees in order to see what the problem has been since they introduced agile development. However, it was found out that the problems were many and that the hardware and software developers were not fully synchronized. This further led to the two defined research questions that are within the focus for this thesis. Moreover, unstructured interviews were also used to increase the understanding after semi-structured interviews, e.g. if some information was unclear for the authors. Also, by being located at the company also implied that much chitchatting was done e.g. in the corridors and at lunch and coffee breaks, which provided valuable data for the thesis and learning outcome.

Semi-structured interviews follow a template or a guideline of prepared questions or topics, but they are also flexible and leave space for discussions during the interview time (Denscombe, 2000 and Wilson, 2014). The open-ended exploration gives an opportunity to find new insight to new issues that can be missed through a structured interview. It is also good in that sense that it gives no limits of the length on the response, and additionally an even more depth in the subjects is covered in the interview, which is an advantage. The length of the interview can vary from some minutes to hours, but it is good to inform the interviewees on the expected time of the interview. Other strengths of the semi-structure

interviews are that complex topics through probes and clarification are addressed. (Wilson, 2014). The main interviews for this thesis were semi-structured, both at GTT and at the benchmarking companies, to get both a good discussion and direct answers on the research questions. Semi-structured interviews both imply that the answers will be well documented, in which make sure that the interviewees' actual answers are being remembered and when it comes to the phase to analyze the data, it also becomes easier if the data are well documented. Moreover, the fact that the first research question is picking up the environment where the interviewees are today working, provide possibilities to catch their experiences with direct questions from the template in semi-structured interviews. The second research question is covering a more unexplored area and it is therefore good that the template is more flexible as it is in a semi-structured interview. However, the interview process started with creating a template with interview questions that was directly linked to the two research questions. The interview template was later validated by practicing on interviewees and the template was improved after each interview. So, it took around three practice opportunities to confirm the template, but the first interviews were also be used for the data collection. The template can be seen in Appendix B. In most cases all questions on the template could be answered of the interviewee, in some cases a few questions were skipped due to that the interviewee had limited or no knowledge regarding the question. After all interviews, it was ended up with discussions to summarize the interview. The goal with the interviews was that the researcher should not have given their opinions on the subject, but sometimes discussion were needed since some interviewees or the authors asked for more information or clarification regarding the question.

To summarize, seventeen semi-structured interviews were conducted with different employees at GTT. The different people and roles are described further in Appendix A. The interview length did fluctuate from two hours up to four hours and one interview took around eight hours, since the interviewee was willing to elaborate more about the subject and surrounding information that was good to know for the understanding of the case study at DE.

3.4.2 Face to face interview and phone interviews

The two common interview techniques are: face-to-face interview and phone interview. Face-to-face interview technique is an interview form with eye contact between the interviewees and the interviewer. The technique is beneficial, since it show feelings and expressions on the interviewees. Also, it gives the possibility to show visualizations to provide a better understanding of an area. However, sometimes it can be hard to find a date for the interview that work for both partners, also when the interviewer and the interviewees are not co-located, physically meetings can be difficult to execute. In these cases a telephone interview can be preferable in order to save time and money from the traveling costs. In a telephone interview it is important to sit in a silent area with good connection, in order to avoid being disturbed. The worthiness between the two techniques can be seen as the same, as to the trust

on that the interviewees are honest and sincere when answering the questions and giving the information (Denscombe, 2000).

The interview techniques chosen for this thesis were face-to-face and phone interviews. Face-to face was mainly used, due to the fact that it opens up for a better communication and discussions between the interviewee and the interviewers, more understanding and also better reliability when it is direct communication. Two phone interviews were conducted through Skype, because the interviewees were located far away from the office at GTT in Gothenburg.

3.4.3 Documenting the interviews

It is not only about deliver and go through with the interview; the things that are said must be remembered and reflected on. Therefore, it is important to document the interviews. There are several choices of how the documentation can be done; by taking notes, recording with a cellphone, and make a movie etc. An advantage with recording is that more attention can be put on the interviewees and it is possible to go back and listen again to understand it even better, since some information needs to be listened to several times to get the right understanding. A disadvantage with reordering is to process the data, which is time-consuming i.e. go through the whole recording and sum out the important parts that you are interested in. This takes longer time than taking notes, which instead gives a focus on the most important things directly by the interviewer. Moreover, some people think it is frightening to be reordered and that it can be stated by word what they have said, especially if it is covering a sensitive area, but off course, it differs from person to person (Denscombe, 2000). When taking notes instead, the interviewees can get more time to think with the natural pauses that comes when the interviewer write, which can give a more relax feeling to the interviewees.

In this thesis, the interviews were recorded in the cases where it was allowed of both interviewers with their cellphones. Two recordings were made to secure and to have a backup if one, for example did not work. Also, during the interviews notes were taken to make the transcription of the data process faster and through this way, many of the questions got written answers directly.

3.5 Observation

In business research there are several different forms of observations approaches, but the one used in this thesis is the participant observation. This one of the best-known approaches for data collection and associates with a qualitative research where the observer tries to observe the behavior of people in a social setting during a longer period (Bryman and Bell, 2011). In this thesis observations have been done continuously at DE since the researchers have been sitting at the company during the master thesis work has been going on. The aim of the

observation was to get an understanding of the agile way of working and since scrum teams were introduced into DE the observation was done by participating at scrum meetings, daily meetings, retrospectives, demos and planning meetings. Also, the authors could observe the scrum teams doing their daily work.

3.6 Benchmarking

The purpose with benchmarking was to provide learning outcome from a good practice approach of how to introduce agile practices and principles and lean methods into a company. Furthermore, the aim was to increase the possibility of finding more relevant information for making a comparison to the case study at DE and to be able to answer the research questions more carefully.

To find the right benchmarking companies, several companies were contacted by email and by personal meetings. Out from this, three companies were chosen and semi-structured interviews were held at each of the companies, with the main goal to understand their agile transformation, businesses and products. Then, one of the companies was excluded, since the first interview showed that their situation had many differences compared from the case at DE. The two benchmarking companies that finally were chosen were Ericsson and Saab Surveillance, which henceforth only will be called Saab. Both companies have come a long way in their agile transformation, which mainly was the reason why they were chosen. It was expected to get most benefits and learning outcome from them, which also was the case. The authors were provided much knowledge and inspiration about agile. Moreover, both companies have similar business of products as DE, including both hardware and software where agile have been introduced for both areas.

However, Saab was visited two times and Ericsson three times in order to get all the answers of the interview questions. All interviews were semi-structured, face-to-face interviews and the template that was used was similar to the one at GTT, with a few adjustments and additions see Appendix B. In connection to the interview at each company, a guided tour was performed in their work environment to get a view of how they are working. The interviews were later analyzed together with the other collected data and the literature. Below, a briefly company as well as about the companies' agile/lean introduction and current state will be presented.

3.6.1 Saab Surveillance

At Saab, a chief for the software development was interviewed. The department is working in an area of defense funds that provide the customer with monitoring products. The company Saab has 14 000 employees and is located in several places, one of them is in Gothenburg there they have everything from software development to production. Some of their products are radar system on an airplane and giraffe-radar on a military vehicle. Furthermore, all of

their products have both hardware and software, for a customer-oriented product the lead-time varies from two up to four years.

3.6.1.1 The agile introduction at Saab Surveillance

The agile transformation at Saab was started in the software development for around six to seven years ago, and in the hardware development it was introduced quite recently. The empirical findings are primarily based on experiences from the agile introduction in the software development, since the interviewee got most experience in that area.

The interviewee describe agile as: They can do fast changes after new requirements that is coming. They can fast point towards another direction if they get a more important goal. He is during the interview given a good metaphor to describe this *"the old cannonball is the traditional way of working and today's homing missile is the agile way of working"*. The homing missile can change goal position after it has been send away, which was not possible with the old cannonball, it is just following the original plan strictly. With their products, that are hard theoretical to develop and should also work in all kinds of environments, their products are seen as complex. Furthermore, Thomas describe that agile is a method for complex product development with help of the model Cynefine-model of Dave Snowden, which is because it is not possible to use standard methods to develop complex products which are described with the picture.

Thomas got his agile knowledge by reading by himself and also to take courses and has there met main characters from the Agile manifesto. At Saab the agile implementation has been to take in work methods to use. They learned that the things that have worked best, was to take in the top leaders in the change management process and all of the employees. The leaders believed in it and was teaching the knowledge. The employees got the information and education in agile by reading the book "This is lean" by Niklas Modig. He is in his book referring to the Japanese that invented lean. "It is not possible to lift in new methods and tools into a new market", instead it is the basically principles and valuation, and for example the agile manifesto that is the most important. This mean Thomas is something that does not work, to just lift in tools and methods into another environment and expect it to work as in the original environment. What is agile in a company nor is it scrum or agile or a scrum board, it is to adapt to changes, which is more important than following a plan. One principal from the agile manifesto is the leadership, to coach teams and that they in the teams perform well. Thomas said a comparison to that, "a hockey player is not best self without the team."

Previously, Saab was following the traditional waterfall method that did not work. Then they started working with agile in his department because he convinced his leader that they will in three month do "right things" and in "right time", and then got the trust from his chief. He meant that they will deliver in time and also deliver what is needed. When they started working with agile, Thomas had meetings discussing the principals every week, they also had workshops and invited key persons to talk to the employees. Thomas explain that they have

achieved an agile mindset at his department, in other departments they have another focus and are instead good in small agile parts. When they introduced agile, they tried to use consultants to push in agile principles, but that method did not work at all. The successful method was scaling agile, i.e. to raise agile practices and principles to all levels and dimensions at the organization, it was from testing different methods they find the best one for them. In software development, the most important for them have been sustainable teams and be able to learn of the retrospective that is possible through having mostly full time employees and only 10 % consultants. Thomas has now heard from his co-workers that they have never been so satisfied with their jobs like now, in 30 years.

When they started they put together software development and system development, and build their cross functional teams of system construction, software development and test. The teams are working with scrum and use a product backlog and Thomas is the one that prioritize the backlog and make tasks. The emergent things or the things with a short deadline are the most important to the business and are because that high prioritized. In the backlog is all kinds of projects, to get the backlog working, they have weekly meetings and discussions. It can be regarding problems, if a team needs more help or what the definition of done is for the tasks. Thomas also said that he is going around to the groups and listen how the work is going, so he can catch if they need something.

3.6.2 Ericsson

Ericsson is a global organization and a world leader in communications technology through transformation mobility (Ericsson, 2016). Ericsson Lindholmen, Gothenburg consists of 2500 employees, which is a mix of both hardware and software developers. Although, the majority are focusing on software development and around 500 out of 2500 are focusing on hardware development. At Ericsson Lindholmen, two employees were interviewed, one of them was the sector manager at a hardware development department, where radios are developed and the other one was the lean transformation leader. The hardware development department is similar to a flat organization and consists of 120 employees.

3.6.2.1 The agile/lean introduction at Ericsson

Approximately five years ago, Ericsson introduced agile and lean methods to different parts of the organization. First they implemented the scrum methodology into the software development and further they continued introducing lean product development for the hardware development. For this thesis, the empirical study at Ericsson was based on the lean implementation for one of the hardware development department where they have competences in-house for developing the entire product.

A few years ago they become a design center where they had to go from technology focus to a more product focused organization, which was one reason why they introduced lean. In

connection with the lean transformation they reorganize the employees at the department. The sector manager appointed a lean transformation because he said it is hard to combine the daily operational work with an organizational change and a leader who is working full time to drive the lean transformation forward was required and this person should know the organization very well. Therefore, one of the group managers was directed this role, which was the interviewee. The reason behind the choice was due to the sector manager claimed: *"it is difficult to drive an organizational change without knowing the organization"*. The transformation was aimed to achieve a mindset among the employees, which have been the biggest challenge according to them. *"The lean transformation has been going on for five years now, but still they have much left to do. A transformation has a start but no stop, rather it is about continuous improvement",* stated the interviewees. The transformation started by learning themselves, by reading books about lean. According as their knowledge increased they did coach their group managers, who in turn did coach their developers - "train the trainer". However, they combined the lean transformer with an external coach as well, which acted as an inspiratory. The employees were distributed with different "homework", during office time. For instance, they were told to read the book *"This is Lean"* with Niklas Modig and Pär Åhlström, and after they should try to apply lean into their daily work in order to apply lean in their own work environment. The managers divided different parts of lean to go deeper into and thereafter produce own lean material that was later used to educate the employees with.

At the hardware department they have three cross-functional teams and in this case it means competences that cover the whole product cycle, which follow some parts of scrum methodology. Each team has one manager and consists of around 15 members. In their teams, they have designers, people who write requirements, integrator so the hard and soft parts goes hand in hand, verifications people, people that can check that roles are followed, project leaders and technical coordinators; represent the customer's voice. In 2012 they started with one team and changed the matrix organization. The resources were placed in teams, there they wanted them to be 100 % occupied in the team. At Ericsson they have around 5-15 % consultants, depending on the work load curve. Their philosophy is that the team is sustainable, closed, have good communication to each other so they can be self-organized, create good motivation so it can be to switch tasks every day. They still have competence areas so everything is not known for everyone. Each team is working with several different products, but of similar art. The interviewees promoted that that they tell their teams to make experiments and to see how it works in order to always looking for new and better ways of working.

Their teams are using some of the things that exist in the scrum methodology, but they have adapted it to suit their organization and they work with hands on program for the backlog. However, they stated the importance to get an interaction point between HW and SW, due to it is important to understand that they are dependent on each other. The backlog is prioritized by a product owner; that person is on a high level and can take decisions. The product that can give most money on the market is always prioritized in the backlog, when it come to a critical decision to choose between two deadlines.

One method that are well-used at Ericsson, also today, is the “100 days model”, which is a tool for continuous improvement where the employees get 100 days to get used with a new way of working, since it takes time to change behaviors or habits. The aim is that you work with one improvement at the time during 100 days. For example, this method was used for the implementation of the pulse board. After the first 30 days of using the pulse board they have one day for evaluation, how did it go, what have they done, are they going at the right direction or do they need to change the direction. Thereafter, they continued with 30 more days, having one day for reflection and then 30 more days. The last days up to 100, they have only evaluation of how the process did go. The thought is that the employees should have achieved a new habit after 100 days and are standing at the pulse board without thinking. Today, Ericsson is on their eighth time of using this method and they are very satisfied with the method.

3.7 Workshop

In order to collect data in another form than interviews, two workshops were organized. The purpose was to let people with different collaborate and discuss in order to share agile knowledge. The outcome of the workshops was a part of the empirical data that were used for the analyze phase.

The first one was the “external workshop” that was similar to a seminar. The attendances people were five members from GTT, one from Saab and one from Ericsson, see Appendix A for attendance list and the workshop was located at GTT and lasted for three hours. The purpose with the seminar was to get inspiration from other views and to share experiences from the companies different agile. Before the workshop execution, the company members were told to prepare a short presentation of their experiences of the agile transformation and what challenges they have met. The workshop was recorded with cellphones and the authors also took notes. The participants appreciated the idea of having this kind of workshop when involving different companies and wanted to meet once again for the same purpose.

The second workshop was the “internal workshop”, when only employees from GTT attended, see participating list in Appendix A. The workshop lasted for three hours, the method that was used was the Affinity-Interrelationship Method (AIM), which is a problem solving tool based on two of the seven management tools in quality management, the affinity diagram and the interrelationship diagram, and is accomplished step by step (Alänge, 2009). Due to the human factor, the rules stated by Alänge (2009) were not fully adapted, which is stated below. The aim of conducting the workshop was to get constructive brainstorming ideas about how agile can be used in the hardware development. In the AIM method the initial question is the subject of the workshop. According to Alänge (2009) this question should be formulated as *“what is the problem”* or “*what was the problem”*, but for this workshop it was instead a *“how”* question. Therefore, for this workshop, the method was tweaked a little bit and some assumptions were taken in order to make a conclusion in the end

of the workshop. Also, according to the rules the method should be executed without any talking (Alänge, 2009), but this was not totally followed as well. Finally, some participants attended at the beginning and left before the completion of the workshop due to different reasons. Overall, the workshop provided lots of ideas and five main groups were identified, which can be seen in Appendix D.

3.8 Data analysis

When all interviews were completed at GTT the answers were written out fairly and put under each question and the data under each question were analyzed. If several interviewees had similar answers, the quantity was calculated and if a knowledgeable person had an own suggestion that answer was given a high score. The other answers were summarized and also added in the empirical findings together with the concluded data from the benchmarking companies and the external workshop.

3.9 Reliability and validity

In qualitative research it is important to consider the validity and reliability of the data for the purpose of establishing and assessing the quality of the data (Bryman and Bell, 2011). However, some researchers argue that validity and reliability are inappropriate for qualitative research, which this thesis is. Therefore, writers have changed these criteria to concepts that are more appropriate for qualitative research. They suggest that trustworthiness is the assessment of how good qualitative study is (Bryman and Bell, 2011). Each qualitative concept has a parallel with the quantitative criteria, mentioned below.

3.9.1 Reliability

Reliability concerns the consistency of measuring a concept and in qualitative research this is mentioned as dependability (Bryman and Bell, 2011). Three factors are discussed when considering whether or not a measure is reliable: Stability, internal reliability and inter-observer consistency (Bryman and Bell, 2011). Stability means whether or not a measure is stable over time to make sure that the result of the measure does not fluctuate. Internal validity means instead whether or not the observers agree of what they see, note that there is more than one observer. This is similar to the inter-observer consistency.

Moreover, external reliability means the degree to which a study can be replicated (Bryman and Bell, 2011). In qualitative research this criteria is hard to fulfill, due to the human factor. Bryman and Bell (2011) means that social settings and the circumstances of an initial study are impossible to “freeze”. However, there are suggestions of how to approach the external reliability. For instance, the ethnographic researcher should find similar social roles to the procedure in order to increase the possibility of getting repeated results (Bryman and Bell,

2011). It is also important that the researcher make sure to document the procedure of the data collection.

For this study, reliability is meant whether or not the outcome of the case studies would be the same if conducted once again. Further, to increase the reliability the case studies are executed by two persons and more than one method are used for the data collection. Furthermore, to reduce the possibility of bias both authors conducted all interviews together, both recorded and took notes. Moreover, both authors also conducted the workshops and the generation of the final codification jointly. After each interview and workshop, the authors took fifteen minutes off to discuss the outcome and thereafter both authors transcribed the interviews as time permits. Consequently, this was time-consuming but also rich source of evidence. Additionally, the researcher was provided much continuous informal feedback from the involved project members during the interviews, meetings and coffee breaks.

3.9.2 Validity

Research validity is the most important criteria in many cases. It deals with whether or not a measure of a concept really measures the concept (Bryman and Bell, 2011). Furthermore, validity concerns with the integrity of the conclusion, which is based on a piece of the research. There are several ways of establishing the validity.

Internal validity is about ensuring the correct relationship between two or more factors, observed by the researcher (Bryman and Bell, 2011). For instance, if x is believed to cause the variation of y, how to make sure that this is the right conclusion to make? This criterion will be significant for this thesis since the first research question is about finding challenges between agile development and traditional development. The goal is to find the reasons of the different challenges and to further give suggestions of how to manage them.

External validity, aims to assess whether or not the outcome of a case study can be generalized, which is important for qualitative research (Bryman and Bell, 2011). It is suggested to do multiple case studies and to concentrate on the uniqueness in the case to get deeper understanding. The outcome of this thesis is quite generalized and in a broad perspective rather than going deep into for instance one challenge.

3.10 Ethical consideration

In research, it is important that the researcher are not making any conclusions if the founded data is morally correct or wrong, but it is important to be aware of that the authors can get questions after publication about the study (Carlsson, 1997, pp.55). Ethical consideration to take, can be where the data was found and how the data was read etc. Ethics are seen as the value to get more knowledge and still maintain the individual freedom and integrity.

Furthermore, all people have the right to decide over their privacy, which can be religion and political view etc. So, to not expose the individuals the questions can be anonymously answered. Moreover, this is a security, so an unamortized should not be able to connect the data to a specific person. However, in some cases it is important to connect data with a person and then it is important to use precaution to still keep the anonymity protection, for example through coding of names that are later destroyed when they are no longer needed. Because, the data can contain information about for instance political view, the information should be secured so it should not be able to be used against the information provider (Carlsson, 1997, pp.57). Since this thesis relies mainly at interviews, it is important to have in mind that some information in this report can have been misunderstood when the authors have transcribed and analyzed the data.

Universalism
Ethical principles should always be followed in research, otherwise it will hurt social research i.e. relationships among individuals within a society research area.

Situation ethics
In some cases the principles have acceptance for that they have to be broken. In the following two cases it is possible:

1. When the end justifies the meaning i.e. if the information is special and need to be given to the world to give out knowledge that was unknown before.
2. If an issue is met and the researchers want to dive deeper into it.

Ethical transgression is pervasive
In research it can be parts that are questioned if it is ethically correct, this is the nature, and it will always be like this. If the researcher is totally honest about the intention of the research, the questioned people can intentionally hide information, then the researcher need to be dishonest to get further information.

Anything goes (more or less)
A name is an identity, but in some cases a name is not everything and a person don't get harmed if the ethical rules are broken, and then *only* it is okay to break them.

Diener and Crandall (1978) in Bryman & Bell (2011) categorized ethical principles into four groups:

Harm to participants
Research that will probably harm the activity person is unaccepted of the most people. The word harm can also be hard to define exactly what it means. It can be everything from the meaning of physical harm to losing a job. In some cases is it also to consider the non-participants from getting harmed.

Lack of informed consent

Participants that maybe are going to be in the research should be given as much information as possible about the research, before a decision to participate in the study are taken. In a meeting with possible participants it is important that the researchers are introducing themselves with this title, also give the purpose of the study and extra information that is important to know for the participants, but even if this is done you cannot be sure on that everything is understood correctly by people.

Invasion of privacy

All of us have the right to privacy and it mean to not state names in the report without permission. It can also be that sensitive information should not be able to link to a specific person, the participants should always have the right to withdraw. If recordings are made during the interview it is the same principle to consider, they should not be able to be used to harm a person.

Deception

When research is presented as something else than what it is, it is called deception. This should of course be avoided, due to that trust is lost to the researcher and it can then be hard to find more attendant.

4 Empirical data

In this chapter the information and data given from the interviews at DE, Saab and Ericsson, and the workshops is presented aiming for being able to answer the research questions. All data was analyzed and compared before it was put it into the report. In Appendix B and C, the interview templates are presented and the outcome from the internal workshop is presented in Appendix D. To facilitate the reading, valuable words and sentences that symbolize the content in the paragraph are marked in the paragraphs in this chapter. However, the main challenges identified during the analysis of the empirical data are broken down into smaller challenges, which are presented in subchapter 4.2 and are the final ones. The data analysis does also consider what agile parts the interviewees believe are possible to use in the hardware development process.

4.1 Analysis of the empirical data

Based on the empirical findings several challenges were found. So far, the agile introduction at DE has been for the software developers. This means that the challenges are most related to their experiences. In addition, experiences based on Ericsson's and Saab's lean and agile transformation are added as well, which are both related to hardware and software development. Since Ericsson and Saab started their lean and agile introduction a few years before GTT, they have already come across some of the challenges and have found ways of how to manage them. Moreover, the empirical findings indicate that agile development provides several benefits to the organization and to the development process. The suggestions of what can be used into the hardware development process are based on the interviewees at GTT and the benchmarking companies.

Based on the interviews several challenges were faced due to a number of reasons. GTT is an automotive development organization, consisting of well implemented processes that have been used for many years and due to this it will increase the challenge of introducing agile methods, which was especially stated by one interviewee. Today there are ***dependences*** between departments, partly due to vehicles consisting of several components are developed in different departments and in different ongoing projects, which means that several departments and projects must be ***synchronized.*** The well implemented processes and the dependences between departments are reasons behind one of the major challenge faced during the data collection, namely the interface between DE and the other departments at GTT, which are still working according to the traditional approaches. Basically, the challenge is how to synchronize the newly introduced agile software development teams with other departments of the organization. According to the interviews at GTT, 8 out of 17 interviewees did emphasize the difficulties when several parts of organization are either working according to the agile methodology or having the understanding of agile development. When agile teams are introduced into a traditional organizations adaption from both directions is required, both for the agile teams to adapt to the processes at GTT and that GTT adapts to the agile teams, was stated by one scrum master.

One problem related to the difficulties of the synchronization, is that the other departments have different time plans compared to the agile teams at DE. At Electrical & Electronics Engineering (EE & E), they did do an organizational change in connection with the agile introduction, for instance they reorganized people from different departments to put them close together in a cross-functional team and they changed the time plan in order to adapt the actual time plan at EE&E with the agile software development teams´ time plan, namely the baselines. Though, the issue is that the other departments outside EE&E have not adapted their time plans to the agile teams, which implies lack of understanding to one others different needs and time boundaries.

When there is ***lack of understanding*** and adaptation from other departments problems occur. For instance, other departments are still expecting deliveries at the same time as before from the software developers, but since the software developers adopted the scrum methodology the deliveries are not the same anymore. Currently, the agile teams are working in sprints of two weeks and after each sprint they are able to deliver functional software. However, an interviewee who is both a product owner and a scrum master said *"It is very fun to work according to agile principles and practices! It makes a clear progress, things are happening all the time. However, the drawback is that the other departments do not have the same pace in their development...in the agile teams, we are running in a very high speed compared to the other departments and this is hard for people around to understand and to see"*.

In addition, ***the way of how the agile teams plan differs a lot from the traditional way*** and even though the different plans must be synchronized, was stated by several interviewees. Based on agile principles one should plan all the time in order to be able to go to the right direction, therefore several short-term plans are conducted in the agile teams. The plan is more like a guideline to get an overall view of where to go, but in the traditional way they only make a few plans and they have to strictly follow the plan as well and motivate why changes occur. One of the interviewees stated *"they put much effort on planning that becomes out of date the day after anyway, and then they must reflect upon why we did go out of the plan"*.

However, the different ***time plans*** affects the test and verification process as well. The agile teams are now able to deliver functional software every second week and the desire from the software developers´ are to immediately get the tests performed, though this is not always the case. One interviewee stated *"In worst case scenario, they only put the software on the shelf, where it will stay for three months before it gets tested"*. The issues regarding the time plan was stated by the 8 of the 17 interviewees as well, who meant that GTT has currently not fully adapted their organization to suit how the software development or how the agile development are shaped.

Another challenge related to synchronization and the time plan is the ***strict deadline***, which is the moment when all projects should deliver the final and complete components that are prepared for the final testing for the whole vehicle, namely a 100% working component. This

implies that all requirements must be fulfilled and after this moment none changes are acceptable and the deadline is aimed to both the hardware components and the software. However, this strict deadline was determined a long time ago at a higher management level and at that time it was conformed to the hardware development cycle rather than to the software development cycle. What many interviewees agreed on, is that the software development differs a lot from the hardware development, e.g. in terms of the changing rate, which means that more rapidly changes appear in software development than in the hardware development, such as new upcoming market needs and requirements.

Depending on how long the time period is between the ***strict deadline*** and the product launch, they must be able to adjust to the new requirements and needs, otherwise there is a risk that the product is obsoleted too fast after it is released to the market. Based on the interviews with software developers they stated that, *"nevertheless the differences between how the software compared to the hardware, the software developers are not allowed to make any changes after the deadline. However, as long as the changes do not have any impacts on the entire vehicle or system, we used to do some adjustments anyway"*, which they claimed should be acceptable from the management. According to some interviewees, the baselines developed at EE & E are not fully appropriated for the software development either, since there are several deadlines of what they should have done until each baseline. From the interviews it is understood that everyone are aware of the importance of long-term goals, but anyway it is not adapted to how the software development is shaped.

Since the development of the software was taken in-house at DE, ***the strict deadline was not reformulated for the software development***, which means that the software must be 100 % finished at this time as well? Though, this has not clearly been clarified, which have made the software developers confused. The interviewees were asked to define the strict deadline for software; many interviewees agreed on that it does not really exist for software development, and other interviewees did not even had a clue of what this deadline means. For those who knew about the deadline said, *"The deadline exist today because the project want to have finished end-to-end functions that can be implemented and without any bugs that cause problems with the product. Actually, the deadline means that so much as possible is implemented, so it is an accepted and releasable product on the market"*. An end-to-end function is a well-used term for a software function, for example it can be the cruise control or the brake in a truck.

As stated above it is desired that ***extraordinary things can be added into the software after the deadline.*** For example, some interviewees stated that depending on what software that are developed they do not think it is necessary to fulfill all requirements until the deadline and as long as it has no impact on the product or system changes and functions can be added afterwards. What actually impacts the hardware is for example, end-to-end functions that are directly connected to the hardware. However, based on the interviews it clearly showed that a strict deadline a long time before the start of the production and product launch is not easily met for software, when rapidly changes constantly appear. This problems must be exposed at the top of the project management in order to drop the control to the software developers.

According to several software developers, they claimed that they are so into their fields and have much knowledge about software development and therefore they think they can take decisions by themselves of what changes can be made or not after the deadline.

A major challenge connected to those challenges that emerge is that those parts of an organization that introduce agile development are ***still governed from the surrounding world that is not agile,*** which they must relate to. Many interviewees agreed on that parts of GTT are still drudged in the traditional way of working and with a traditional mindset and it will take a long time before coming away from this. However, still the interviewees thought that the traditional way of working had its advantages but also lot of disadvantages. E.g. the waterfall model is a simple method to understand, but the process takes a very long time. Many changes appear during the development process and the traditional approaches limits the ability to adapt to these late changes. The risk of not adapting to the changes is that the customer will be unsatisfied in the end. The requirement that was specified at the project initiation is out of date and new customer requirements have appeared. An agile coach said; *"In traditional approaches the faith is that everything can be known before the development has even started, which is the opposite of agile principles"*.

However, the benefit of strictly following the planned specification is that the budget is clearly stated in the beginning of the project and in the project end they are hopefully within the budget. ***The traditional way of specifying product requirements is perceived as an issue to many of the developers***. One interviewee said *"In some projects I have written lot of specifications, which take a very long time and I also put effort on things that they later did not use or things that did not work. Also, the product cost much more in the specification phase than compared to the production cost"*. This means that ***much effort is put on activities that later on results in waste***. Another interviewee who belongs to a scrum team stated an advantage about the traditional approaches that *"you feel that the work will be finished some time*".

A product owner for software development shared experiences of working according to traditional approaches: *"Once when I was responsible for the software we used the waterfall model and it worked pretty well then. The environment did not fluctuate much at that time, but when the cellphone with displays were released, the fluctuating environment did increase. At that time the waterfall approach was not preferable since it limits the possibilities to adapt to changes. Today it is even more important to design and fulfill customer needs in order to add high customer value. Also, less resources were needed for developing products before because than I was working on six projects at the same time and today I only work on one project"*. However, the interviewee continued saying that he thinks that ***agile development might be difficult in automotive industries***.

Introducing agile principles and practices into a traditional organization implies an ***organizational change***. According to one interviewee, who is a member of a scrum team, was sure about that the problem is not the introduction of agile teams into a traditional organization. Rather the problem is that a cultural change is needed and to make other

departments understandable and aware of the agile way of working. Another one claimed that it will be a huge challenge to influence the surrounding world to get the agile mindset and it will take a long time. A project leader also claimed *"theoretically, there should not be any problems of introducing agile teams into a traditional organization, the problem is that people do not want to change",* but he further stated that a reorganization is needed in order to get the agile implementation to work. DE has already started the reorganization in order to find out how agile can be used in a best way.

According to the interviewees at Ericsson the "biggest" challenge is ***the classical change management process*** itself and to get all the key persons on board. At Ericsson that they did not get other departments on board at the beginning of the lean transformation. Since they also have boundaries to other departments, means that they need to collaborate and it becomes easier if everyone have the same way of thinking. The sector manager said, *"If we should have done the transformation once again, we should have put more focus on getting other departments on board earlier, since it is important to influence as many as possible in an organizational transformation. In that case we had not been in the same situations that we sometimes were in. For instance discussions regarding resource-related issues would have been avoided".*

Another challenge that was brought up by the global process manager; today GTT is to a large extent a project oriented organization and in the agile methodology it is rarely you see projects, he stated. Their desire is to go from this project focus and instead continue to become a more ***product oriented organization*** in which will improve better flows in the processes and in the development. The product focus was also stressed out at the internal workshop, but it was emphasized that this will take a long time to achieve, but still essential to achieve in a long-term perspective. The process manager stated that they want to strive for having an even pace in the entire organization. The wish is to find a cadence with multiples so it is adapted for both the hardware and the software. For instance, if the software developers are capable to deliver every second week - the hardware deliver every eight week. With this saying the hardware is 4 multiples of the software.

Within the ***EvolvE initiative***, much focus is to achieve product focus and flows rather than projects and successively let the projects fizzle out. However, the transformation leader at Ericsson stated that they still have projects coming in for the hardware development, but not for the software development. According to the manager at Saab, they have reached the goal of thinking in terms of products, or actually they have become a product oriented organization. They utilize a so called program (platform), where 80-90% of the employees focus on developing for the program, rather than projects. However, the customers still order in forms of projects, but now these projects have stepped up one level and parts of the projects are distributed into the program and other parts that do not fit into the program are immediately distributed to a suitable backlog. What was meant is that projects are still there, but they try to break it down as much as possible.

Since they introduced agile teams into DE, they are currently dealing with ***resource related issues***, such as allocation of the different resources. For instance, how to organize people for the different projects. When projects are initiated, the traditional way of allocating resources is to distribute people from different departments or lines (functions), depending on what competence is needed in the projects. Also, people are used to be involved in several projects at the same time, and when the projects are ending up, people are going back to where they "belong". This becomes a problem if striving for achieving persistent teams where all members are 100% occupied. At DE it has been a challenge to achieve persistent teams, since some of the team members are involved in more than one project. This is frustrating for the project managers, which was especially claimed by one ETL. He prefer that his resources only focusing on his project in order to protect that the deliveries are in time, but currently this is not the case.

Due to teams that are involved in several projects and belongs to different departments, managing the ***budget*** becomes a challenge. The budget for a project is fixed to one specific project and it is anticipated that all resources are working 100% for this project. The problematic is when people from different departments are distributed into the same team, which means that they belongs to different budgets and departments. The issue is which budget goes to which team member? Contrary, this is not seen as a problem at Saab since the product owner is bookkeeping the team members in how many hours they are working in a specific project. In this way the managers get control.

Moreover, in some cases their ***budget*** is not enough for full time work or the budget might be too narrow so that the team member must go back to the department where they become. Another issue is, for instance at DE where they have designers from the design department, sitting at DE together with software developers that belong to DE. The issue in this case is if the manager at DE should pay the salaries to the team members in her team, even though they do not belong to her department. Also, the managers lose control over their employees, since they are distributed to other departments.

Overall, the interviews at GTT indicated that several issues faced by the employees are related to the ***resources***, and many interviewees thought that these issues limit the ability to apply agile development into the organization. On the other hand, the software manager at Saab said, since they become ***product oriented and focused on flows*** rather than resources the resources was no longer any problem. He said that his team members have different managers, but it was not perceived as a problem, rather the relationships and cooperation among the different managers have been improved. In turn, this has led to smoother interfaces between the departments and the awareness and the understanding about the agile way of working increased.

At Ericsson, they were a traditional matrix organization before their lean transformation, where resources were constantly distributed to different projects. Since they realized that they wanted ***co-located and persistent teams*** rather than project teams, the traditional way of

allocating resources was no more appropriated. They did not want to lend their resources or to have them part time in different projects, they wanted teams working full-time.

However, it takes time to find the perfect team combination and what resources are required for one team. At Ericsson they did reorganize two times before they found their core teams. The first time, they took resources from all functions/lines, but realized that all team members were not 100% occupied, only 50%, which is unsustainable in a long term perspective. Moreover, they reorganized the managers as well, so they took all line/functional managers and distributed one managers per team. Unlikely, if extra competence is needed in one team, they move this competence to where it is needed. This is not according to lean, but the sector manager said and emphasized that they must adapt lean into the reality. Sometimes they must deviate from what the book says. He said, *"Sometimes Stina is best on what she is doing, then let her do what she is best in"*. He further said that in some cases it is easier to adapt the lean principles and tools and sometimes it is more difficult, but they still try to have a "red thread" in all decisions they take. Contrary, the software manager at Saab said that he does not want that one team member is running between different projects or teams. Rather one should focus on to put everything that must be done in one backlog. He said as an example, *"if Kalle is needed 20% in another project, then take those 20% and put it into the backlog that belongs to Kalle's team, so he can stay in his team"*.

The software manager at Saab stated that one challenge of introducing agile principles and practices was that ***the way of thinking became differently***. *"Now our focus is not what resources the projects are needed, instead the focus is on the teams. Parts of the project is being distributed to a team in an even flow. In other words, the movement is from resource efficiency to flow efficiency, where there is a fixed resource at a fixed time. The product owners' focus is to prioritize the backlog and to communicate with the stakeholders to fulfill their needs"*. Further he said that this has provided an improved accuracy in the delivery.

The interviewees at Ericsson also stated the ***difference from traditional to lean***, which is the focus on flow efficiency rather than resources efficiency. He said that lean is about optimizing the resources, but in order to optimize the resources one must start creating knowledge and flows. The sector manages' definition of resource efficiency was, *"The focus is to optimize you as a resource on your expertise competence and to make sure that you work 8 hours a day. The thing is that you should always work 100% and never be inactive. So, if all of us work as much as we can, we are going to be as effective as we can"*. The sector manager got the question if the employees get stressed in this way of working. His answer was: *"Are you kidding? Yes, they become stressed"*. At Saab EDS they had the similar perception as Ericsson. Before the agile implementation they had another structure than today, where they had single competence workers located at different areas and heavy project leading roles. The project leaders knew who was good at something and was "running" after these people. The employees got stressed in this way of working and also due to many ongoing projects at the same time.

When agile development is introduced it implies that organizations go from a resource-oriented organization to a flow oriented organization, which requires a ***new mindset***. For instance, the project leaders do not own their resources anymore. The software manager at Saab said, *"The focus was no more aligned with how to manage the resources as it was before, instead the focus became how to manage and prioritize the backlog"*.

Working with ***backlogs*** is a new method that was not used before agile teams were introduced into the companies. Nowadays, the project leaders must prioritize the backlogs for the scrum teams, which was a totally new way of working and it required a new way of thinking as well. This was not easily done at the beginning one interviewee said. A project leader at DE said that using a backlog is good, since one can easily see the progress in the development. However, he stated that he thinks that some have missed the purpose with the backlog and do not update the progress immediately when they have finished their tasks. This limits the efficiency in the teams and it becomes hard for the project leader to see what have been done or not. At DE some interviewees stated that it takes time to learn how to prioritize the backlogs and the developers will not immediately be experts on how to work with the backlog.

However, when they introduced ***scrum teams*** at Saab they put together software development and system development, and build their teams with people from system construction, software development and test. The teams are working with scrum and a product backlog which is prioritized by the software manager. His thought was only to prioritize the backlog the first time, but he continued having the responsible for the backlog and it works fine he said. In the backlog all kinds of projects is gathered. The emergent things or the things with a short deadline are the most important to and are therefore of high prioritization. All projects are put in one backlog and prioritized in a way so the team gets an even pace in the development, *"there is nothing that is more effective than to work in an even pace"*, the software manager at Saab EDS said.

According to an agile coach *"**the scrum methodology** higher the efficiency and the flexibility in the development. The progress can easily be visualized, but though the traditional project structure is still there, including much documentation and resource issues, which limits the possibility to fully adopt the methodology."* The sector manager at Ericsson compared scrum with a cookbook and said *"The danger with a cookbook is that you will not be a good chef"*. Basically, what he mean is that scrum is a handbook with tools and methods that you directly can implement in the daily work and the danger is that you just do things without reflecting about what value it adds. Another issue stated by one interviewee who is an agile coach said that many of the employees associate agile with scrum, which he wants to change. He wants them to think of agile in terms of flexibility, feedback and quickness.

Moreover, two interviewees at GTT stated that the negative thing of using ***scrum*** is that a bunch of meetings emerged, which they must attend to. One interviewee said that some meetings are unnecessary at those times when the team has nothing to say, for example at the

retrospectives. It is several meetings that are there just because they should be there. One of the scrum teams at DE, recently reorganized the meetings just because they were too many.

Another challenge when going from the traditional approaches to the agile approach is to start develop without knowing where to end and when only the short-term goals are known. This can cause rework, when it in the end does not fit with the long term plan. Another problem stated by several interviewees at GTT was that they started to ***make use of the word "agile",*** since they introduced agile and scrum teams. This has led to consequences in some situations, for instance, a member in a scrum team said that they did stop documenting just because *"in agile we do not have to document"*. The consequence of this is that too less documentation is conducted and they might miss some documentation that is important. Sometimes they do not even know what they did "yesterday" because they did not document it. The software manager at Saab EDS said that they have not stopped documenting since they introduced agile, they just do less documentation today than before. At present, they put less effort on the pre-documentation and more on the after-documentation. This means, first they develop and test to make sure that the solution works and when they know it works, they do the documentation work.

As stated before the agile and lean introduction implied that ***teams*** have been created at all three companies. These teams should include all competences required for the purpose and further they should be self-organized. This have both Ericsson and Saab achieved. Today, their teams do not need much management support at all. All competences needed are within the team in order to cover the entire product development chain. If problems emerge the teams manage most of them by themselves without management support. The managers do not tell them what to do every day, because the teams make up their daily plans by themselves. At the very beginning when Ericsson tried to create self-organized teams, one team came to their manager and said that they needed a cable, and asked for one more person that could fix this, but the manager said, *"all competence that is needed do develop your product, exists in your team, so you can develop your cable by yourself..."*. Such a question does not appear anymore. At Saab EDS the team takes own responsibility to reach the goals and the leaders do not have to turn them to the right or to the left anymore. The teams constantly become better within the team, they learn from the retrospectives as well as learning by doing. Both Ericsson and Saab said that the teams must be allowed to experiment in order to learn and at last they will be a perfect team. They also striving for reaching T-shaped team members, meaning that everyone can do everything. To achieve this, Ericsson is letting their team members try all areas, this takes time but it is worth it they said. The transformation leader said *"The best course do you have every day when you are sitting with your colleague"*. It was understood that Ericsson put much effort on their teams to make them as good as possible. However, at both Ericsson and Saab the teams have improved themselves as a team according as they have been working together. For instance, they become better at estimating their time and how much time is required for each task in the backlog, which leads to a better flow in the development. At GTT is a high amount consultants, from the interviews at GTT, it was pointed out the fact that a consultant may not have the good will to become T-shaped and grow with a team to become self-organized.

However, in order ***to reach self-organized teams,*** the traditional organizations have one common challenge, to overcome the management control and to achieve trust at lower levels. This was also emphasized during the internal workshop and essential to achieve in a long-term perspective. The software manager at Saab said that dropping the control was a big challenge and scary as well, but it was needed in order to achieve self-organized teams. He stated, *"A manager wants to have full control on everything, but actually, who knows better than the team itself?"*. According to several interviewees the project leaders have too much control; the agile teams do not have enough trust in order to fully adapt the agile practices and principles. The reason for this can be many, but some interviewees believed that the less trust is because of earlier projects that were delayed, which was further due to unclear roles and responsibilities within the team.

The issue regarding ***unclear roles*** is one big issue that was repeated during the interviews. One role that was especially pointed at both the interviews and the internal workshop, was the product owner. Sine agile teams were introduced at GTT, it was no longer only component owners (product owners), and since now product owners appeared in the different scrum teams as well, which made the employees even more confused. For example, it is hard to know, who takes the different decisions regarding the product? Another issue that software developers at DE brought up was that since they become a scrum team, they are supposed to have a team responsible rather than individual responsible. This is confusing for the other project leaders outside DE since they still want to have individual responsibilities. They claimed that clear roles and responsibilities should be distributed within the teams and the teams should have more authority.

It was claimed that the today´s product owners have too ***less authority*** to make decisions that are needed. The decision process is quite difficult at GTT, since many people must be involved in all decisions that are taken. For example, the component owner has less authority and cannot take fast decision by him/herself. This is a challenge they are dealing with since agile was introduced, because decisions should be taken fast. Several interviewees stated that the decision making process is way too long today. It is time-consuming and goes towards the agile methodology, which emphasize fast-feedback loops. In contrary, the newly introduced agile teams at DE thought that since they introduced agile development, the decision process is sometimes too fast compared to what they are used too. For instance, decisions can be taken at demos when the stakeholders and developers are collected. One scrum master said, at one meeting they did accept something, which they were not aware of at that moment, and become aware of what they accepted at that time in a later phase. In some cases that acceptance leads to wrong decisions, which can cause project delays. The scrum master further stated that the details goes pretty fast at these meetings, which is a challenge they deals with for the moment.

According to several interviewees, the ***lack of management support*** is one contributed factor of why challenges emerge when agile practices and principles are introduced into a traditional organization. This was also emphasized by the interviewees at Ericsson and Saab

EDS, that management support is the most crucial in a transformation process and preferably, the transformation should start from the top. According to Ericsson, one of their challenge was because they did not successively influence the higher level of management at the initiation of the transformation, which lowered the speed of the transformation process. The interviewee at Saab was one of the persons who started the agile transformation by convincing his manager that *"within three months we will do things right and in the right time"*. He got management trust and started the agile initiative, today his department and other departments have adopted agile practices and principles more or less.

One interviewee at DE who is a transformational leader also stated that the wish is that the ***initiative of a transformation comes from the management*** of the organization, "*a successful transformation starts from the outside and the result from this is a much smoother transformation"*. The transformation process is seemed to be hard, but another interviewee, who is a hardware component owner claimed, *"It will take time for this organization to overcome those challenges that exists today, but we must document and inform about the problematic we face in all projects and further transfer it to the future projects and managers. At last, we will become much better for each project."*.

The interviews at GTT, Ericsson and Saab EDS clearly indicate that the most important factor when introducing agile principles and practices into a traditional organization is to have ***management support*** and which is constantly coaching their employees. At Ericsson they used the concept of "train the trainer", which means that the managers train their employees, which has been perceived as successful for them. Actually they adopted a top-down approach in their transformation, where they started with themselves. "*It is difficult to convince others, if you do not believe in it yourself"*, stated by the interviewees at Ericsson. In order to avoid small islands with agile teams *"that are living their own life"*, which was state by a hardware component owner, it is important to strive for an agile mindset among all employees. Moreover, the software manager at Saab clearly emphasized the importance of coaching at a team level rather than individuals. At Ericsson they distributed different parts of lean to different managers in order to go deeper into an area. From this, they produced own materials that were later used for educating their employees in lean.

Once again the dependences between departments increases the importance of having a better synchronization among the departments. Since the components at DE consisting of both hardware and software, the hardware and the software departments did get much attention. However, what was understood during the interviews, is that ***the hardware developers and the software developers have limited understanding about one others fields*** and the interfaces between them are unclear. At one interview, a hardware component owner claimed that they had too less understanding and knowledge about the software and how it is shaped. Besides this, the hardware unit is located in another building than the software developers, which limits the possibility for fast communication and the possibility to improve one others understanding of the different fields. Contrary, from the agile teams' perspective at DE, they think that the hardware development process is the bottleneck.

There are strong dependences between the software and hardware units since they set ***boundaries*** for each other. The hardware developers state that they must set the requirements and design early due to the long lead time for developing the hardware. If the agile teams make late changes in the software it can have high impacts on the hardware. The software developers state, it is important that the hardware developers plan for unexpected changes that can emerge in the software so the hardware can adopt to these changes. One suggestion was to develop the equipment for producing the hardware more flexible. One interviewee who is a product owner for software said, "*It is important to continuously negotiate the relations between the hardware and the software units, to continuously communicate and discuss what parts that need to be finished and when. In the first phases of the development of the hardware it is about deciding what requirements that need to be decided at first so the software can work after that."*.

Moreover, several interviewees at DE as well as the benchmarking companies emphasized that ***decisions*** should never be taken before it is needed nor before you have enough knowledge to take the decision, which means that it is good to leave options for upcoming changes that are yet unknown. The interviewee at Ericsson said, *"One talks about taking decisions at the right moment and iteratively build up the complexity of the product, but it is quite difficult and a cultural thing as well"*. The sector manager compared this with saying: *"You don't marry someone on the first date, you go on several dates, before that! We do not want to make an option when we have little knowledge*." For instance, they work with two solutions or concepts in parallel.

However, it was understood in the interview with the component owner for hardware that more ***communication*** is desired, especially between the designers, the hardware developers and the software developers, who are involved in the development of one component. This will help them to clarify what has to be done and at what time. At Ericsson they also have dependencies between the software and the hardware and at different times they must be interacted with each other in order to make tests on the wholeness. For this purpose they have used the pulse board, which has been working well for them and a backlog including both the hardware and the software for better integrating among them.

Moreover, the ***pulse board*** helps Ericsson to keep a ***constant pace*** in the development. *"On the pulse board we see the resources and how the process is going"*. This is a problem at DE, when the software has a different pace compared to the hardware. A member in a scrum team said *"the main challenge is that the other departments do not have the same pace as us"*. Also, an interviewee said that there are several stakeholders that must be synchronized before decisions can be taken and everyone have their own opinions that must be considered, which causes delays in our sprints. Several employees complain that there are many "pundits" involved in the development chain and it slows down the process.

One component owner for hardware further stated, *"The agile teams does not know what they want at the initiation of the project and should therefore start their work before the hardware"*, which has been stated by other employees at DE as well. It was emphasized that

for some components the hardware and software development cannot be done concurrently and it becomes a mismatch if the software and hardware have the same time plan. It has been suggested to have some shifting relation between the hardware and the software; when software vs. hardware should start in the project and when they should have the strict deadline?

However, today it is noticed that just a little ***island*** of one department has adopted the agile methodology, which causes much friction in the development chain, which impact both the hardware and the software. Many interviewees stated, to make the work more effective it is important that all involved departments adapts the agile methodology or at least understanding one another. When one department is not synchronized with the agile teams it is directly noticed and the processes gets slower.

However, the ***communication between the hardware and the software*** units at DE is perceived to be a problem, since it was mentioned by several interviewees. It was emphasized that no scrum teams include the hardware developers in their planning, e.g. at the sprint planning. When they have the big room planning, which is a meeting time with other departments), no input from the hardware developers are asked for. Many of the interviewees said that the cooperation is not good enough and said that it is required that the hardware planning is synchronized with the software planning to get an efficient work flow. It is important to report how the process is proceeding for both the hardware and the software. An example where the communication works is for one component at DE, where the product owner is responsible for both the software and the hardware of the component. This implies that the product owner attends on the big room planning and gets information about how the software is proceeding.

One interviewee said, *"The most important is to know dependencies between the hardware and the software and the time plan is the big connection here, when everything should be finished etc. However, I do not see any big limitations when working agile in the software development and not in the hardware development, but there is probably things that can be better than today. It is important to decide where feedback can be given. When it comes to testing, it is some parts that need to be tested with both a hardware and a software, then it is good if both units are integrated. Today, in the time plan it is no agile iteration when both the hardware people and the software people should talk, the work is actually dislocated until both the hardware and the software should be put together."*

Today DE use ***external suppliers***, mainly for developing the hardware of the components, for some components parts of the software is outsourced to external supplier as well. The hardware components are specified by the component owner at DE and later the specification is being send to the external supplier, who is doing the development. Generally, the external supplier is not located at DE, which results in consequences such as limited face-to-face communication, long feedback-loops or if changes must be done it takes much longer time to get the changes accomplished or probably it is even impossible to get them accomplished. With this saying, having external suppliers is seen as a challenge when agile practices and

principles are adopted into an organization. Especially, since the external supplier is involved during a long time in the development process. For instance, if fast feedback is desired there is a second partner that must be involved in order to get it work.

Today the ***requirements are set*** early in the development process for hardware products and these cannot be changed in downstream phases. This is seen as a disadvantage for many interviewees, partly due to it limits the software developers to do late changes, since the software changes can impact the hardware. Basically, the hardware components limit the software developers to fully take benefits of the agile methodology. Therefore, it is a desire that the hardware developers should have the agile mindset as well so no one affects the other. One member in a scrum team and former scrum master said *"make the hardware more flexible so we can add and take away requirements that we realize are unnecessary or out of date"*. Another interviewee said *"we do not know everything in the beginning of the project, we learn along the way while developing"*. What is meant is that all requirements should not be decided at the beginning of the development process since all information is not available at that time. Therefore, it is desired to let those parts that probably will be changed along the way be "flexible".

However, discussions have been going on if ***agile*** development can be used in the ***hardware development process***. According to the software manager at Saab, *"Some people do not think that we can be agile just because we cannot deliver functionalities every second week, but who actually says this? Instead, continue focusing on part deliveries and determine what the definition of done is for each part delivery based on what you are developing"*. What is meant is that many focus on that something functional must be demonstrated every second week, something that can be directly implemented. Instead, if one think of part deliveries one can self decide how often to meet and what should be discussed and demonstrated at each meeting. The purpose is to have constantly face-to-face meetings so fast feedback can be given and adopted to. *"In some cases it is not important or required to have a totally finished hardware, instead good communication on approximately what it is going to be done is more essential"* was stated by one interviewee. Everything does not need to be finished in detail in the hardware, instead smaller parts should be finished. However, the definition of done differs a lot in hardware compared to software and the backlog needs to be adapted based on this.

Another interviewee who is an agile coach said, *"I think it is enough to adapt **lean** into the hardware development process and to strive for **flow efficiency**"*. Another agile coach also emphasized that lean principles and tools is appropriated for the hardware development, e.g. a kanban board, which is suitable to have in combination with scrum board. Furthermore, the software manager at Saab said, *"If the word software is changed to hardware in the agile manifesto, all principles can be used in hardware development process as well"*, which an agile coach agreed on. However, the software manager stated that it is essential to think of principles rather than tools. He means that it is not successful to just implement tools that have been used by someone else, since the value of the tool might not be reflected over and

the value of the implementation might disappear. Therefore it is better to lift it up to a level of principles and think of what value it provides to your organization and how it can be adapted.

Some of the interviewees were skeptical of implementing agile practices and principles into the hardware development process. When the question was asked if they have faced any problems in the hardware development process, one interviewee clearly said *"no"*. Though, when the following question was asked if any possible improvement were seen, the answer was *"yes"*. For example, ***improvement possibilities*** that have been seen during the interviews are the ***specification process***, which at present takes a long time and since suppliers are involved much communication loops emerge during the process; the requirements are decided early in the process and cannot be changed in later phases; the lead times are long; the decision process is complicated since the component owner has less authority and cannot take decisions by them self; the feedback is limited when the work has been handed over to the supplier.

Based on the interviews it was shown that interviewees from the hardware development unit, who had ***limited knowledge*** about the agile methodology and did just associate agile with scrum, were those who were skeptical of adopting agile into the hardware development. For instance, they thought it would be difficult to break down the assignments to smaller tasks, because the hardware is not shaped as the software. Also, the hardware products require performed tested on the products as a whole in order to verify if it works or not, compared to the software that can be implemented and tested continuously.

The component owners at DE only get ***prototypes*** at A-sample, B-sample and C-sample and the time between those timelines are long. For instance, when they get the prototype it might be wrong and not exactly what DE wanted. The reasons for this can be misunderstanding in the specification list or in some cases the requirements can have been wrong specified from the component owner's side. Some interviewees at DE stated that they receives to less prototypes and has limited impact on what the suppliers are prioritizing during the development. In order to give more feedback on the development of the hardware, the supplier must update the component owners at DE more often. If physical models are not feasible, other tools such as 3D-modeling and virtual tools can be utilized.

Regarding the ***decision process*** several interviewees stated that this process takes a long time. Two different component owners for hardware describe that it takes very long time to make decisions regarding a change in the specification of the component, because no one with mandate are close to the component owner, and it is unclear who is allowed to take the decisions. It is a long process to come up to a decision, since one question is going around between stakeholders and other departments until it has met consensus. It is many different "pundits" in these situations because people are from different departments and have their own "most concern thought" about the component. For instance, the design department want a perfect surface and instead the HMI team want it to be user friendly.

However, many interviewees from software development think that several ***parts of the scrum methodology can be used for the hardware development,*** e.g. teams, backlogs, sprints, retrospectives, daily meetings, short-term plans rather than long-term. However, it must be considered that the lead times are longer in hardware development and therefore the sprints duration should be adapted to suit the specific needs of the hardware in order to be able to show something to the stakeholders.

A mechanical manager at Saab claimed that agile methods can be used for hardware development. They have created teams with mechanical engineers that are responsible for the entire development chain. The teams are self-organized, persistent, and everyone is working 100%. If extra support it is needed, they have more resources outside the teams that can be allocated to the teams, e.g. expertise competence. The team use three weeks sprints, they define part deliveries and decides upon what they should have finish within these three weeks. Moreover they have adopted scrum methods such as daily meetings, reviews demos etc.

However, based on the interviews several ***benefits*** were found with ***the agile methodology***. For instance the focus is on short-term planning rather than long-term planning. The short-term planning makes it easier to set realistic goals and higher the accuracy of deliver in time. Since the planning is done regularly and together with stakeholders, product owners and the team the connection between them improves, and it also increase the customer value since the stakeholders have possibility to explain their needs. According to the agile methodology one should plan regularly instead of doing one big plan, which was liked by the interviewees. Now the agile teams plan, but they do not have to strictly follow the plan, *"it is good to be able to look back on something at least"* one interviewee said. Even though agile implies short-term planning it is still important to see the "horizon", to have a vision of what you want to achieve, with some brief long-term plan. This means that you have thought through what you are going to do. For instance, a product owner for a scrum team said that the big room planning provides a good overview of what the long term goal is, and can make sure that there is a plan of what to achieve.

Furthermore, by adopting agile development it becomes easier to ***rapidly manage incoming changes***, for instance software changes can be done within one hour. Several interviewees emphasized that working in teams and with short-term planning makes it possible to accomplish fast changes. Many interviewees like working in teams, and preferred cross functional. Five interviewees think that cross functional teams is good to have for the hardware development and some emphasized that especially in the concept phase. Since external suppliers are involved in the hardware development process, several interviewees said it would be preferably to involve them in a cross-functional team at least until B-sample. Other departments that were suggested in the cross-functional teams are, packaging, component owner for software and hardware and design. Further, it is suggested if working in teams that the team members should sit close to each other, since this will improve the communication. For instance solutions can be discussed much faster and decisions can be

taken faster as well. Today, those who are involved in the hardware development process are located in different buildings, which obstruct fast communication.

However, some interviewees think that having ***external supplier*** might limit DE´s possibilities to adapt agile practices and principles in the hardware development process since a second partner is involved, but others state that it will not have any impact. Either DE tries to involve the supplier in the agile "wave", which will require partnership and closer cooperation, or DE only adopt agile principles and practices by themselves in the process until the specification is sent to the supplier.

Today it is much discussion with the ***suppliers***, what they can deliver and to what price etc. The price is set before the supplier knows if they can deliver everything on the specification list or not, if the price is accepted a contract is signed with the supplier. After the specification list has been sent to the supplier they must translate the specification list to their terms, which takes a long time. If it is realized that the supplier cannot fulfill for instance one requirement on specification list, a negotiation is started to see if GTT can accept this or not. A discussion is started at GTT with all stakeholders and people who are involved in the development in order to take a decision on what to do. All these communication loops takes much time, which was said by a few interviewees, mainly component owners for hardware since they are the take responsible for the supplier contact. However, since the component owner has less authority and it is quite unclear who is taking the decisions, the decision making process takes a long time when the suppliers questioned the specification, a contributing factor to long communication loops is also due to that GTT is a large hierarchical organization. One way to overcome those communication loops with the suppliers is to involve the supplier in an earlier phase in cross-functional teams.

Another interviewee said that the ***supplier can be involved*** when the specification list is created as well so they can do it together. Because now a product owner is doing the specification himself and the interviewee explained that he does not have knowledge enough in all the specifications requirements. *"One idea is to have four hours cross functional meetings to go through the specification list stepwise"*, said a product owner of software and component owner in software and hardware. The supplier has valuable knowledge and they can make sure that effort is not put on things that cannot be developed later on and they can immediately say if some requirements cannot be fulfilled. *"Instead of just thinking and thinking, involve the supplier early",* said the software manager at Saab.

Though, the risk of early ***supplier involvement*** has also been discussed. For instance, if the supplier and the customer (DE) have not build up enough trust to each other this can have consequences that someone is getting fooled. However, a supplier for software development was interviewed who thought that there would not be any problem of involving the suppliers in a team. Further, he said that they have already started to work in agile teams with some of their customers, but this was for software development then. Also, one interviewee at DE who is a product owner for software is working with a supplier that has adopted parts of the

scrum methodology, where she prioritize the backlog of what the supplier should do, which works good.

However, more ***communication*** is needed between the software developers at DE and the supplier who is developing the hardware component, so both know what the others are doing. At DE they have weekly meetings with the supplier, often through Emails or Skype. Some interviewees mean that they are synchronized in the baselines, some other say the opposite that they do not have good integration among the software and the hardware. *"We want to prioritize what the supplier should do, but they should work alone and say how much time the work need"*, said a product owner for software and a component owner for hardware.

One issue regarding the ***suppliers*** was discussed at the external workshop, in what time the suppliers should be chosen and how many suppliers to work with in parallel. It is good to have many suppliers in the beginning of the concept phase. As a suggestion one could work with several suppliers at the same time and exclude once at a time if they cannot meet the customer's (DE's) requirements. One project leader for both hardware and software said *"The process of choosing the supplier before the projects are initiated can be agile"*, meaning that the process is more flexible and "agile" if more than one supplier is used. This might open for a market competition between suppliers, which leads to better prices and better results in the end.

If having ***agile cooperation with the supplier***, many interviewees state that an agile contract must be conducted. At least a statement of work must be signed, where all conditions between the supplier and the customer should be described. For instance, if late changes is possible, it should be explained which type of changes that can occur. Another manager at GTT who outsource their software development, said that they have been started to create statements of work including agile way of working with their software suppliers and several suppliers seem to be positive to that type of cooperation. In the statement of work it also says that the payment is done for shorter periods and per person who is doing the job rather than doing the entire payment in the beginning for the entire development. This was also suggested by one interviewee to pay the suppliers based on the hours they are working.

However, to create ***partnership and cooperation with the suppliers*** before signing contracts was emphasized at the external workshop. At Saab they have chosen partners to work with that they have built up good relationships with, and nowadays they do not have to set up any contract in between them. At Saab they have transparency in the backlog at both Saab and their supplier, which facilitates their work and makes it easier to prioritize if changes occur.

Today the supplier use something that is called a "black box" ***contract,*** which means that they only show their customers the outside of the component and not the inside, which is mainly due to company secrets. Another alternative contract is a "white box", which means that the inside is shown as well. This would be preferable in order to be able to have a more agile teamwork with the supplies, which was suggested by a few interviewees. In that case

everything can be showed and the customer can have more influence in the suppliers work, for instance prioritize their backlog.

Having a "white box" ***contract*** would provide more control over the development for DE, for instance to check the cost so they are not being fouled around. But it also implies that they must take full responsible if something goes wrong with the product and be responsible for those costs that can arise. It was discussed if a "white box" is more expansive than a "black-box", but according to an interviewee the cost will be presumable in a long-term perspective. Further, one interviewee stated that a white-box contract would make the supplier to treat us more as customer, rather than the opposite, because today the employees at GTT are "standing on their toes" for the supplier. The use of a white box should make that the component owners at DE have more influence on what choices the suppliers are making. It will also result in better prices and products, *"but it will also be more work for the component owners if they must be involved in all decisions"*, said a component owner from hardware.

One component owner for software was interviewed who is working ***agile with the supplier***. She acts as the product owner in a scrum team and prioritize the backlog, even though the scrum team is sitting at the supplier's location. The product owner is often invited to participate in their scrum meetings, which is good since things can be cleared out rapidly. However, in this case the supplier was from the beginning opened to work with agile, when they got the question in the project start. *"It can be a common decision with the supplier that they should work with agile. It is good when they work with agile because questions can be answered quickly, but it can also be slow, it is much to individuals how they want to coordinate it. I think it is important that all parts are very agile".*

"If the supplier is able to be flexible as well when working with hardware development. I do not think it is needed that the supplier should work agile, but they should accept the agile way of thinking, like late changes etc.", said a product owner for a supplier scrum team and component owner in software.

4.1.1 Concluding findings from the interviews

To sum up what have been seen in the interviews several challenges emerge when agile principles and practices are introduced into a traditional organization. The reasons are many, but what can be seen is that the agile way of working differs a lot from the traditional. However, the traditional organization is based on development for hardware products and how it appears. Since software is appeared differently, there will be consequences when software development is directly introduced into the traditional approaches. Several of the agile principles and practices are the opposite of the traditional. For instance, agile strives for short-term planning, rather than long-term planning such as in traditional. Many of the challenges were due to that the traditional way of working is still deep-rooted into the organization, which will take time to overcome at the first moment. The challenges that were found are the following:

1. ***The classic organizational change process,*** the agile way of working differs a lot from the traditional way. Therefore the agile introduction implies a huge organizational change, which is a common challenge for organizations when introducing news.
2. ***Resource allocation and persistent teams,*** for instance, people are distributed to different projects as well as involved in several projects, which limit the possibility to achieve persistent teams.
3. ***Create self-organized teams,*** too much management control and less trust in order to achieve self-organized teams that for instance, plan their own work and take decisions by themselves.
4. ***Non adapted strict deadline,*** a strict deadline when both hardware and software should be 100% completed before field tests and the production, which is unadapt for how software appears.
5. ***The decision making process,*** the reasons are several why the decision making process is faced as a challenge, all of a sudden decisions are taken too fast, compared to what people are used with; decisions are taken before having the right knowledge and information, which impacts the software to adapt to the changing environment; less authority, unclear roles and responsibilities obstructs the process.
6. ***Dependences between the agile software teams and the remaining organization,*** unclear interfaces and less understanding on one another's fields such as hardware and software.
7. ***Overcome small agile islands,*** the teams are still govern from the surrounding world and the introduction is lacking management support.
8. ***Supplier involvement in the development process,*** limited communication, slower changing processes and involved people are not close integrated.

Based on the empirical findings the following might be applicable in the hardware development process:

- ***The scrum methodology,*** e.g. backlogs, sprints, daily meetings can be directly implemented into the hardware development.
- ***Cross-functional teams,*** integrate people in teams that have influence on the product development. Decide if the suppliers should be integrated or excluded in the team.
- ***Agile mindset,*** spread the agile mindset to hardware developers in order to better understand each other and also to improve the cooperation between software and hardware developers.

5 Analysis

In this chapter the main challenges based on the empirical study will be presented as well as possibly solutions for how to manage them, without categorizing which one of them that is the best solution. The structure will be the same as the chapter, concluding findings from the interviews, followed with a solution and a comparison with the literature. At last the agile principles and methods that can be implemented into the hardware development process is discussed. The words that are marked with bold in the analysis are the values that will be the foundation for the conclusion in the next chapter. The same titles from this analysis will be given.

An analysis was conducted in order to find answers to the research questions. From previous chapter, the empirical findings, the data were analyzed and challenges were taken out. The challenges in the empirical findings were found through a brainstorming of both the researchers, the tools used were two whiteboards to draw relationships between different challenges that was first identified. These were later broken down in order to find the main challenges. Further, they were described with the reasons to why each of them was a challenge as well as possible solutions. Afterwards, the different theories between the traditional waterfall methodology and agile methodology were analyzed with the empirical findings. As a result, eight challenges were identified which are presented and described below.

5.1 Analyzing research question one

RQ1: How to manage the challenges that emerge when agile development is introduced into a traditional development organization?

- ***The classic organizational change process***

The organizational change has been faced as a challenge since the agile practices and principles implies a totally new way of working and thinking, compared to what the traditional approaches accomplish, which Olsson et al., (2012) agreed on. This is also what Nerur et al. (2005) means when saying *"Most organizations cannot ignore the agile wave, but for those steeped in traditional systems development, adoption of agile methodologies will likely pose several challenges since the two software development methodologies are grounded in opposing concepts"* (p 74). The new way of working implies for instance, working with a backlog, which is a new way of planning and prioritizing of the work, the development speed is much faster and scrum is also another way of working in teams.

Since agile teams that have been implemented into DE one interviewee stated that the agile teams are not the problem itself, rather it is the organizational change. A project leader confirmed this as well, who further stated that it is the people who do not want to change. The transformation leader at Ericsson clearly stated that the biggest challenge in this agile and lean introduction is the classical change management process and to get the key person on

board. The traditional mindset is deep inculcated in people's habits and Boehm and Turner (2003) say that mindset and habits are not easily changed, in which causes that a transformation will take time. This is important to have in mind in an organizational change.

However, in order to achieve a successfully transformation, the organization must be open for those changes it is going to bring (Nerur et al (2005). The authors claim that organizations must for instance reformulate their business goals and change the mindset of their employees and the management. An organization that usually adapts the traditional plan-driven approach changing to a change-driven approach will notice that the organizational culture, structure and management are going to be influenced (Nerur et al., 2005).

Moreover, management support and a coaching management is important in an organizational change. According to an agile coach, the employees are associating agile with scrum, which he wants to avoid. Instead he wants them to think of agile in terms, such as feedback, flexible and fastness, with this saying the three F. This will help to create an agile mindset among the employees. An organizational change implies much changes in the daily work, therefore, the 100-days-model has been perceived as a successful method to use for Ericsson in order to allow time for changing a habit.

- ***Resource allocation and persistent teams***

The reasons why it is challenging to achieve persistent teams are due to several things. Today GTT is a project-oriented organization where lots of projects are running in parallel with the different lines. According to the traditional way the resources or people are distributed to different projects and some persons do also have several ongoing projects, which they are running in between. This was not wanted from either the interviewees at Ericsson or Saab. They want to achieve persistent teams where all of the members are 100 % occupied in the teams. This is not achieved at DE since a project leader claimed the issue that he want his team to work full-time on his project, to make sure that the deliveries will be in time, but today his team members are going between projects. The budgets are fixed to a specific project and related to those who are working 100% within it. According to a project leader, the consequence of having people running between projects is that he does not know how many working hours are put on his project, meaning that it is unclear how much of the budget that really goes to his project through its team members. Another reason for the challenge is that the members in the agile teams are from different departments, meaning that they have different line managers and then also different budget accounts. Sometimes the budget for one team member run out, then the member must go back the department he belongs to. Furthermore, the managers become unsure on who should take responsible over the different team members, since all the members in the agile teams at DE do not belong to DE´s manager. According to Ulrich and Eppinger (2008) either the functional (line) manager or the project manager takes more or less responsibility over the team members, since both the managers cannot take responsibility for the administrative work. Therefore, either the functional or the project organization is used to get out of the control in the reality. Also, the managers said they lose control over their members that are distributed to other departments.

However, it was seen in the interviews that much focus is on how to allocate resources and that this was currently a challenge that they were dealing with.

In the agile methodology it is rarely you see projects; instead the focus is on products in order to create better flows in the development process. On the internal workshop it was also said that they wish to achieve a product focus instead of project focus. This has been reached at both Ericsson and Saab and DE is on their way to strive for a product focus as well, but still the projects are hard inculcated. Through EvolvE initiative at EE & E includes that the focus should be more on products and flows and to let the projects successively be fizzled out. The manager at Saab said, that the issues regarding the resources were not a problem anymore when the focus became on flows and products rather than resources. At Saab the product owner is bookkeeping the scrum team members by how many hours they are doing for a project in order to make clear of how many hours are spent and in which project. Further, it was stated that a backlog facilitates the work distribution as well and to see who has done what. Modig and Ahlström (2012) are also talking about the negative things as too much focus on resources, as increased amount of work and longer lead-times. All of the effects have bad influence on the flow of the process. Therefore, the focus should be on the flow of the processes rather than resources, in which will eliminate waste in the development process. Both Saab and Ericsson pointed out the importance with having persistent teams, at both companies it is a low percent of consultants. At DE today is high amount consultants that can make it hard to get persistent teams.

- ***Create self-organized teams***

The challenge to create self-organized teams is partly due to the strong management control that is still within the organization. Interviewees stated that the project leaders have too much control over what the teams are doing. The higher management has less trust to the lower levels which impacts the agile teams, since they cannot completely adapt to the agile principles and practices that exist. For instance, the agile teams should be allowed to take decisions within the team since they have enough knowledge within their field, but they are not fully trusted in order to have that authority. Also, the teams are supposed to have a "team responsibility", meaning that they as a team is responsible for what they have developed, but the management still wants to have individuals who take the responsibility. The software manager at Saab said that the management must release the control, even though it is scary. He further stated, "*Who knows better than the team itself?"* The interviewees at Ericsson said that the team must be given space and be allowed to experiment and at last they will for instance plan their daily work by themselves and solve daily problems. At Ericsson they introduced a pulse board and after a while the project leader did not lead the pulse meeting anymore, it was the team.

However, it was understood from the interviews with the benchmarking companies that the management could focus on other things rather than controlling the teams, for instance driving the organizational change forward. From the internal workshop, it was said that a change from a "controlled based leadership" to a "trust leadership" is required and crucial for creating an agile organization. Karlström and Runesson, and Cooper (2005, 2006); (Cooper

2016) are also pointing out another similar effect that resistance may emerge among the managers because of losing control over the agile teams. In the internal workshop, it was ascertained that it will take several years to reach this, since it induce an organizational cultural change. This cultural change will take at least 10-15 years to create, said the people at workshop. Cao and Ramesh (2007) point out the importance of focusing on coordinating people into teams and that self-organizing teams is a contributing factor for successful innovative projects is stated by Hoegl and Parboteeah (2006); Takeuchi and Nonaka (1986). Nerur et al. (2005) point out the fact that having self-organized teams empower can be shift from the management to the teams. Saab and Ericsson pointed out that persistent teams are a foundation, where people can get to know each other and the longer time they have work together, the better they will know the competences in the team, i.e. more self-organized teams will be created. Then, that will higher the group efficiency and performance. From the interviews at GTT, it was pointed out the fact that a consultant may not have the good will to become T-shaped and grow with a team to become self-organized, at GTT is a high amount consultants.

- ***Non adapted strict deadline***

Many software developers have claimed that a strict deadline is not appropriated for software development and how software appears. The strict deadline is perceived as a challenge due to the big difference in how the planning is executed in traditional approaches than in agile approaches. According to the traditional approaches the planning is based on one big and long-term plan, which is done in an early phase. It was believed that all requirements could be anticipated in the early phase and consequently eliminate late changes and reduce the cost. According to Highsmith and Cockburn (2003), early elimination of changes implies limited or non-responsiveness to changing business conditions. Henceforth, dealing with unexpected changes, requires planning in short and iterative cycles that enables dynamic prioritization (Cockburn and Highsmith, 2001). Actually, this means that plans usually go out of date and generally, too much effort is put on executing long-term goals, which is waste of time and resources. Nerur et al., (2005, p. 77) state "*agile methodologies rely on speculation, or planning with the understanding that everything is uncertain, to guide the rapid development of flexible and adaptive systems of high value".* Agile aims for small-scaled planning in a short-term perspective rather than long-term and it should be an ongoing activity throughout the entire project. Nyman (2010) states that the most important in agile is the planning, but not the actual plan itself. According to Cooper (2016) agile is about; *"plan and build on the fly"* (p.22)

In software development the plan is changing often due to that rapidly changes appear during the development process. If not adapting the plan to the unexpected changes the risk is that the product will be out of date when it is released to the marked. Contrary, in the hardware, the requirements and needs are not changing in the same rate as the software. That changes appear more rapidly in software than in hardware is also a reason to why the strict deadline is a challenge, since nothing are allowed to be changed after that deadline. Depending on how long the time period is between the strict deadline and the product launch, they must be able

to adjust to the new requirements and needs, otherwise there is a risk that the product is obsoleted fast after it is released to the market.

In hardware development, the development time is much longer than in software, therefore the planning is based on long-term goals and decisions about the product are made early. The early decisions made in hardware prevent the possibilities to make changes in the software, since it can have impacts on the hardware. Many interviewees claimed that the strict deadline is decided based on the hardware development cycle and was not reformulated when software development was taken in-house at DE. However, what the actual strict deadline means for software development is not actually outspoken, which have made the software developers frustrated. With this saying, software development is a more complex activity due to its requirements have high variability. Therefore, the software development cannot be considered with the same defined processes as the traditional (Boehm and Turner, 2004; Highsmith, 2003; William and Cockburn, 2003).

In order to adapt the development planning for both software and hardware, it has been suggested to find an even pace for the entire organization, which will also better synchronize the different departments that have boundaries to another. The process manager stated that they wish to find a pace with multiples so it is adapted for both the hardware and the software. For instance, if the software developers are capable to deliver every second week - the hardware deliver every eight-week. With this saying the hardware is four multiples of the software. According to Wells (2009) the usage of sprints contributes to an even pace, "a project heartbeat". However, other suggestions from agile team members are to allow the software developers to make changes after the strict deadline. They claimed that they know their field so well that they can make decisions of what is possible to do or not, without affecting the hardware. Another suggestion is to make the hardware developers more aware of that the software plans are changing and that the hardware need to be more flexible to follow the software development speed.

- ***The decision making process***

The reasons why the decision making process is faced as a challenge are many. For instance, all of a sudden when agile practices and principles are introduced, decisions are taken rapider, compared to what people are used too. One interviewee said that many decisions are taken at demos, where details are described really quick, where some details can be missed to get the right understanding. Also, sometimes they take decisions before information has matured, which can cause that wrong decisions are taken. In the agile methodology, decisions should be taken fast based on feedback from stakeholders, where the feedback is aimed to guide the development in the right direction. By delivering useful results frequently, it provides faster feedback-loops between the developers and the stakeholders that guides the development into the right direction (Nyman, 2010). Cockburn and Highsmith, (2001) state that immediately feedback is required in a turbulent changing environment.

In contrary, the decision making process is also perceived as slow and that it causes delays, which was emphasized by several interviewees at GTT. The traditional and hierarchical

organization makes the decision process slow, due to all "pundits" that must have their opinions, people have less authority and also that the decision must come from the top. According to the interviews, the problem is due to that people working close to the product are not allowed to take decisions. For instance, component owners have too less authority in order to take decisions fast if they receive questions from e.g. the supplier. All opinions from people involved in the development must be taken into consideration, and after all it is still unclear who has the last word in order to take the final decision. It was suggested that the managers should give the component owners more authority and trust. Furthermore, interviewees claimed overall that it is unclear roles and responsibilities for instances in the agile teams and also who the product owner is? Since the scrum methodology was introduced, more product owners emerged, the product owner in the scrum team as well as product owner for the different components. This increased the confusion even more about who is taking the decisions and who has the authority and responsibility for doing it? However, one of the successful principles in the agile manifesto is to create self-organized and responsibility-taken individuals, meaning that teams are self-driven and capable to take decisions in the team (Nyman, 2010). Further, researchers state that self-organizing teams is a contributing factor for successful innovative projects (Hoegl and Parboteeah (2006); Takeuchi and Nonaka (1986)). As a result of having self-organized teams the power can be shift from the management to the teams, which is hard to accept for many organizations (Nerur et al., 2005).

Another issue that has been expressed from several interviewees, is that decisions about the requirements in the hardware are taken too early, even before having the right information and knowledge about the product. This set boundaries for what changes the software can do. This affects the software developers to adapt to the changing environment, because changes in the software might have a negative impact on the hardware. Both benchmarking companies and interviewees agreed on that decisions should not be taken without knowledge and enough information, for example when stakeholders should give an acceptance at demos, they need to consider that they know what they say okay to. As the sector manager stated on the interview *"You don't marry someone on the first date"*. This means that you must have all the information and the choices clear before a decision can be taken to get married with a person. This is the same principle if it is regarding a technical solution as well. Nyman (2010) also states this; that information and knowledge increase constantly during the project, and decisions should therefore not be taken before getting the right information and knowledge.

- ***Dependences between the agile software teams and the remaining organization***

At present, agile teams exists mainly at DE, where agile practices and principles have been introduced. The dependences become a challenge due to that large parts of the organization are still working according to the traditional approaches, without having any clear interface between the departments that have dependent components to DE and their agile teams. This implies that the agile teams are still governed from the surrounding organization that is not agile, in which limit the agile teams´ ability to fully implement agile practices and principles.

Since the components at DE consist of both hardware and software the dependences between these two units are most essential. However, based on the interviews it was understood that for instance the hardware developers and the software developers have limited understanding about one others fields and that the communication between. A hardware component owner said that she thinks that the agile teams are living their own lives.

Furthermore, it came up that there are also limited, defined checkpoints for hardware and software developers. For instance, meeting points where both the hardware and software developers are gathering in order to share their results and to inform each other of where they are ahead for, if they are able to deliver in time and if they are going to meet any constraints in the nearby. As an example, the big room planning is suggested as a meeting point where both units can be integrated. Relevant information to share can be what they are working on right now, what their plans are and what requirements the software has on hardware and the other way around. At Ericsson they have introduced a pulse board and integrations events in order to synchronize the hardware and software developments and when deliveries are expected from both sides. Boehm and Turner (2005) stated that if both a traditional team and an agile team are developing software for the same product the integration might be a challenge due to the different ways of developing software. Therefore, they suggest that the management put lots of attention and effort to synchronize the teams, since it is not easily done.

- ***Overcome small agile islands***

The agile introduction at DE implicated that agile teams were created, utilizing adapted forms of the scrum methodology. As stated above, the agile teams are dependent on those who have not adapt the agile way of working and that the traditional approaches are still deep-rooted into the organization. This means that small islands of agile teams have appeared into an organization that is governed by the traditional approaches, which obstructs the agile teams to adapt agile practices and principles, for instance to make late changes and to receive fast feedback. However, lack of management support and a bottom-up approach instead of a top-down approach have indicated that small agile islands can appear stated by one interviewee.

Instead of first striving for an agile mindset in the organization, scrum teams are just introduced, which means that employees in the remaining organization or department does not understand why the agile teams do what they do. For instance, interviewees stated that the deliveries were expected at the same time as before the agile implementation. The interviewees basically mean that, since the scrum methodology was adopted the deliveries became differently compared to before. For instance, one agile team member said the scrum teams are now able to deliver functional software every second week and their desire is that the functionalities should be verified and tested immediately, but since the test and verification department is still following traditional approaches there is a risk that the software is just “put on the shelf” for several weeks. This is also mentioned by Boehm and Turner (2005) who said that agile teams deliver functionalities frequently while traditional approaches develop over a longer time period, which is seen as a challenge.

To get a successful agile transformation and to overcome small agile islands, management support is needed. Both Ericsson and Saab said that a supporting and coaching leadership is required in an agile and lean transformation. Also they said that the transformation should start from the top and then use "train the trainer", which further will create a mindset among the employees in the organization. Ericsson used the top-down approach for their transformation. The sector manager said, "*It is difficult to convince others, if you do not believe in it yourself"*.

Karlström and Runeson (2005) made a case study at Ericsson the agile team did not get any management support in their agile teams. Then the management faced the agile teams as a threat to the rest of the organization.

- ***Supplier involvement in the development process***

Since a second partner is involved in the development process it becomes a challenge due to a second partner must follow the "agile wave" as well, which can be both hard and easily done. To what extent the supplier should be involved depends on when the supplier usually comes into the development process. According to Wheelwright and Clark (1992) early involvement of suppliers is an advantage since they provide knowledge and insight of the product, which helps to avoid eventually upcoming problems.

It is suggested that otherwise the supplier is involved in the "agile wave" or excluded. Though, it is important to make clear interfaces if the development organization adopts agile principles and practices. Overall, it is recommended from interviewees to create good partnership and much communication with the suppliers. Also they should try to create "agile" contracts that include for instance, that late changes is accepted and then through discussions, they can together come up with what changes are possible for the supplier to do. At the internal workshop a type of agile contract was suggested that for instance includes information of how to manage the flexibility and where an own model for the budget can be set up.

5.2 Analyzing research question two

RQ2: What parts of the agile methodology can beneficially be used in the hardware development process?

- ***Agile combined with stage-gate processes for hardware development***

Since agile became a trend the interest have increased in how to adapt agile practices and principles into other areas than what it once was aimed for. Therefore, many researcher and companies have asked the question if it is feasible to adopt the agile methodology into hardware development processes. Even though the existed research about agile and hardware development is limited, a few case studies have been conducted and indicate that agile methods can be applied into traditional stage-gate processes (Cooper, 2016; Cooper and Sommer, 2016; Karlström and Runeson, 2005). The outcome of using agile methods together

with stage-gate processes are for instance, better response to changing customer needs, more proactive and effective way to build in customer needs, improved ways of dealing with resource issues and the lead time has been reduced and becomes more productive (Cooper and Sommer, 2016).

According as, the positive outcome of adapting agile methods into stage-gate processes have influenced leading firms to integrate this into the development of hardware products as well (Cooper, 2016; Cooper and Sommer, 2016). According to manufacturing companies the agile methodology have embraced in order to be able to quickly react to changing customer requirements, which Naylor et al. (1999) stated is the best way of managing fluctuating demands. Further, Brown and Bessant (2003) have also emphasized the importance and need of integrating agility into manufacturing firms in order to achieve a proactive and reactive flexibility in the development. However, agile methods and stage-gate processes are seemed to provide positive results for physical product development, such as improved productivity, increased time-to-market and success rates (Cooper, 2016; Cooper and Sommer, 2016). Further, Cooper (2016) state that it helps to guide the development in the right direction due to fast feedback from customers and stakeholders, uncertainties are better accommodated, the development is accelerated and the team becomes more focused and the teamwork is improved. Both the benchmarking companies have adapted agile and lean practices and principles into their traditional organizational processes with the aim of developing hardware products, which has been seed as successful and they have achieved to adapt the new working methods with their existing processes.

- ***The scrum methodology***

According to Cooper and Sommer (2016) the scrum methodology is seemed to be the most applicable agile method for hardware development or according to the existing research, scrum is even the only agile method that has been used for hardware development so far. This was indicated by Sommer et al. (2015) as well, who said that out of those few companies that have embraced agile methods for the hardware development scrum seems to be the most popular one. For instance, sprints, daily scrum meetings, burn down charts, backlog list and scrum masters are parts of the scrum methodology that can be used (Cooper and Sommer, 2016). Based on a case study conducted by Cooper and Sommer (2016) where a company adopted the scrum methodology the outcomes were increased productivity, improved team communication, decreased misunderstanding and improved workflow.

In order to adopt agile work with the traditional processes, Sommer and Cooper (2016) say that scrum can be used within the stages in a stage-gate process, but it has only been used for selected stages and most commonly in the technical phases and testing. However, they continue saying that it can be used for earlier phases as well such as the concept phase, feasibility, launch or when building the business case etc. Though, the earlier phases are seemed to be more challenging due to adjustments that must be done (Cooper and Sommer, 2016). For instance, the definition of done must be clarified as well as what should be done within one sprint. This is particular due to the hardware is not divisible to the same extent as software, which makes it hard to always have something visual or physical to show the

customer and the stakeholders after each sprint. Therefore, instead of deliver functional software alternative methods such as pictures, words, prototypes such as 3D printings, CAD drawings or computer simulations can be used (Cooper and Sommer, 2016; Cooper, 2016). However, the most important is to have something that customers and stakeholder immediately can be given feedback on. The software manager at Saab said that *"Some people do not think that we can be agile just because we cannot deliver functionalities every second week, but who actually says this? Instead, continue focusing on part deliveries and determine what the definition of done is for each part delivery based on what you are developing".* What he means is that many focus on that something functional must be demonstrated every second week, something that can be directly implemented. The purpose is to have constantly face-to-face meetings so fast feedback can be given and adopted to. *"In some cases it is not important or required to have a totally finished hardware, instead good communication on approximately what it is going to be done is more essential"* was stated by one interviewee. Everything does not need to be finished in detail in the hardware, instead smaller parts should be finished. It was also in the interviews that the definition of done differs a lot in hardware development compared to software and the backlog needs to be adapted to suit the hardware development.

However, Saab has adopted scrum methods to their agile teams and the mechanical manager said that it is several scrum methods that could be taken directly implemented to hardware development teams such as daily meetings, reviews demos etc. Ericsson have adopted forms of scrum as well for hardware development. Though, their teams have been allowed to select what methods they want and that suit their specific need and one team have especially adopted scrum methods such as daily meetings. According to Nyman (2010) daily meetings are good since it helps to manage the product uncertainty.

- ***Cross-functional teams and supplier involvement***

Cooper and Sommer (2016) stated that dedicated and co-located teams such as scrum teams are not that typical for hardware development. For instance this is because of much waiting time in the process, which makes it unnecessary to have dedicated teams working full time. Therefore, they suggest to create cross-functional teams in order to create integration among departments who are involved in the hardware development, in which will create better cooperation. The study conducted by Karlström and Runeson (2005) showed that scrum teams were perceived as isolated from the rest of the organization, but when integrating agile with stage-gate processes the communication was improved because of integration among functions was achieved and still the team was focused like in the scrum teams. Anyhow, it will be challenging to achieve 100 % dedicated cross-functional teams throughout the whole development (Cooper and Sommer, 2016). Therefore, they suggest to only have the needed functions in the different phases when the specific competence is needed. For instance, to have marketing when the customer requirements are collected. As a complement, the integration with marketing also provides better customer relationships (Wheelwright and Clark, 1992). Compared to what Cooper and Sommer (2016) said, Ericsson has successfully been able to achieve self-organized, persistent and co-located teams for developing hardware

products, where everyone are 100% occupied and if extra support is needed, they have some resources with special competences that move between the teams when they are needed.

Both the interviews at GTT and the internal workshop indicate that cross-functional teams should be used for the hardware development in order to improve e.g. the communication and the feedback loops. Wheelwright and Clark (1992) and Maylor (2010) say that typical functional structure are usually slow because of limited communication and coordination across different functions, which Wheelwright and Clark (1992) mean is because of people are not co-located. Allen (1970) stated early that physical proximity is the key element in order to improve the communication and interaction among functions. Therefore, Wheelwright and Clark (1992) and Clark and Fujimoto (1991) emphasize the importance of achieving integration across functions, at the right time and in an effective way, which will help to rapidly and efficiently deliver products to the marketplace. Likewise, both Ericsson and Saab have cross-functional teams, but in their case the team members are cross-functional with regards to different competences for covering the whole development cycle. For instance, at the mechanical department at Saab, all members are mechanical engineers. This was suggested to have at GTT as well, which was brought up several times at the internal workshop. However, most focus for the cross-functional teams at GTT was to involve different departments that are involved in the development process and have influences on the product, these were for instance: component owners for hardware and software, packaging, design, test and those who are responsible for the appearance and attributes for the product.

On the other hand, Wheelwright and Clark (1992) further state that cross-functional integration is not required in all types of projects. For example, it might not be needed when the design is stable and the requirements do not fluctuate much, when the interfaces are clear between the functions and when they know what activities are within their scope (Wheelwright and Clark, 1992). Instead, it is required when the market is more uncertain, i.e. when the market requirements and the technologies are changing faster and the time-to-market is getting more crucial (Wheelwright and Clark, 1992). However, Nyman (2010) says that companies must deal with the ability to live with the uncertainty and the need for managing the uncertainty is required in today's environment. Therefore agile work has been adopted even more, but Cooper and Sommer (2016) mean that agile methods should only be used in large projects including high uncertainty as well as there is a need for much experimenting, since it provides incremental product versions, fast learning and much customer involvement. Cao and Ramesh (2007) also ascertain that agile way of working is needed when the uncertainty is increasing and the tasks becomes more challenging, and when the task interdependencies increases.

However, integration among functions brings lots of advantages, especially if some functions are integrated in an early phase (Wheelwright and Clark, 1992). In contrary, Cooper and Sommer (2016) said that it was more challenging of using (in their case) scrum teams due to adjustments needed, but according to Wheelwright and Clark (1992) e.g. an integrated design can be achieved earlier if developers integrate with manufacturing, since information is easier

shared about what is feasible or not. Cooper (2016) also said that the design becomes more flexible meaning faster response to change and when using time boxing the efficiency in the concept evaluation improves.

Due to the suppliers are involved in the development process of hardware components, it was discussed several times, if the suppliers should be affected if the hardware developers at DE adopting agile working and if they should be involved in the agile "wave" or not. However, based on the interviews and the workshops it was suggested to involve them in the agile "wave" in order to take benefits of the actual outcome of the agile methodology. Based on the interview with a software supplier he thought there would not be any issue of involving suppliers, since already today they work with agile with some of their other customers that are buying the software. However, if involving the suppliers they can be included already in the specification or concept phase in order to make use of their knowledge. This might decrease the lead time, due to that questions usually asked in later phases when the specification has been handed over to the supplier can be answered early, in order to avoid to put unnecessary effort on things that later will be waste. Wheelwright and Clark (1992) agreed on that early supplier involvement provides knowledge and insight, which also helps to avoid eventually upcoming problems. During the interviews and the workshops it was much discussions of how to integrate the suppliers in the development at DE and it was suggested to integrate them in the cross-functional teams. A few interviewees thought that involving the supplier in a team could be a threat to the development and could cause that somebody gets folded. In order to avoid that it is important to strive for partnership and trust with the supplier, which was stated in the interviews and the workshops.

Regarding the specification process, many changes cannot be done in the requirements after the specification has been handed over to the supplier. In order to get the benefits out of the agile methodology, changes is wanted to be adjusted so the product suits the actual market needs and requirements. The advantage of involving the supplier in the agile "wave" the possibility for achieving a higher flexibility in the hardware development increases. Based on the empirical data it was suggested to build good relationship with the supplier and become partners instead of using terms of customer and supplier. Furthermore, during the negotiation the statement of work should be discussed as well as an agile contract should be created and signed by both partners. In the contracts it was suggested to discuss upon what requirements that are possible to change and only make early decisions upon the critical parameters. This will increase the flexibility in the hardware development, which was agreed by many interviewees who thought it would be feasible to achieve.

If the requirements can be changed and redefined during the process, it was suggested to pay for the resources utilized rather than to pay for a whole package. One manager at DE said that they have started to formulate a statement of work for agile work with suppliers, where it is suggested that they pay for the time and resources that are utilized for the development. If it after a while shows that less resources are needed than expected, they can adjust to this and then pay less. The manager said for instance they only pay for one month ahead. Other suggestions to increase the flexibility in the hardware development with the supplier is to

work with several concepts in parallel and according as one concept shows to not withstand the current requirements they can drop the concept off, which was suggested by the interviewees at Ericsson. However, in order to achieve and manage this flexibility in the development process, partnership and trust must be created and in order to keep up to the flexibility in the development, incremental documentation is required, which was emphasized during the internal workshop.

Finally, during the thesis execution it was discussed where in the development process agile practices and principles can be adopted based on the current problems or improvement possibilities that the interviewees faced at present. The main focus and suggestions have been at the very early phases, since the supplier involvement starts already after the product requirements are specified. However, many interviewees said that agile can be used in the concept phase and also to set the requirements, which was also emphasized in the internal workshop. For these phases, it was suggested to create cross-functional teams in order to early collect all information needed for the product, in which will decrease the later communication-loops.

- ***Agile mindset***

The software manager at Saab, an agile coach and Nyman (2010) have clearly emphasized that the agile manifesto and its principles and values are applicable in other fields than software development as long as the word "software" is changed to e.g. product or system instead of software. However, the manager at Saab stated that it is always important to think of principles rather than tools. He means that it is not successful to just implement tools that have been used by someone else, since the value of the tool might not be reflected over and the value disappears. Therefore, it is better to "lift it up one level" into principles and then think of what value it provides to your own organization and in what way it can be adapted. However, this is what the agile manifesto is, a bunch of principles. Furthermore, Nyman (2010) reformulated the twelve principles from the agile manifesto and took out nine of them, which he thought did suit in other environments that is not software. He further gave suggestions of how agile principles can be used in a project context. Though, he states that it is important to adapt agile into the specific organization and its environment because one way is not the perfect way for everyone.

Today the requirements are set early in the development process for hardware products and these cannot be changed in downstream phases. This is seen as a disadvantage for many interviewees, partly due to it limits the software developers to do late changes, since the software changes can impact the hardware. Basically, the hardware components limit the software developers to fully make benefits of the agile methodology. Therefore there is a need for hardware development to adapt to an agile mindset as well, many software developers established. One team member in a scrum team said *"make the hardware more flexible so we can add and take away requirements that we realize are unnecessary or out of date"* and another team members said that *"we do not know everything in the beginning of the project, we learn along the way while developing"*. Basically what the interviewees mean is that different possibilities should be adapted in order to make the hardware more flexible

and be able to adjust to upcoming changes and further that all information does not exist in an early phase and therefore the requirements should be able to be more "flexible" and just decide those that are critical. Nyman (2010) stated that due to the need of being able to manage uncertainty is one big reason why agile work is adapted by organizations, which is central in the agile mindset. In the first research question the three F were stated, flexibility, feedback and fastness, which are central terms in agile and are recommended to have in mind in order to successively, reach an agile mindset.

5.2.1 Concluding analysis of research question two

When this thesis was initiated it was known that agile and hardware development is not well researched. Therefore, the theory that was found together with the empirical data was used to answer this question.

According to the theory that exist, it indicates that agile and stage-gate processes are a good complement to each other, so the advantages from both development approaches are added and what do we get then? Of course, an even better process. The combination of agile and stage-gate has also been used for hardware development and several companies have already started to use it. However, the theory indicates also that so far, only the scrum methodology has been used in hardware development. Further, the theory state that scrum might not be 100% perfect for hardware development because of the long lead-times in the process, therefore they instead suggest to have cross-functional teams and to include those people that are needed in the team during selected phases. Also, cross-functional teams is something that has been emphasized in the interviews with employees at DE, so what was found is that cross-functional teams are both needed and asked for in order to make the process more agile at DE.

So to continue with the empirical findings; the benchmarking companies were used as inspiration and the reason why these two companies were chosen is because they have already adopted parts of the agile way of working in their hardware development, which is for instance parts of the scrum methodology. But as mentioned before, scrum is a handbook, telling you exactly how to work with it and one of the interviewees at Ericsson compared scrum with a cookbook and said *"you will not become a good chef if you are just following recipe",* with this saying, scrum is often just implemented without reflecting about if it useful or not. Therefore it is important to emphasize, if scrum is being applied into hardware; only selected parts that put value into the development should be used.

To conclude, parts of agile is feasible to use in the hardware development, but it cannot be forgotten that hardware and software differs a lot in how they are developed and since agile is aimed for software development it makes clear that a few adjustments are needed if agile is adopted into the hardware.

6 Conclusion

In this chapter the conclusion of both research questions will be presented. The identified challenges for the first research question are briefly presented, followed with suggestions in bullet points of how the challenges could be managed. In the second research questions the identified parts of the agile methodology that can be used in the hardware development process are stated with a brief discussion to each of them.

6.1 Concluding research question one

How to manage the challenges that emerge when agile development is introduced into a traditional development organization?

The classic organizational change process, the agile way of working differs a lot from the traditional way. Therefore the agile introduction implies an organizational change, which is a common challenge for organizations when introducing news.

- *Supportive and coaching management, a change must come from the top.*
- *Create a new mindset, think of agile in terms of the three F; feedback, flexible and fastness, and create new daily routines.*
- *A transformation takes time; a transformation will not happen over the night, have patience with that people's mindset and habits take a long time to change.*

Resource allocation and persistent teams, for instance, people are distributed to different projects as well as involved in several projects, which limit the possibility to achieve persistent teams.

- *Change the focus to flow efficiency from resource efficiency, add focus to a product and take away the focus on resources that create wastes.*
- *Use the backlog more to easier locate resources and calculate budgets, through bookkeeping the resources and schedule hours spent in a project.*
- *Make it possible to have persistent teams, have people that are motivated to grow in a team and are there to stay.*

Create self-organized teams, too much management control and less trust in order to achieve self-organized teams that for instance, plan their own work and take decisions by themselves.

- *Put value in team responsibilities, instead of individual responsibilities. Change the focus from individual credits to lift the group's performance.*
- *Let go of the control and create a trust leadership, believe in that the team have the knowledge and knows best.*

- *Persistent teams are a foundation to create self-organized teams, they must get to know each other to know the competences in team.*
- *Opt for firm hiring to create T-shaped competences and good relationships in the teams.*

Non adapted strict deadline, a strict deadline when both hardware and software should be 100% completed before field tests and the production, which is unadapt for how software appears.

1. *Create an even pace for the whole organization that will better synchronize the departments that have boundaries to one another when they should deliver to the entire truck project.*
2. *Create a deliver pace for both hardware and software, for hardware can it be multiples of the cadence if less deliveries are wanted, since the hardware takes longer time to develop.*
3. *Allow changes for the software after the strict deadline, the software developers know their limitations on what they can change and not to get it compatible with the hardware.*
4. *Be aware on that the software is changing, the plan in software development is continually changing, focus on that the hardware should be more flexible to be able to follow the software development speed.*

The decision making process*,* the reasons are several why the decision making process is faced as a challenge, all of a sudden decisions are taken too fast, compared to what people are used with; decisions are taken before having the right knowledge and information, which impacts the software to adapt to the changing environment; less authority, unclear roles and responsibilities obstructs the process.

- *Decide who should take the decisions, give a selected person more authority to take a decision and has the responsibility. If it should be the product owners, give them more authority and trust, clear directions from managers are needed.*
- *Make clear roles in the agile teams, who is the product owner and who is the component owner?*
- *Take decisions at the right time, when there exist enough information and knowledge, which increase during a project time.*

Dependences between the agile software teams and the remaining organization, unclear interfaces and less understanding on one another's fields such as hardware and software.

1. *Create understanding between hardware and software, better communication between the units.*

2. *Create common checkpoints, when hardware and software are going through their results and what they are heading for. A suggested meeting point is the big room planning.*
3. *Communicate requirements needed from each other, e.g. what do hardware need from software and the other way around, which can be visualized on a pulse board.*
4. *Define where the interfaces are to the rest of the organization, where should they be?*

Overcome small agile islands, the teams are still govern from the surrounding world and the introduction is lacking management support.

- *Start with creating an agile understanding and mindset, then the rest of the organization can understand why the deliveries have changed.*
- *Create management support and coaching, to avoid that agile is seen as a threat to the rest of organization.*
- *Start a transformation from the top, to convince others that it work and then the change coming from the right direction seen from a change management perspective.*
- *Let leaders train a new trainer, so the understanding goes top-down. Then the employees will know their trainer, which create better trust in change processes.*

Supplier involvement in the development process, limited communication, slower changing processes and involved people are not close integrated.

- *Make decision if the suppliers should be excluded or included in the agile wave.*
- *Make clear interfaces, so the agile principles and methods can be followed from in-house.*
- *Create a good partnership that have good communication and good cooperation between each other.*
- *Create an own type of agile contracts, make contract from the beginning and state what is expected from the supplier and the other way around, so it can manage the flexibility.*

6.2 Concluding research question two

What parts of the agile methodology can beneficially be used in the hardware development process?

Yes, it is feasible to implement agile principles and practices into the hardware development process. Though, the agility is needed to be combined with the existing stage-gate processes for the hardware development.

The scrum methodology

It was concluded that the agile method to use in the hardware development process was the scrum methodology, but it is important to have the following in mind when using it:

- *Use only beneficial, selected parts from the scrum methodology that suit the specific organization's needs, since scrum is a "cookbook", which makes it easy to just implement something without any reflection of what value it adds.*
- *Adapt scrum methods within selected stages in the stage-gate process, where there is a need for dedicated and co-located teams. E.g. technical phases, testing and the concept phase. For the concept phase, make sure to define what is meant with a "done sprint".*
- *Make sure to have something to get feedback on after each sprint. Instead of deliver functional software, alternative methods can be used such as, pictures, words, prototypes such as 3D printings, CAD drawings or computer simulations.*
- *Important is to have frequent face-to-face meetings.*

Cross-functional teams and supplier involvement

Further it was concluded that cross-functional teams can be adopted into the hardware development. The concept of cross-functional teams did not emerge from the agile methodology because it has been existed since a long time. However, Cooper and Sommer (2016) stated that the rules in the scrum methodology e.g. dedicated and co-located teams do not perfectly suit the needs for the entire hardware development process due to long lead times and much waiting time. Instead, cross-functional teams were recommended. The following are recommended:

- *Decide about what purpose the cross-functional teams are used for and create teams based on, either competences to cover the entire product development cycle or competences from different departments that have influences on the product development. Experiment until the perfect teams are found.*
- *Example of competences needed in the cross-functional teams are e.g. component owners for hardware and software, packaging, design, test and responsible for appearance and attributes.*
- *Reflect on if cross-functional integration is needed. Use it when the product uncertainty is high, when the tasks become more challenging and the task interdependencies increases. With other words, when the market requirements and the technologies are rapidly changing and the time-to-market becomes more crucial.*
- *Make sure that the team can be co-located to improve the communication.*
- *Decide upon if the supplier should be included or excluded in the cross-functional teams. If they are included, create partnership and discuss and formulate an agile contract. As a suggestion they can be included in the specification process and/or in the concept phase. If they are excluded, make clear interface to the supplier.*

Agile mindset

The agile manifesto can be applied into the hardware development process, since it is possible to use the values and principles in another area than the original area. Thinking of agile principles and values, it can foster an agile mindset. The mindset is basically to live with the uncertainty, *"plan today but be aware of that the plan might change tomorrow"*.

- *Think of agile in principles rather than tools and adopt them to suit the specific hardware development process.*
- *Create an agile mindset among hardware developers, think of the three F; flexibility, feedback and fastness and make benefit out of them.*
- *Change the word "software" to "product" or "system" in the values in the agile manifesto, then the values can be applied in the hardware development as well. To put this in practice, the four values in the agile manifesto stated by Beck et al. (2001) are:*

The first value, *"individuals and interactions over processes and tools",* is adapted by integrating employees from different departments in forms of cross-functional teams. This will for instance improve the communication since the possibilities for face-to-face communication will increase. The use of more face-to-face communication provides that faster changes can be adjusted to, since information will be shared between individuals much faster.

The second value, *"working software over comprehensive documentation"*, which can be changed to working *product* instead of comprehensive documentation. By showing the customer and internal stakeholder something visual or physical such as 3D-printing or CAD drawings, quick feedback can be given so the development can be guided in the right direction and the frequent integration between developers, stakeholder and customers foster less documentation, since there will be a constant information sharing.

The third value, *"customer collaboration over contract negotiation",* this will be achieved by e.g. integrating the suppliers in a cross-functional team, which include partnership and collaboration with the supplier and agile contracts that open up for more flexibility in the development process.

The fourth value, *"responding to change over following a plan",* by focus on short-term planning rather than long-term planning the possibilities for making adjustments to changes will increase. Usually the plan goes out of date, so adapt the incremental and continuously planning before one big plan in the initiation of a development process.

7 Discussion

For this thesis, interviews were used as one of the main research methods. Due to that the agile implementation at DE has mainly focused on scrum teams and the implementation was done quite recently, many people did not have so much agile knowledge that we thought when we started the research work. This made it harder to get deeper and concrete information on which specific principles or values of agile that were working good today in the software development, or to find the disturbances of that the rest of the organization does not, which is directly related to the first research question. This made that the authors had to know the agile practices and principles well to distinguish if the information they said was a challenge due to the agile implementation or not related to it, which a balance is made of the interviewers. Furthermore, the interviewees were mainly from the software development department and it was showed that many had less knowledge or experience from the hardware development. So, it was because of that hard for them to answer questions related to the second research question. Also, the people working with hardware are not working with agile, since it obviously is not implemented there. This made it hard to find interviewees that had both agile and hardware knowledge at DE. Another factor is also that agile is a quite new research area and agile in hardware is an almost unexplored area, which make that limited theory is available to read. Therefore, the decision was taken to focus on interviewing people with a lot of agile knowledge to get answer on the second research question, since it was believed that it is enough to know the basics from the hardware development and that they are using the waterfall approach to give suggestions of agile practices. However, the challenges that were found based on the empirical data seem to be trustworthy, due to the research have brought up similar challenges.

It is important to be aware of that agile development is not always the best way of working in product development. The literature state that agile practices and principles suits in high velocity environments; as long as the task is uncertain, interdependencies increases and when the product requirements are rapidly changing (see the agile chapter for a longer description), with other words when the product is complex. So, only because the answer on research question two is yes, it does not mean that agile practices and principles always should be used for hardware development. For instance, when the products are clearly defined and the design is stable the traditional approaches can still be used. Moreover, the agile methodology was from the beginning developed for the purpose of developing software and since hardware and software have much differences in how they are developed and how they appear, also implies that one should reflect over the need of integrating agile practices and principles in the hardware development. However, the agile mindset is not negative to have, since it provides much benefits and it fosters a good organizational culture with lots of trust, good relationships among employees and customer, increased work motivation, improved communication and integration among employees in the organization. Moreover, during the thesis it was realized that the second research question is not if the agile methodology can be applied into the hardware development process or not. A bigger challenge is rather the organization itself that is strictly governed by the traditional approaches, where the traditional mindset and way of working is hard inculcated.

7.1 Recommendations

The recommendation is to consider the suggestions for managing the challenges from the conclusion, which are further described in the analysis chapter with theory. Furthermore, when GTT want to improve an area, for instance the software development speed and make a decision of implementing the methodology scrum. It is suggested to use a tool like the 100-days-model, which is a fast implementation tool that helps to figure out if the chosen method helped to improve the area, which is today as well used at Ericsson. Firstly, start and work with scrum for 90 days and learn how to do it. Secondly, take 10 days to look at retrospective, which means reflect on how it went. Was scrum the right methodology to higher the development speed? How did it go to work with it?

In the agile methodology it is needed to create trust to the teams, which is further stated in the *conclusion*, *chapter 5*. In a big organization it can be hard to let go of all control and then visualizations tools as the pulse board is good to be able to visualize if a team needs help from other teams or can indicate that they will be late with a delivery to the big truck project. Then managers can see a red dot that indicate a problem in the specific category it is concerning.

In hardware development it is suggested to look into how the suggestions from research question two can be implemented. For example how to make the scrum teams and exactly when they should appear.

In a longer perspective it is a recommendation to go from project focus to product focus, which will create a better flow efficiency and coordination of the backlogging. Start with splitting up projects into smaller parts so they are easier to work with, then place them in program and then they are splitter up to the right backlog and product owner. This will make it easier to coordinate in the backlog, make prioritizes that must be done due to the deadlines, which can be when the big truck project needs to have a product it in order to do field tests.

As with all implementations and organizational changes, keep on improving yourself in becoming more agile, there are always improvements to do in an organization to keep a position or grow in market share, which can be done through better processes. And remember, *you are never finished with a transformation!*

7.2 Suggestions for future research

- During the research, it was detected that the implementation of agile in a traditional company works good in many organizations and even in the hardware development process. Research has pointed out that a company with strong hierarchy can have other and more challenges than this thesis has mentioned. Therefore, it is of interest to further investigate how a hierarchical organization is impacting, when introducing agile practices and principles? What are then their challenges and is it even possible?

- In this thesis, the challenges of implementing agile in a traditional organization and possible suggestions for solutions have been found. Then it leaves space on how to implement it? How should the implementation plan look like in a short term and long term perspective? How should people be trained in the agile implementation?

- When research was made on the scrum teams at DE, it was detected that there were several product owners; one for the whole product, one responsible for the backlog in a scrum team and as well component owners responsible for components. How many owners is optimal and which responsibilities should the product owner or the product owners have? Which competences are required of this person to have?

8 References

Abrahamsson, P. (2000). Is Management Commitment a Necessity After All in III: Software Process Improvement?, In Proceedings of EUROMICRO 2000, Maastricht, The Netherlands, IEEE Computer Society, pp. 246-253.

Agile Methodology-Understanding Agile Methodology, (2008-10-23) ”Agile Methodology”, http://agilemethodology.org/ (2016-02-17).

Alänge, S. (2009). The Affinity-Interrelationship Method AIM. Lecture notes distributed in Six Sigma Black Belt course at Chalmers University of Technology, Gothenburg January 2015.

Allen, T. J. (1970). *Communication Networks in R&D Labs.* R&D Management, Vol. 1, pp. 14-21.

Bacon, G., Beckman, S., Mowery, D., and Wilson, E. (1994). Managing product definition in high-technology industries: A pilot study. California Management Review 36:32–57

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), 26-30.

Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *arXiv preprint arXiv:1205.6904*.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. et al. (2001). Manifesto for Agile Software Development. http://agilemanifesto.org/.

Beck, K. Beedle, M. Bennekum, A. Cockburn, A. Cunningham, W. Fowler, M. Grenning, J. Highsmith, J. Hunt, A. Jeffries, F. Kern, J. Marick, B. Martin, R. C. Mallor, S. Shwaber, K. Sutherland, J. (2001) History: The Agile Manifesto. Available from: http://agilemanifesto.org/history.html [2016-03-01].

Begel, A., and Nagappan, N. (2007). Usage and perceptions of agile software development in an industrial context: An exploratory study. In ESEM ‘07: First International Symposium on Empirical Software Engineering and Measurement, 255–264. Washington, DC: IEEE.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61-72.

Boehm, B. (2002). Get ready for agile methods, with care. Computer, 64-69.

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

Boehm, B. and Turner, R. (2004). Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley, Boston, MA.

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

Björkholm, T. och Brattberg H.(2010) Prioritera, fokusera, leverera: din snabbguide till Lean, Agile, Scrum och XP.

Brown, S. L. and Eisenhardt, K. M. (1995). Product development: Past research, present findings, and future directions. Academy of Management Review 20:343–378.

Brown, S., & Bessant, J. (2003). The manufacturing strategy-capabilities links in mass customization and agile manufacturing-an exploratory study. International Journal of Operations & Production Management, 23(7), 707-730.

Bryman A., Bell E., (2011). "Business Research Methods", 3rd edition, University Press, Oxford New York.

Carlsson, B. (1997). "Grundläggandeforskningsmetodik", 2nd edition, Sweden, Graphics System Ab,
55-57.

Cavaleri, S. and Obloj, K., (1993). Management Systems: A Global Perspective. Wadsworth Publishing Company, CA.

Cao, L., & Ramesh, B. (2007). Agile software development: Ad hoc practices or sound principles?. IT professional, 9(2), 41-47.

Clark, K. B., & Fujimoto, T. (1991). Product development performance: Strategy, organization, and management in the world auto industry. Harvard Business Press.

Clark, K. B. and Wheelwright, S. C. Managing New Product and Process Development. New York: The Free Press, 1993.

Collier, Ken W. (2011). Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. Pearson Education. pp. 121 ff. ISBN 9780321669544.

Cooper, R. G. (1990). Stage-gate systems: a new tool for managing new products. *Business horizons*, *33*(3), 44-54.

Cooper, R.G. (2001). Winning at New Products, 3rd ed., Perseus Publishing.

Cooper, R. G. (2011). Winning at new products: Creating value through innovation (4th Ed.). New York: Basic Books.

Cooper. G. R., (2016). Agile-Stage-Gate Hybrids: The Next Stage for Product Development. Research- Technology Management, 59(1), 21-29.

Cooper, R. G. (1994). Third-generation new product process. Journal of Product Innovation Management 11:3–14.

Cooper, R. G. & Kleinschmidt, E. J. (1994). Determinants of timeliness in product development. Journal of Product Innovation Management 11:381–396.

Cooper, R. G., & Sommer, A. F. (2016). The Agile–Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity. Journal of Product Innovation Management.

Creswell, J.W. (2009). Research design: qualitative, quantitative, and mixed methods approaches, 3rd ed., Sage, Thousand Oaks, California.

Davis, Barbee. (2012). Agile practices for waterfall projects shifting processes for competitive advantage. Publisher J. Ross Publishing Inc.

Deemer, P., Benefield, G., Larman, C. & Vodde, B. (2012). The Scrum Primer 2.0. Available from: http://assets.scrumfoundation.com/downloads/1/scrumprimer20.pdf?1352449266 [Accessed 25 July 2013].

Denscombe, M. (2000). Forskningshandboken. För småskaliga forskningsprojekt inom samhällsvetenskaperna (in Swedish), Studentlitteratur, Lund (Sweden).

Denzin, Norman K. (1978). The Research Act, 2d Ed. New York: McGraw-Hill.

Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: (2012). A Decade of Agile Methodologies: Towards Explaining Agile Software Development. *Journal of Systems and Software.* Volume 85, Issue 6, PP. 1213–1221.

Eisenhardt, K. M. and Tabrizi, B. (1996). Accelerating adaptive processes: Product innovation in the global computer industry. Administrative Science Quarterly 40:84–110.

Ericsson, (2016) Company facts. http://www.ericsson.com/se/thecompany/company_facts (2016-04-03).

Engwall, M. (2002). The Futile Dream for the Perfect Goal.

Ettlie, J. E. (1996). Global comparisons of early manufacturing involvement (EMI) in new product development. EIASM 3rd International Product Development Management Conference. Fontainebleau, France, April 15–16, 1996, pp. 275–298.

Fujimoto, T. (1997). Shortening lead time through early problem solving- A new round of capability building competition in the auto industry. International Conference on New Product Development and Production Networks, WZB, Berlin, March 20–22.

Galbraith, J. R. (1977). Organization design: An information processing view. Organizational Effectiveness Center and School, 21, 21-26.

Gilley, A., Jerry W. G., and Heather S. Mc., (2009). Organizational change: Motivation, communication, and leadership effectiveness. Performance Improvement Quarterly 21.4, pp. 75-94.

Gupta, A. K. and Wilemon, D. L. Accelerating the development of technology-based new products. California Management Review 32: 24–44 (1990).

Haddad, C. (1997). Involving manufacturing in the early stages of product development: A case study from the North American automotive industry. International Conference on New Product Development and Production Networks, WZB, Berlin, March 20–22.

Hart, S. (1993). Where we've been and where we're going in new product development research. Workshop on Meeting the Challenges of Product Development, Groningen, The Netherlands, May 15–18.

Highsmith, J. (2001) History: The Agile Manifesto. Available from: http://agilemanifesto.org/history.html [Accessed 21 February 2013].

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. Computer, 34(9), 120-127.

Highsmith, J. (2002). Agile Software Development Ecosystems, Addition-Wesley.

Highsmith, J. (2003). Cutter Consortium Reports: Agile Project Management: Principles and Tools 4, 2, Cutter Consortium, Arlington, MA.

Hines, P., Holweg, M., & Rich, N. (2004). Learning to evolve: a review of contemporary lean thinking. International journal of operations & production management, 24(10), 994-1011.

Hayes, R. H., Wheelwright, S. C., and Clark, K. B. Dynamic Manufacturing. New York: The Free Press, 1988.

Iansiti, M. (1993). Real-world R&D: Jumping the product generation gap. Harvard Business Review May-June: 138–147.

Iansiti, M. and Mac Cormack, A., (1996). Developing products on Internet time. Working Paper 97-027, Harvard Business School, Boston, MA.

Johansson, C., and Mats H., (2008). Speaking of change: three communication approaches in studies of organizational change. Corporate Communications: An International Journal 13.3, pp. 288-305.

Jürgens, U. (1995). Anticipating problems with assembly in the early stages of product development. Third International Workshop on Assembly Automation, Venice, October 12–14.

Karlstrom, D., and Runeson, P. (2005). Combining Agile methods with Stage-Gate project management. IEEE Software 22(3): 43–49.

Karlstrom, D., and Runeson, P. (2006). Integrating Agile software development into Stage-Gate managed product development. Empirical Software Engineering 11:203–225.

Khurana, A. and Rosenthal, S. R. (1998). Towards holistic “front ends” in new product development. Journal of Product Innovation Management 15:57–74.

Hays, P. A. (2004). Case study research. Foundations for research: Methods of inquiry in education and the social sciences, 217-234.

Highsmith, Jim & Cockburn, Alistair. (2001). Agile Software Development: The Business of Innovation. Computer Science Department, University of Southern California, Los Angeles. PP. 120-122.

Jick, T. D. (1979). Mixing qualitative and quantitative methods: Triangulation in action. Administrative science quarterly, 602-611.

Larman, C. and Vodde, B. (2009). Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum, Pearson Education: Boston.

Layton, Mark C. (2015). Scrum for Dummies. Publisher; John Wiley & Sons (US). CH. 1http://library.books24x7.com.proxy.lib.chalmers.se/toc.aspx?bookid=82533.

Liker, J. K. and Meier, D. (2006). “The Toyota Way Fieldbook: A Practical Guide for Implementing Toyota's 4Ps”. New York, McGraw-Hill. Chapters 4, 5 and 6.

Liker, J. K., & Morgan, J. M. (2006). The Toyota way in services: the case of lean product development. The Academy of Management Perspectives, 20(2), 5-20.

Liker, J. K., Kamath, R. R., Wasti, S. N., and Nagamachi, M. (1996). Supplier involvement in automotive component design: Are there really large US-Japan differences? Research Policy 25:59–89.

Locke, K. (2001). Grounded Theory in Management Research. London: Sage.

Maylor, H., (2010) Project Management 4th ed., Harlow: Pearson Educational Limited.

Modig, N., & Åhlström, P. (2012). This is lean. Halmstad, Sweden: Rheologica.

Moe, N. B., Dingsoyr, T., & Dyba, T. (2008, March). Understanding self-organizing teams in agile software development. In Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on (pp. 76-85). IEEE.

Monden, Y. (2011). Toyota production system: an integrated approach to just-in-time. CRC Press.

Naylor, J. B., Naim, M. M., & Berry, D. (1999). Leagility: integrating the lean and agile manufacturing paradigms in the total supply chain. International Journal of production economics, 62(1), 107-118.

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

Nevis C. E., (1987). Organizational Consulting: A gestalt Approach, Gestalt Institute of Cleveland Press, pp. 141-157.

Nyman, M. (2010). Agila metoder: Radikal revolution eller enkel evolution.

Olsson, H. H., Alahyari, H., & Bosch, J. (2012). Climbing the" Stairway to Heaven"-A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. In Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on (pp. 392-399). IEEE.

Palhan, S.K (2007). Managing projects, Slough: New Dawn Press

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. In Product-focused software process improvement (pp. 386-400). Springer Berlin Heidelberg.

Popli, r., & chauhan, n. (2013). A mapping model for transforming traditional software development methods to agile methodology. International journal of software engineering & applications,4 (4), 53, 57, 58.

Reagan, B. (2012). Going Agile: CA Technologies, Clarity PPM Division's transformative journey. Digital Celerity, San Francisco, CA, September 22. http://www.slideshare.net/DCsteve/going-agile-with-ca-clarity-ppm-agile-vision.
Rosenthal, S. R. (1992). Effective Product Design and Development. Homewood, IL: Business One Irwin.

Sommer, A. F. Dukovska-Popovska, I., and Steger-Jensen, K. (2014). Agile product development governance—On governing the emerging Scrum/Stage-Gate hybrids. In Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World: International Conference, APMS 2014, Proceedings, Part I, 184–191. Berlin: Springer-Verlag.

Smith, H. W. (1975) Strategies of Social Research: The Methodological Imagination. Englewood Cliffs, NJ: Prentice Hall.

Schwaber, Ken. & Sutherland, Jeff. (2013). Scrumguiden.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. Informatica Economica, 17(4), 64.

Strauss, A., & Corbin, J. (1990). Basics of qualitative research (Vol. 15). Newbury Park, CA: Sage.http://www.li.suu.edu/library/circulation/Stein/Comm%206020ksStraussCorbinBasicsQualitativeFall07.pdf

Stake, R.E. (1995). The Art of Case Study Research. Thousand Oaks, CA: Sage

Techjini. Process Methodologies. Available from: http://www.techjini.com/jp/ourapproach-methodologies.html [Accessed 6 April 2016].

Timmermans, S., & Tavory, I. (2012). Theory construction in qualitative research from grounded theory to abductive analysis. Sociological Theory,30 (3), 167-186. http://grap.ulb.ac.be/wp-content/uploads/Timmermans-and-Tavory_Abductive-Analysis.pdf.

Trygg, L. (1991). The need for simultaneous engineering in time-based competition. In: Technology Management: Technology Strategies and Integrated Information Systems in Manufacturing. J. Ranta and S. Å. Hörte (eds.). Helsinki, Finland.

Ulrich, K. T., and Eppinger, S. D. (2008). Product design and development. Fourth Edition. New York.

Verganti, R. (1977). Balancing between feedforward and feedback planning for integrated product development. EIASM 4th International Product Development Management Conference, Stockholm, Sweden, May 26– 27.

Verganti, R. (1997). Leveraging on systemic learning to manage the early phases of product innovation projects. R&D Management 27:377–392.

Verganti, R., MacCormack, A., and Iansiti, M. (1988). Rapid learning and adaptation in product development: An empirical study of the Internet software industry. EIASM 5th International Product Development Management Conference, Como, Italy, May 25–26.

Verganti, R. (1999). Planned flexibility: linking anticipation and reaction in product development projects. Journal of Product Innovation Management,16(4), 363-376.

Veryzer, R. W. (1998). Discontinuous innovation and the new product development process. Journal of product innovation management, 15(4), 304-321.

Vizteams, (2013). Weeding out buzzword agile from real agile. Available from: http://www.vizteams.com/blog/weeding-out-buzzword-agile-from-real-agile/ [Updated at 2 June 2016]

Webb, Eugene J., Donald T. Campbell, Richard D. Schwartz, and Lee Sechrest (1966). Unobtrusive Measures: Nonreactive Research in the Social Sciences.

Wells, D. (2009). A project heartbeat. The Agile Process. Available from: http://www.agile-process.org/heartbeat.html. [Available at 6 June 2016].

Wilson, Chauncey (2014). Interview Techniques for UX Practitioners: A-User-Centered Design Method. Chapter 2- Semi-Structures Interviews.

Wheelwright, S. C. and Clark, K. B (1992) Revolutionizing Product Development, The free press, United States of America.

Womack, J. P., Jones, D. T., & Roos, D. (1990). Machine that changed the world. Simon and Schuster.

Appendix A - Interviewees

A:1 Internal interviewed people

- Scrum member with previous experience as scrum master.
- Global Process Manager and involved in an agile transformation project called EvolvE.
- Component owner in hardware.
- Engineering task leader (ETL, similar to project leader) for both hardware and software.
- Scrum master in an agile team.
- Engineering task leader for software (ETL, similar to project leader)
- Product owner in a scrum team and component owner of both hardware and software.
- Product owner in software and component owner in software
- Agile coach, consult.
- Scrum member.
- Product owner in software, campion in Evolve, member of a research team called Software-Center.
- Supplier from Delphi.
- Product owner and component owner in software.
- Scrum master in an agile team.
- Group chief at another close department to driver electronics.
- Component owner for hardware.
- Component owner for hardware.
- Project Manager and agile transformational leader.

A:2 Benchmarking companies attendants

- Software Manager at Saab Surveillance.
- Sector Manager at Ericsson in hardware development.
- Change agent at Ericsson
- Team Leader at CPAC

A:3 External Workshop attendants

- Group manager at Driver Electronics GTT
- Global Process Manager and involved in an agile transformation project called EvolvE.

- Product owner in software, campion in Evolve, member of a research team called Software-Center.
- Operative Technology Strategy Responsible.
- Chief at Saab Surveillance in software development.
- Change agent at Ericsson.

A: 4 Internal Workshop attendants

- Component owner for hardware.
- Component owner for hardware.
- Product owner in a scrum team and component owner of both hardware and software.
- Engineering task leader (ETL, similar to project leader) for both hardware and software.
- Group manager at Driver Electronics GTT.
- Global Process Manager and involved in an agile transformation project called EvolvE.
- Group chief at another close department to driver electronics.

Appendix B - Interview template for GTT

- Name, work title/role, can you explain what you do?
    - Do you have an agile role? Which?

- What is agile for you?

- Can you please describe the product you are working with?
    - Does it exist of both hardware and software?
    - Is the product on the platform?
    - Do you consider that your product is complex and can you please define what you mean with complex?
    - If you have a hardware product, how long is the lead time of the development?

- What are your perception of working agile in software development?
    - What are the pros and cons according to you?
    - Have you faced any challenges? Which?
    - Can you compare the time before agile was implemented, when you worked according to the traditional way? Are there any differences from now?

- What do you see as the challenges/problems/conflicts of introducing agile teams into a traditional organization?

- What do you see as the biggest challenges/problems/conflicts today when working agile in software development and not in hardware development, have any challenges emerged since the change to agile software development, according to you? Exemplify!
    - In what way do you think these challenges can be managed?

- What is the strict deadline for software development, according to you?

Agile in hardware development

- Have you worked in a team that have used the waterfall model and stage-gate-process? In that case:
    - In what kind of project?
    - What were the pros and cons?

- Have you faced any challenges/problems in the hardware development process?
    - What do you see as the bottleneck in the hardware development process?
    - Do you see any potential improvement possibilities in the hardware development process today, according to you? In that case, what

improvements?

- If you have not faced any challenges/problems/conflicts with the hardware development process, can you please describe the process? What are you doing? What are the pros and cons?

- What elements of agile do you see as possible to use in the hardware development process and where in the process?
    - How do you think these elements can be applied in the hardware development process?

- What could be the risks of implementing agile elements into the hardware development process?

Relationship with supplier

Software

- Do you have external supplier for the software development?
    - If yes, how does the process between the software developer and the supplier look like/works today?

- If the software development process is agile and you have external supplier:
    - How will it affect the work between the software developers and the external supplier?
    - What challenges could emerge between them?
    - How do you think the challenges can be managed?

Hardware

- Do you have external supplier for the hardware development?
    - If yes, how does the process between the hardware developer and the external supplier look like today?

- If the hardware development process is agile and you have external supplier:
    - How will it affect the work between the hardware developers and the external supplier?
    - What challenges could emerge between them?
    - How do you think the challenges can be managed?

Appendix C - Interview template for Ericsson

A. Name, work title/role, can you explain what you do?
    a. Do you have an agile role? Which?
B. Company background, can you please explain about your company?
C. Can you explain what relation Ericsson has to agile?
D. What is agile for you?
E. Which agile knowledge do you want to have?
F. How did you get the lean/agile knowledge?

Background

G. To what extent are you working with agile/lean at this department?
    a. Which parts of the agile/lean methodology are used?
    b. How strictly are the agile/lean parts followed?

H. What information and education did the employees at the department take part of in connection with the agile introduction?
    a. Who did take part of the information and education in the working group?
    b. Have other departments got agile training/knowledge in connection with the agile implementation?

I. How was the agile implementation performed?
    a. Which steps were included in the change management process to achieve agile/lean hardware development?
    b. Which changes did you do at the department when you changed?

J. Tell us about your process of implementing agile into software development and lean into hardware development:
    a. What was the outcome of your implementation?
    b. What would you have done differently today?

K. Can you please describe the product you are working with?
    a. Does it exist of both hardware and software?
    b. Is the product on the platform?
    c. Do you consider that your product is **complex** and can you please define what you mean with complex?

L. If you have a hardware product, how long is the lead time of the development?

M. What are your perception of working agile in software development?
    a. What are the pros and cons according to you?
    b. Have you faced any challenges? Which?

c. Can you compare the time before agile was implemented, when you worked according to the traditional way? Are there any differences from now?

N. What do you see as the challenges/problems/conflicts of introducing agile teams into a traditional organization?

O. What do you see as the biggest challenges/problems/conflicts today when working agile in software development and **not in hardware development,** have any challenges emerged since the change to agile software development, according to you? **Exemplify!**
    a. In what way do you think these challenges can be managed?

P. What has to be finished in the software and in which time/phase in relation to hardware development parts that are included in the product?

Agile in hardware development

Q. Have you faced any challenges/problems in the hardware development process/ What do you see as the bottleneck in the hardware development process?
    a. Do you see any potential improvement possibilities in the hardware development process today, according to you? In that case, what improvements?

R. What elements of agile do you see as possible to use in the hardware development process and where in the process?
    a. How do you think these elements can be applied in the hardware development process?
    b.
S. What could be the risks of implementing agile elements into the hardware development process?

Relationship with supplier

Software

T. Do you have external supplier for the software development?
    a. If yes, how does the process between the software developer and the supplier look like/works today?

U. If the software development process is agile and you have external supplier:
    a. How will it affect the work between the software developers and the external supplier?
    b. What challenges could emerge between them?
    c. How do you think the challenges can be managed?

Hardware

V. Do you have external supplier for the hardware development?
    a. If yes, how does the process between the hardware developer and the external supplier look like today?

W. If the hardware development process is agile and you have external supplier:
    a. How will it affect the work between the hardware developers and the external supplier?
    b. What challenges could emerge between them?
    c. How do you think the challenges can be managed?

Appendix D - Internal Workshop

The internal workshop was a brainstorming session in a cross-functional team, which was aimed to identify what possibilities there are to become more agile in the hardware development process. The AIM was used to facilitate the data collection of the workshop and seven employees from GTT were attended.

After the AIM workshop execution, five main groups were identified. These were: (1) *Classic agile development*, (2) *Controlled leadership*, (3) *Work contract with suppl*iers, (4) *Product over projects,* (5) *Hardware organization.* The different groups are summarized and presented below. In some of the main groups a few sub-groups were established where each of them will be presented as well. After each group there is a small reflection based on the discussions that was held together with the attendances after the workshop.

1.Classic agile development

In this group four sub-groups were established:

Create clear product ownership who has authority and responsible
To create clear product ownership did get much attention during the workshop. This implies that the manager should clearly point out a product owner. Further, the manager should give the product owner the authority to make the decisions needed for the product. In this way, product delays will be avoided. Moreover, clear roles and responsibilities should also be distributed within the cross-functional teams including authority and responsibility.

"Living document" – Pace, continuous planning and follow-up at lower levels
It was emphasized that a cadence should be introduced to the development process, which everyone within the organization is able to follow. The cadence should be built on multiples since all products have different lead-times and both hardware and software products should be able to adapt this. Moreover, they wanted to implement a pace in the time plan, telling at what time different products should have deliveries.

It was also brought up to lower the amount of documentation, and to introduce daily planning rather than detailed planning as well as implement daily follow-up in order to get continuous feedback from stakeholders etc.

What decisions we must take and when - Life cycle model based on what time different parts must be finished
In order to foster agile development the hardware product should be developed in sections. The basic functionalities that require early verification can be defined early and the remaining parts can be implemented according as in order to let the hardware product be more flexible so it provides possibilities for unexpected changes.

Define cross-functional teams with the right competence and need for communication in different phases

All attendances agreed on introducing cross-functional teams, including roles such as: mechanical engineers, designers, packaging suppliers, component owners and possible support functions. Define core teams that are persistent and have authority.

Use cross-functional teams in the early concept phase, where the aim is to reach decided goals such as appearance, packaging and attributes. Include the supplier in the teams, though it requires both that the customer must strive for creating a partnership with the supplier as well as an agile contract that must be signed from the both partners (the partnership and contracts are included in another group as well). Even though, the suppliers would not accept the agile contract, only employees from GTT can be involved in the team. However, the suppliers will provide knowledge that might be helpful in the concept phase, telling what is possible or not.

A challenge is to create teams with people located at different sites. It was brought up that maximum 2 sites within one team should be acceptable since meetings should be able to book within the same time zone.

Furthermore it was stated that the hardware developer and software developer should have closer cooperation, in order for both partners to increase their understanding of each other's areas.

After the workshop the process manager stated that the activities brought up in this group did have a direct link to the Evolve initiative at GTT and could be directly implemented to the company already this year in order to fulfill parts of the goals stated in Evolve, therefore these activities were not only aimed for the hardware development. See the sup-groups below.

2. Controlled leadership:

In this group there was only one single group, but that was highly prioritized.

The controlled leadership should be replaced with a leadership built on trust and further, the authority should be pushed down to lower level, which is also dependent on trust from the leadership.

The reflection after the workshop was that a change from a "controlled based leadership" to a "trust leadership" is crucial and a requirement in order to be agile, but it will take several years to reach this since it means a cultural change. This cultural change will take at least 10-15 years to create.

3. Work contract with suppliers

In this group, three subgroups were established. Since external suppliers are utilized for the development of the hardware product it is suggested to get the supplier on board to start utilizing agile methods in order to get the benefits out of agility throughout the entire development process. In doing so, agile contracts must be discussed and established with the supplier, therefore suggestion of what could be included in the contract was brought up during the workshop and what suggestions they had in order to deal with different issues.

Flexible and procurement model
Set up an own agile model over the budget. Instead of paying for packages, pay for working hours. E.g. GTT acts as the product owner and pays for the hours spent from the supplier and the supplier has continuously demos.

In order to make the payment more flexible, it was suggested to include in the procurement that only the known requirements should be included in the budget at the beginning, and the unknown requirements that comes in later can be paid according as.

In order to have a more flexible development and welcome new incoming requirements and needs, it could be discussed in the procurement of what changes can be done without giving any impact on the lead times and the costs.

Contracts that managing flexibility
Today the flexibility in the contracts with the suppliers is limited and must be changed, since it is not durable in today's environment. In order to be more flexible regarding the product requirements during the development process, discuss with the supplier to gradually build up the specification, instead of stating everything at the beginning and include this in the contract. Moreover, it was also suggested here to discuss payment per working hour rather than packages.

Partnership with the supplier and "living documentation"
Use the term partners rather than suppliers, create partnership and build trust. Create more unspecified contracts including close cooperation between the customer and the partner. As a suggestion the specification should be developed together with the partner in order to save time in later phases, where the supplier usually must translate the requirements into their language.

The reflection after regarding this group was that there is no requirement to involve the supplier in the agile transformation as a first step just because GTT introduce agile methods into their hardware development process. Although, if the supplier is going to be involved, agile contracts must be established. However, as a first step build on the partnership with the supplier and secondly, discuss the content of the statement of work (a document where

project activities are defined and signed by both partners) with the supplier. Thereafter, involve the supplier in the development process, e.g. in the concept phase.

4. Product over projects

Only one group was identified:

In the long term it is desired to that the focus on projects disappear. Therefore, the today's project focus should stepwise be replaced with product focus.

It was claimed after the workshop, that in order to become an agile organization it is a prerequisite that the project focus disappear.

4. Hardware organization

In this group, only one subgroup was identified:

Find the best way of working and the best knowledge transformation in the hardware development

Do the entire hardware development in-house or buy a whitebox, meaning that the supplier has transparency in their development.

Create cross-functional teams with only hardware developer and put all hardware developers closer to each other.

Find alternative ways to validate and verify the hardware products, today there are too long lead-times on testing.

There was some discussion regarding how to place the hardware developers, some thought that only hardware developers should sit together and others thought that a mix of hardware and software developers was more appropriated.

The final reflection

"Who is the product owner?" was a repeated problem during the workshop. Who the product owner is not clearly defined at the moment and that makes people frustrated and confused. Also, since agile teams were introduced, one more product owner appeared within the team, more confusion appeared.

The group that got most attention was the classic agile development group, which can directly be included into the EvlolvE initiative. However, it was also concluded that in order to become an agile organization in a long-term perspective, the projects must be replaced and the leadership must build on trust rather than control.