Institutionen för datavetenskap
Department of Computer and Information Science

Master Thesis project

Critical success factors in Agile software development projects

by

Larsson, David and Walander, Tomas

LIU-IDA/LITH-EX-A--15/023--SE

2015-06-16
Master Thesis project

Critical success factors in Agile software development projects

by

Larsson, David and Walander, Tomas

LIU-IDA/LITH-EX-A--15/023--SE

2015-06-16

Handledare: Kristian Sandahl
Examinator: Ola Leifler
Abstract

The demand for combining Agile methodologies with large organizations is growing as IT plays a larger role in modern business, even in traditional manufacturing companies. In such organizations, management feel they are losing the ability to plan and control as the developers increasingly utilize Agile methodologies. This mismatch leads to frustration and creates barriers to fully Agile software development. Therefore, this report aims to evaluate what factors affect Agile software development projects in an organizational context, and in particular how these factors can be monitored by the effective use of measures.

This master thesis project has conducted a case study at Scania IT, a subsidiary of truck manufacturer Scania, as well as an extensive literature review, which together help identify several critical success factors for combining Agile methodologies with an organization.

The report concludes that several aspects are important when agility is introduced to a functional organization and also when combined with a project stage gate model. Moreover, it was found that measures, in particular software metrics, can greatly aid the organization in overcoming several organizational barriers. However, to succeed, corrective actions must be defined that help the organization prevent the measure from becoming yet another statistic data, but rather learn and improve its way of working.
Acknowledgements

This study was possible through hard work and extensive help and support of the following dedicated people. We owe them big time.

Firstly, we send our gratitude to our examiner, Ola Leifler, and supervisor, Kristian Sandahl, at Linköping University, as well as, the opponent of this report, Andreas Pierrou.

Secondly, great thanks to Scania IT, the employees at the Project Management Office, our mentors, Markus Töreholt and Marie Sandegren.

Finally, we thank all interviewees who contributed with their time and knowledge to the case study.
Table of Contents

1. **Introduction** ................................................................................................................. 4
   1.1 Background .................................................................................................................. 4
   1.2 Scope ............................................................................................................................ 5
       1.2.1 Research questions .............................................................................................. 6
       1.2.2 Limitations of research ....................................................................................... 6
   1.3 Case study ..................................................................................................................... 7
   1.4 Disposition .................................................................................................................... 7

2. **Method** ......................................................................................................................... 9
   2.1 Research design ........................................................................................................... 9
       2.1.1 Research questions .............................................................................................. 9
       2.1.2 Pre-study ............................................................................................................ 9
       2.1.3 Theory ................................................................................................................ 10
       2.1.4 Empirical studies .................................................................................................. 10
       2.1.5 Analysis and conclusions ...................................................................................... 15
   2.2 Credibility of the study ............................................................................................... 17
       2.2.1 Validity ................................................................................................................ 17
       2.2.2 Reliability .......................................................................................................... 18

3. **Theory** .......................................................................................................................... 19
   3.1 Agile methodologies in an organizational context ..................................................... 19
       3.1.1 Traditional project methods ................................................................................. 19
       3.1.2 The Agile approach ............................................................................................. 19
   3.2 Barriers when combining Agile with the organization ............................................... 24
       3.2.1 Organizational Layout ......................................................................................... 25
       3.2.2 Combining Agile with a stage gate model ........................................................... 26
       3.2.3 Project portfolio management and resource allocation ..................................... 27
       3.2.4 Knowledge about Agile methodologies .............................................................. 29
       3.2.5 Working with requirements .................................................................................. 29
       3.2.6 Agile teams in an organizational context ............................................................ 30
   3.3 Measures ....................................................................................................................... 35
       3.3.1 Software Metrics .................................................................................................. 35
       3.3.2 Project Scatter Factor ......................................................................................... 36
       3.3.3 How to use measures effectively ........................................................................... 37

4. **Empirical Findings** ...................................................................................................... 41
   4.1 Background to Scania IT ............................................................................................ 41
       4.1.1 The parent company Scania CV AB ..................................................................... 41
       4.1.2 Overview of Scania IT ........................................................................................ 41
       4.1.3 Project Success and the project parameters ....................................................... 45
       4.1.4 Agile at Scania IT ............................................................................................... 45
       4.1.5 A new project arrives .......................................................................................... 46
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2.1</td>
<td>Project manager and Group manager</td>
<td>106</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Scrum Master</td>
<td>108</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Product owner</td>
<td>110</td>
</tr>
<tr>
<td>8.2.4</td>
<td>Developer</td>
<td>112</td>
</tr>
</tbody>
</table>
1. Introduction

*The following section gives brief overview of the background, organization and scope relevant to the study in this report.*

1.1 Background

The purpose of this chapter is to give an overview of the history of and reasons for the Agile approach and outline where this study fits in this context.

According to Schmidt et al. (2001) the never-ending change in market needs and technology makes it continuously troublesome for software developers to meet customer requirements and, moreover, respond to changes. Agile software development has recently emerged as the best available practice with regards to allowing software teams to embrace and respond to this changing environment (Coad et al. 1999; Schwaber and Beedle 2002; Stapleton 1997). In fact, the term “agility” is commonly used to denote the ability to embrace and respond to changes (Erickson et al., 2005; Henderson-Sellers and Serour, 2005; Larman, 2004; Qumer and Henderson-Sellers, 2008).

Since the creation of the Agile manifesto in 2001, research has devoted great attention to Agile software development (Dingsöyr, 2012). For example, Cohn and Ford (2003) and Lindvall et al. (2004) highlight four factors that affect the overall perception of success of a software project;

- quality, i.e., delivering a good working product;
- scope, i.e., meeting all requirements by the customer;
- timeliness, i.e., delivering on time; and
- cost, i.e., required resources and effort.

The latter two are also addressed by Melo et al. (2013) when they state that management of software development productivity is a key issue in software organizations, where the major drivers are lower cost and shorter time-to-market. Thus, it is alarming when the Chaos study, performed by the Standish Group, showed that 26 percent of all software projects fail, and 46 percent experience cost and schedule overruns or significantly reduced functionality (Reel, 1999).

In literature the term critical success factors are commonly used for business, and has recently been applied to Agile software projects as is argued in the following two paragraphs. For example, Bullen and Rockhart (1981) define critical success factors as:

“The limited number of areas in which satisfactory results will ensure successful competitive performance for the individual, department, or organization. Critical success factors are the few key areas where ‘things must go right’ for the business to flourish and for the managers’ goals to be attained” (Bullen and Rockhart p.385, 1981).
Furthermore, Chow and Cao (2007) report that problem areas can be classified into four categories: organizational, people, process and technical. Similarly, Agile project success factors can be classified into five categories: organizational, people, process, technical and project.

Critical success factors in software projects do not differ significantly from fundamental project management techniques (Reel, 1999). Additionally, they are found to relate to a combination of software engineering and business strategy (Bytheway, 1999). As stated by Boehm and Ross (1989), the primary problem in project coordination is the number of parties with interest in the software project’s progress; users, customers, development teams, maintenance and management are simultaneously demanding the project to satisfy their needs. Therefore, Agile methodologies should be tailored for each project’s distinct situation, in order to achieve maximum effect (Fitzgerald et al., 2006).

In relation to critical success factors, Dingsöyr et al. (2012) aggregate the result of several articles on Agile project success, and hence, establish five categories of success factors in Agile software development;

- organizational, e.g., mismatch in organizational culture;
- people, e.g., knowledgeable team members;
- process, e.g., strong customer commitment;
- technical, e.g., well-defined coding standards up-front;
- and project, e.g., project type being of variable scope with emergent requirement.

However, according to Melo et al. (2013), little empirical research examining which factors do have an impact on agility. Moreover, Fruhling and De Vreede (2006) and Moe et al. (2008) both report that, although the Agile approach gains increasing support among organizations and many have claimed its benefits, little empirical evidence exist that shows if, how, and why Agile development is effective.

Another aspect, presented by Fenton and Neil (2000), is the field of software metrics that has for a long time been driven by the urge to forecast resources and quality in software development projects. Moreover, the authors argue that most existing software metrics are motivated by one of two activities: (1) the desire to assess or predict effort/cost of development processes; and (2) the desire to assess or predict quality of software products.

### 1.2 Scope

This section establishes the aim of this report through posing research questions to be answered, as well as, stating known limitations to the study.

In conclusion, the background above has described the eminent need for further research on critical success factors in Agile software development projects, what organizational parameters affect them and, particularly, how those critical factors can be measured to estimate project outcome.
Therefore, the objective of this report is to, based on a literature and case study, evaluate what factors affect Agile software development projects in an organizational context.

### 1.2.1 Research questions

To complete the objective described above, the following research questions have been formulated:

- **RQ1.** How is Agile software development projects affected when the surrounding organization regards IT services as support for its core business?
- **RQ2.** How does a traditional project stage gate model affect Agile software development projects?
- **RQ3.** How does resource allocation in an IT organization affect Agile software projects?
- **RQ4.** How is the field of performance measures, and particularly software metrics, used to concurrently communicate status and facilitate decision making in Agile software development projects at the case company? and how does project workers and managers suggest that it could be used in the future?.

### 1.2.2 Limitations of research

Apart from the research questions above, that form a natural limit for the scope of this report, the limitations herein and set forth are necessary to allow the authors to analyze theory and case study and draw conclusions, which would not be feasible without clearly defining the range of the study. Moreover, limitations will help the reader determine whether any drawn conclusions of this report are applicable in other situations.

Firstly, the study is focused on projects developing software. i.e., projects regarding IT management processes, maintenance, or buy/rent projects are not considered part of the scope. The main reason for this limitation is that maintenance, process and buy/rent projects face different circumstances regarding time frame, customer involvement and planning.

Secondly, projects that are part of the study was selected on the premise that they had a clear directive to apply Agile practices, i.e., the studied projects strived to be, but were not necessarily completely Agile. Since, the study aims to determine what factors have an impact on productivity in Agile software development projects, it is reasonable to only study projects with a spoken Agile policy. However, the level of agility may very well be one of the factors that have an impact on project success and productivity, it is not mandated that the team succeeds in working according to an Agile methodology.

Thirdly, as the case company lives in a manufacturing industrial context that commonly applies Lean principles it is natural for some Lean processes to have affected the case company as well. Although, there are similarities between Agile and Lean methodologies, e.g., focusing on continues improvement, Lean principles will not be covered in depth in this study.

Fourthly, this report aims solely to study obstacles and complications of working according to Agile methodologies at an IT organization in a larger manufacturing industrial context. i.e., the
research will not cover how Agile methodologies are actually used but rather what factors could affect productivity in Agile projects.

Finally, Agile methodologies have its origins in Lean principles (Wang et al., 2012; Smith, 2007). Moreover, Pernstål et al. (2013) and Wang et al. (2012) argue that Lean could be a suitable complement to Agile methodologies' poor performance to meet the industries need for scaling Agile software development to an organizational level. However, Lean principles in themselves are not further investigated in this report.

The limitations described above have, mainly, been done due to the fixed time frame of twenty weeks to complete this master thesis. Moreover, the study have to be limited due to the case study only considering the situation at one case company.

In addition, this report does not cover the foundation of Agile methodologies nor an in depth description of the various frameworks that exist. Thus, the reader is expected to have a basic understanding of Agile methodologies, although some of the more important concepts will be briefly outlined in the literature review.

1.3 Case study
This report is mostly based on a literature study that has broadly covered the field of Agile methodologies, although, focusing on the area of success factors on Agile software development and relevant metrics that help predict project outcome. To provide more real-world related analysis the theoretical review has been complemented with a case study of software development projects at Scania IT, a subsidiary of Scania CV AB, that will be described in more detail in section 4.1 Scania IT.

1.4 Disposition
The report is divided in five chapters through which the scope and research questions are treated. Each of the chapters are described briefly below.

The Method chapter describes the approach used in this study, i.e., the research methods used along with a critical review of advantages and disadvantages. Alternative methods are also shortly presented.

Chapters 3 through 5, Theory, Empirical findings and Analysis, present the reviewed literature, the result obtained from the case company; and an analysis of the two respectively. To guide the reader through the report, those three chapters are similar in structure. i.e., for level two and three all headlines of those three chapters coincide, with the exception that 3.1 and 4.1 are reserved for relevant background descriptions of the literature review and empirical findings respectively. Table 1.1 below shows how the major headlines of chapters 3, 4 and 5 coincide. For example, headline 3.3.1, 4.3.1 and 5.3.1 all cover the same topic of the study, namely Software Metrics.
### Table 1.1: Relation between major headlines in chapters 3 through 5.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Theory</th>
<th>Empirical findings</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barriers when combining Agile with the organization</td>
<td>3.2</td>
<td>4.2</td>
<td>5.1</td>
</tr>
<tr>
<td>Measures</td>
<td>3.3</td>
<td>4.3</td>
<td>5.2</td>
</tr>
</tbody>
</table>

The final part, Conclusions, summarizes the analysis and draw conclusions with the sole aim to answer the given research questions. Furthermore, the last part of chapter 7 has been dedicated to a review of the result which will try to critically argue whether the conclusions should be deemed reliable or not, as well as, try to point out areas for further research related to, but outside the scope of, this report.
2. Method

The purpose of following chapter is to describe the methods used by the authors to complete this academic report. The choices made regarding used processes and methods are motivated, and furthermore, a description of the process for reaching a certain level of credibility is presented.

2.1 Research design

The following section briefly describes how the research was carried through three phases: pre-study, theory and empirical studies.

2.1.1 Research questions

At an early stage of this master thesis project, the authors decided upon which research questions to study. However, as the author’s knowledge of the theoretical area and the involved case company evolved, the relevance of the initial research questions decreased. The authors realized that the time frame of the project and the layout of the ongoing case study were insufficient to answer the initial research questions, which motivated a slight modification in scope. As a result, the research questions were rephrased mid-way during the master thesis project and the final versions cover a narrower field of study and are better suited for the purpose of this study. Below are the initial research questions listed in order to show what changes have been made;

1. how is agile software development projects affected by pre-determining the traditional project steering parameters; such as scope, quality, time and cost?;
2. how is an IT organization, with only manufacturing customers, affected by allowing traditional project steering parameters to change throughout its agile software development projects?;
3. what critical success factors affect agile software development project productivity and customer satisfaction?; and
4. what performance measures could be used to concurrently visualize productivity and customer satisfaction in agile software development projects before the final result?.

2.1.2 Pre-study

The purpose of the pre-study is to determine the area of interest for this report. There are two fundamental parts of the pre-study. Firstly, a literature review to get an understanding of the subject. Secondly, an empirical pre-study at the case company to gain knowledge of the general situation and identify possible problem areas for the case study. The empirical pre-study was carried out by interviewing stakeholders, with the purpose of understanding their expectations of the master thesis project. I.e., the case company, and particularly the management team, was included when choosing the field of study. However, after the scope was determined the individual employees’ expectations were given limited, if any, consideration. The empirical pre-study was mostly conducted through informal interviews and, moreover, complemented by
participation in various areas of the daily business; e.g., group meetings, to gain further knowledge of case company's way of working.

2.1.3 Theory
As reported by Backman (2008), the literature review's intent is to provide an overview of, and a historical perspective on, research connected to the field of concern. Furthermore, the review gives insight into current research and serves as assistance in defining the problem area (Backman, 2008).

The primary source for the theoretical study was academic journals. The academic journals are authored and reviewed by researchers within similar areas, which increases their credibility. A wide selection and detailed description of academic journals increases the validity of the study and, further, provides an accurate historical perspective, as mentioned by Backman (2008). The list below outlines the various search phrases that were used in the literature review, and often a combination of the search phrases were used to narrow down the results.

- Agile methodologies
- Agile organizations
- Agile organizational context
- Agile software product development
- Critical Success Factors
- Performance Measures (key performance indicator)
- Software metrics (resource metrics, product metrics and process metrics)
- Software product development
- Resource allocation

Figure 2.1 visualizes how the authors chose to process the literature. Firstly, the authors read the academic journal and extracted relevant information as bullet points. Secondly, the extracted bullet points were categorized by topic, which resulted in an overview of each topic and eventually gave the authors an understanding for relations between the academic journals and assisted in defining the problem area. Furthermore, the categorization assisted in contrasting the contents and deciding an initial disposition of the theory chapter.

![Figure 2.1: The literature review process](image)

2.1.4 Empirical studies
On arrival at the case company, the authors signed a non-disclosure agreement proposed by the case company. According to Melo et al. (2013), this is an important step to establish a formal link between the researchers and the case company resulting in them feeling more comfortable
with the researchers presence observing the internal activities. In the end the agreement did not cause the authors to remove topic from this report. However, the interviewed projects’ assignment may be subject to the non-disclosure clause, why the authors have chosen not to reveal details about the individual project.

2.1.4.1 Selection of projects
From the literature review the authors defined the purpose of the study and, further, specified the purpose through four research questions. Concerning these research questions, certain constraints on the Projects subject to the case study was necessary;

- the project aims to work according to Agile methodologies;
- the project is a software development project, i.e., enhancement of an existing system, development of an entirely new system or integration of an existing systems are projects qualified for the case study. However, projects aiming at redesigning a process are not eligible;
- the project is ongoing;
- the project is an internal project at Scania IT, meaning that it is not outsourced to an external party; and
- positive towards reserving adequate time for the authors' case study.

In other words, the requirements listed above served as guidelines for the authors in the process of choosing relevant projects for the case study. The last requirement is relevant to ensure that a selected project has sufficient time in their schedule to fully participate in the study. However, it was only when all other requirements was fulfilled that the time was considered, and in the case of this interview series, no project were disregarded due to time constraints. Moreover, it should be noted that projects experiencing a lack of time could potentially, if not likely, pose different aspects and problems than what is demonstrated in this study.

The process of selecting relevant projects was carried out in several steps. Firstly, the authors reviewed Scania IT's project portfolio and extracted projects that adhered to the above stated criteria. Secondly, the authors consulted the group managers at the Project Management Office to gain a deeper insight into the chosen projects and based on their input the list of potential projects were further narrowed down. Finally, each project manager was contacted to decide whether the particular project possessed sufficient resources for participating in the study. This process resulted in four projects, which participated in the empirical study, outlined in table 2.1.
As described by table 2.2, that has been aggregated from the information in table 2.1, the four selected projects provides a wide range for the interview series. Firstly, the purpose of the projects varies from being a new product development project to a replacement of existing systems project. Secondly, which project parameter being prioritized within the projects differs significantly, meaning that, e.g., one project focuses on functionality while another project operates under a tight time-to-market. Thirdly, the organization within the projects ranges from a single team to a multi-team setting, which implies a difference in the amount of project workers involved in the projects. Fourthly, the way of working within the projects varies from purely Scrum to a combination of Scrum and the traditional Waterfall method. Fifthly, the projects range from being a highly prioritized project at Scania to being an internal pilot project at Scania IT. Finally, the projects have either one customer or multiple customers. The wide range in settings for the projects implies that the case study answers for the majority of possible projects at an IT organization in general, and for the average IT project at the case company in particular. As a result, the empirical data and eventually the analysis and conclusions of the case study are likely to apply for other projects at the case company than those included in this particular case study.
As reported in section 4.3.2 Project Scatter Factor, ten projects at the Project Management Office were chosen to serve as foundation for the Project Scatter Factor analysis. The project managers where contacted through E-mail and were asked to provide information for the project, e.g., estimated duration of the project and number of involved resources, that later was used to calculate the Project Scatter Factors. Due to a restricted time frame, the authors set a response deadline and only the projects that provided sufficient information within set time frame were used in the analysis.

The projects participating in the Project Scatter Factor analysis were not subject to deeper investigation, i.e., the information gathered through E-mail was deemed adequate for the intended analysis and no further questions to the corresponding project members were asked. Additionally, the projects have no direct relation to the interview series described above, and therefore, should be viewed as an independent section of this report without impact on other subjects than the Project Scatter Factor.

2.1.4.2 Interviews

The pre-study phase, specifically the literature review, assisted in creating interview topics. Since the authors aimed to interview several members, with different responsibilities, from each project the interview topics were designed differently depending on the interviewee's role and responsibilities. The interviewees were chosen through contact with each project manager respectively, although the authors had the final say regarding the interviewees' relevance. Each interview lasted between one and two hours and all interviewees were informed of the importance of their participation and of the main research goal, although, details were spared due to the risk of biasing their opinions on the research subject, which is argued for by Melo et al. (2013). Additionally, the interviewee was informed that all gathered data would be made anonymous, i.e., it would not be possible to identify who said what in the final report. The authors also asked each interviewee permission to record the interview to increase the accuracy of the empirical data.

Each participating project was studied by interviewing three to five people from the project, resulting in the list of interviewees in table 2.3. The alternatives, e.g.; fewer projects and more people; or, more projects and less people; lack balance considering both variety and level of detail. With this in mind, the chosen amount of projects and interviewees suited the authors’ intentions.
Table 2.3: List of interviewees from the four selected projects

<table>
<thead>
<tr>
<th>Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Coach A</td>
</tr>
<tr>
<td>Agile Coach B</td>
</tr>
<tr>
<td>Developer A</td>
</tr>
<tr>
<td>Developer B</td>
</tr>
<tr>
<td>Group Manager A</td>
</tr>
<tr>
<td>Group Manager B</td>
</tr>
<tr>
<td>Maintenance Manager A</td>
</tr>
<tr>
<td>Product Owner A</td>
</tr>
<tr>
<td>Project Manager A</td>
</tr>
<tr>
<td>Project Manager B</td>
</tr>
<tr>
<td>Project Manager C</td>
</tr>
<tr>
<td>Scrum Master A</td>
</tr>
<tr>
<td>Scrum Master B</td>
</tr>
<tr>
<td>Scrum Master C</td>
</tr>
<tr>
<td>Steering Group Representative A</td>
</tr>
<tr>
<td>Steering Group Representative B</td>
</tr>
</tbody>
</table>

2.1.4.3 Processing the empirical data

The method for processing the empirical data is shown in figure 2.2. In the first step each interview recording was transcribed verbatim and also summarized including the most relevant ideas covered in the interview. The summary and the transcription were sent to the interviewee for confirmation in the second step. Due to the extensive length of the transcriptions, interviewees were told to focus on the summary and confirm that the authors interpretation of the interview discussions were accurate.

![Figure 2.2](image)

After the interviewee had confirmed the validity of the material both transcript and summary were anonymized by the authors. This was done early in the process to ensure that the authors potentially biased opinion about an interviewee did not intervene with the perception and analysis of the data. The anonymization was done at the best of our ability; one author wrote down the interviewee name and role, which was covered up, before the second author wrote a randomly selected letter. However, since both authors participated in all interviews, it is still likely that certain quotes could be recognized.

In the forth step, the authors turned to the research questions and evaluated each interview-statement in the anonymized transcripts based on its relevance for that particular question. Thus, four buckets of data, one for each research question was formed. Each bucket was later internally discussed in order to group statements that related to the same topic. This resulted in several topics being identified under each research question respectively. In the fifth, and final, step of
the empirical data processing the identified topics of each research question was compared to topics of other research questions, which led to the conclusion that several topics co-existed in more than one research question. Those topics, became the base for the disposition of chapter 4. 

Empirical data. Figure 2.3 below exemplifies the research question categorizing and chapter disposition steps of figure 2.2.

![Figure 2.3: Categorization of the interview data according to the research questions with identified topics and their relation to topics of other research questions.](image)

### 2.1.4.4 Alternative approaches

The authors have discussed alternative approaches, e.g., surveys and workshops. Firstly, a survey could serve as a complement to the interview sessions resulting in a wider data range, although less detailed. A survey could be used to identify likely problem areas in projects that can later be confirmed and examined in depth through interviews. However, it is important to note that a survey should not be thought of as a substitute to the interview series, as the richness of the data will be significantly lower. i.e., although a survey can identify statistical relations between various studied aspects, it does not necessarily explain underlying reasons for those relations, while an interview series makes it possible to ask follow-up questions. With this in mind, a survey was not part of the case study.

Secondly, a workshop could be used to brainstorm potential areas of improvement regarding Agile practices. An obvious advantage would be that a workshop utilizes the aggregated knowledge of employees at Scania IT who are regularly working in or with Agile Software Development projects. However, the purpose of the study should rather be thought of as an attempt to identify improvements regarding Agile practices at the case company related to theory and then evaluate their respective performance in practice. Thus, a workshop was not part of the case study as this was considered to be to highly influenced by the employees of the case company.

### 2.1.5 Analysis and conclusions

As argued by Backman (2008), the analysis of qualitative studies is often tightly related to processing of the empirical data. i.e., the analysis and conclusions chapter of this report has been formed iteratively with the chapter on empirical findings.
When analyzing empirical data from a qualitative field study Backman (2008) further argues that it is of uttermost important to related findings tightly to the theoretical framework in order to increase the validity of the study. Failing to do so increases the risk of the analysis being either too narrow or not deep enough (Backman, 2008) Therefore, the authors have adjusted the theory chapter according to the empirical findings and subsequent analysis of the study.

Chapter 5. *Analysis* deeply investigates the literature review and the gathered empirical data presented in chapter 3. *Theory* and chapter 4. *Empirical findings* respectively. Findings in the case study is compared to what is generally argued on the literature, and in contrast, theoretical facts are applied to the case study to prove that the presented findings are indeed trustworthy and likely.

Finally, the analysis leads to chapter 6. *Conclusions* that summarizes the most important aspects covered in the analysis and tries to answer the research questions given in section 1.2.1. The conclusions are also discussed in relation to their respective generalizability, i.e., how well the findings of this case study would apply under different circumstances than experienced by this master thesis project.

Throughout the report it is important for the reader to remember the scope and limitations of the study, thoroughly described in section 1.2. The literature review, the empirical findings, the analysis and eventually the conclusions are all permeated by the context of the case study, and although some conclusions relate well to literature and are likely generalizable to any IT-organization, the aim of this report is to identify critical success factors in Agile software development projects in the context of an industrial manufacturing organization.
2.2 Credibility of the study

The following section intends to provide a description of the credibility of, and its impact on, the study. Furthermore, the underlying concepts of validity and reliability are presented.

Both Lekvall and Wahlbin (2009) and Björklund and Paulsson (2010) use the terms validity and reliability, with the aim of measuring the credibility of a study. Validity, on the one hand, is considered high if the study measures what it is intended to do, according to its scope and research questions. Reliability, on the other hand, refers to the result of the study's probability to remain the same if the study was to be repeated. High credibility is therefore achieved through high validity and high reliability simultaneously.

Björklund and Paulsson (2012) illustrate validity and reliability by comparing with dartboards. Validity is illustrated as in board A, where high validity is achieved when the dart hits the center of the board. Reliability is achieved when several darts hit the same area of the board. Thus, high credibility is achieved when several darts hit the center of the board.

![Figure 2.4: Illustration of credibility adapted from Björklund and Paulsson (2012)](image)

2.2.1 Validity

According to Lekvall and Wahlbin (2009), measuring validity of a study is considered impossible, since validating methods require knowledge of the study's result in advance. With respect to this, the aim of attempting to measure validity is to improve validity, rather than decide its value.

Björklund and Paulsson (2012) suggest triangulation as a way of improving validity, meaning to present several perspectives on a single topic. With respect to this, the authors intend to use multiple sources for each topic discussed in the study. Lekvall and Wahlbin (2009) suggest that several experts within the area should review the study, in order to increase validity. The authors intend to meet this suggestion by consulting supervisor's at the case company and Linköping's University as well as academic opponents to the report.

In order to increase validity, the authors used an iterative approach regarding the content's relevance towards scope and research questions. When information is continuously evaluated with respect to research questions its relevance is assured, while the probability of answering proposed research questions is increased. The iterative approach will be carried out through
continuous communication with the supervisors at the case company and at Linköping's University respectively, where newly discovered information is discussed along with the authors’ interpretation of this information's anticipated impact on the study.

2.2.2 Reliability
Reliability is mainly related to the information gathered at the case company. In particular, information collected through interviews are subject to special attention according to Lekvall and Wahlbin (2009). They present a few factors, among others, that affect the reliability:

- inconstant personal characteristics of the interviewee such as health, motivation, tiredness, and stress;
- factors depending on the situation, for example the interaction with the interviewer, distractions;
- the way questions are asked by different interviewers; and
- uncertainties in the measuring instrument itself causing several ways of interpreting questions.

According to Björklund and Paulsson (2012), a possible way for increasing the reliability of information from interviews is to ask control questions. This is done by asking several questions regarding similar topics to determine inconsistencies in received answers. Since multiple persons from the same project are interviewed and given similar questions, a comparison will be made in order to increase reliability. If interviewees respond differently to a similar question, there is a possibility that at least one of the listed factors can be seen as an explanation. The authors then have to, e.g., through asking control questions as mentioned by Björklund and Paulsson (2012), draw a conclusion and thereby increase reliability. Although, the different answers may originate from inconsistent knowledge because of the interviewee's position.
3. Theory

3.1 Agile methodologies in an organizational context

The following sections gives a brief introduction to Agile methodologies in an organizational context in order to help the reader understand reasoning in later chapters.

3.1.1 Traditional project methods

Following section aims to describe different methodologies in project management. The most important methodologies for the report, and more specifically the most relevant content in these methodologies, will be presented. Furthermore, which will be argued for in the coming sections, the methodologies contradicts the Agile approach in many ways, which enables a comparison between the different areas.

Since requirements are set early in the process, the plan-based methods are reduced in flexibility in subsequent phases of development. Therefore, an often occurring challenge is to overcome the lack of responsiveness to change when using a plan-based approach. (Barlow et al., 2011) Since the challenge mainly affects the development phase, developers are faced with the need of adapting to change (Austin and Devin, 2009). Provisions for responding to changing requirements are usually a part of traditional development methods, but are costly and time-consuming (Barlow, 2011).

Boehm and Turner (2003) discuss the risk of extensive delays due to rework of initial plan in a rapidly evolving, time-critical marketplace. The authors state that modifying a plan already decided upon, because of changes in technology, organizations, and market conditions would most likely be slow and expensive.

A traditional project management model is Cooper’s Stage-gate model (Cooper, 2001) (Karlström and Runeson, 2005). The basic idea is that each stage is separated by a decision gate, where the project’s current status and deliverables is compared to pre-determined demands, such as documentation. The structure of the gate model tend to be rather sequential with distinct phases, such as pre-study, feasibility study, execution and conclusion. Sponsors of the project then have the mandate to decide whether the project is allowed to proceed to the next phase. In addition, the gate model serves as support for the communication within the project. (Karlström and Runeson, 2005)

3.1.2 The Agile approach

The purpose of this section is to give the reader an introduction to the Agile Approach, its benefits and disadvantages, how it functions in an organizational context and how it relates to the Traditional project methods described above. The reader is expected to have a basic
understanding of Agile methodologies, although, the aspects necessary to follow the result and conclusions part of this report are briefly presented at the end of this section.

Many methodologies exist which can be classified as Agile, some explained later in this section, and according to Cockburn and Williams (2003) and Abrahamsson et al. (2002) they all share the same values and principles. These values and principles was first presented in the Manifesto for Agile Development (herein and commonly referred to as the Agile Manifesto) by 17 software practitioners. The Agile Manifesto as introduced by Beck et al. (2001) is characterized by four bullet points where the items on the left are considered to be of uttermost importance;

- **individuals and interactions** over processes and tools;
- **working software** over comprehensive documentation;
- **customer collaboration** over contract negotiation; and
- **responding to change** over following a plan.

The forming of the Agile Manifesto was not an instantaneous break-through but rather the aggregated result of years of development towards what today is known as Agile Methodologies (Williams and Cockburn, 2003). Although, the term Agile has been used in manufacturing practices for decades (Williams and Cockburn, 2003) and some Agile methodologies date back to the late 1980s (Fowler, 2005), Dingsöyr et al. (2012) argues that since the introduction of the Agile Manifesto research has been focused on explaining the agility. For example, Barlow et al. (2011) exemplifies the values in the Agile Manifesto, one at a time respectively, as;

- “Enhance communication within teams and barrier removal”;
- “Developers spend more time coding and testing than they do writing extensive documentation”;
- “Reduce formalities to start and finish faster, with a strong focus on the customer throughout the development process”; and
- “Give teams the freedom to make changes and adjust to project needs”.

Apart from the four Agile core values described and further explained above the Agile Manifesto as presented by Beck et al. (2001) also includes twelve Agile principles that are listed below to give the reader a deeper insight into the Agile mindset.

- 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 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 design emerge from self-organizing teams.
• At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

According to Dingsöyr et al. (2012) Conboy provides the, by far, most comprehensive definition of software development agility through systematically examining the various facets and definitions from related disciplines. Conboy (2009) relates agility to the two terms flexibility and Leaness. Firstly, the author argues that flexibility speaks of a development team’s potential to “create change, or proactively, or reactively, or inherently embrace change in a timely manner”. Moreover, Conboy states that this flexibility is achieved through the team’s internal components and relationships with its surrounding environment. Secondly, Conboy (2009) suggests that Leaness relates to the “contribution to perceived customer value through economy, quality and simplicity.”. Finally, the author claims that agility not only includes both those descriptions of flexibility and Leaness but also goes beyond, and hence, development team agility is defined as a continued readiness “to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity) through its collective components and relationships with its environment” (Conboy, 2009). In addition, Dingsöyr et al. (2012), themselves, also provides examples of several aspects that should be associated with agility, namely having minimal formal processes, promoting maneuverability, speed of response, nimbleness, quickness, dexterity, suppleness and alertness.

3.1.2.1 Agile methodologies
This report is not a crash course in Agile approach, quite the opposite, the reader is expected to have some basic understanding of the Agile mindset and also have heard about the most common Agile methodologies. However, to provide an overview of what parts of the Agile approach are more central to the case study, a brief overview is provided in this section.

According to Chow and Cao (2007) there are countless software development methodologies that can be classified as Agile according to the above values and principles. However, the authors also narrow the list down to the following methodologies that are more commonly referred to in the literature; Extreme Programming (XP), Scrum, Feature-Driven Development, Dynamic System Development Method, Adaptive software Development, Crystal and Lean Software Development. Although slightly different, many of the methods share the same practices (Cockburn and Williams 2003; Abrahamson et al., 2002)
Remembering the four values of the Agile Manifesto, it would come as no surprise that Cockburn and Highsmith (2001) find that Agile methodologies in general focus on creativity rather than defining complex rules and processes, which they further explain is achieved through having only a minimal set of rules and practices in place. Typically the methods all revolve around the concept of iterations that are divided into four parts; planning, execution, review, and retrospective (Derby and Larsen, 2006; Drury et al, 2012; Schwaber and Beedle, 2002).

Firstly, the iteration planning includes decisions on what shall be performed in the coming iteration (Drury et al., 2012). Typically, decisions are made based on set priorities and the decision made affects both the iteration delivery for this iteration, as well as, the long-term ability of the organization to deliver to its customers (Drury et al., 2012).

Secondly, the iteration execution is the part when the agreed upon work from the iteration planning is carried out (Drury et al., 2012). Decisions during this phase are typically aimed at how to best develop and test the functionality, and feature little uncertainty as the decision is made close to when those are to be implemented (Drury et al., 2012).

Thirdly, the iteration review is where the team focuses on the work that was completed during the past iteration and, most importantly, the outcome is compared to what was initially agreed upon during the iteration planning (Schwaber and Beedle, 2002). Moreover, the review should typically include a demo of a working product that has been tested (Schwaber and Beedle, 2002).

Finally, the iteration retrospective is a facilitated session when the team has the opportunity to reflect on their performance and collaboration over the past iteration and discuss, and preferably agree upon, improvements suggestions based on lessons-learned from past iterations (Derby and Larsen, 2006).

**Scrum**

Scrum is an iterative development method (Dybå and Dingsöyr, 2008), and a framework including roles, events, artifacts and rules that help the Scrum team organize their work (Schwaber and Sutherland, 2013). Thus, Scrum mainly focuses on project management aspects regarding software development (Dybå and Dingsöyr, 2008; Barlow et al, 2011).

In Scrum, developers are organized in self-organizing and cross-functional groups commonly referred to as Scrum teams (Schwaber and Sutherland, 2013). Besides developers, the team also includes a Scrum Master and sometimes a Product Owner (Schwaber and Sutherland, 2013). The role of the Scrum Master is partly that of a gatekeeper, i.e., removing impediments and keeping the team from outside disturbance (Schwaber and Sutherland, 2013; Dybå and Dingsöyr, 2008). Moreover, and perhaps more importantly, the Scrum Master acts as a coach, i.e., makes sure Scrum is followed as intended and understood outside the team (Schwaber and Sutherland, 2013). The Product Owner owns and prioritizes the Product backlog, an artifact containing all thinkable requirements that may or may not at some point be developed by the Scrum team (Schwaber and Sutherland, 2013).

Scrum acknowledges the four rudimentary parts of Agile methodologies that are described above, although they are referred to as Sprint planning, Sprint execution, Sprint review and
Sprint retrospective, respectively (Schwaber and Sutherland, 2013). Together they make up the Sprint, a time-boxed event that should be no longer than one month (Schwaber and Sutherland, 2013). In the Sprint, the Scrum team commits to a Sprint backlog, which is the part of the Product backlog with the current highest priority according to the customer; and the number of items is determined according to what the team finds feasible within the range of one Sprint (Schwaber and Sutherland, 2013). The goal of the Sprint is to deliver a usable Product Increment which will be evaluated during the Sprint review, usually through a demo and feedback from stakeholders (Schwaber and Sutherland, 2013).

During the Sprint, the Daily Scrum is a 15 minute time-boxed event where the Scrum team come together to discuss and synchronize their activities, answering the following three questions; what did I do yesterday that helped the team meet the Sprint goal?; what will I do today to help the team meet the Sprint goal?; and do I see any impediment that prevents me or the team from meeting the Sprint goal? (Schwaber and Sutherland, 2013). Just as the Daily Scrum allows the team to coordinate their work, the Sprint Retrospective is a session where the team evaluate their process and plan for improvement in the next Sprint (Schwaber and Sutherland, 2013).
3.2 Barriers when combining Agile with the organization

As will be proven in this section, much research has embraced the concept of Agile development living in a greater organizational context, in general, and what barriers could be encountered, in particular. This section will briefly examine how the organizational layout affects the potential for team Agility and will then lead on to a section on Critical success factors in Agile projects that do live in an sometimes complex organizational context.

This section starts with a quote that well captures the foundation for many of the obstacles that could arise when Agile meets the enterprise;

“people trump processes” but “politics trump people” (Cockburn and Highsmith, 2001)

What Cockburn and Highsmith (2001) mean by this is that while Agile development teams focus on individual competence to reach project success, lack of user and executive support can put a stopper in even the most skilled development team.

According to Dybå and Dingsöyr (2008) traditional software development methodologies are commonly heavyweight, while the Agile approach attempts to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like through being more lightweight. In addition, Cockburn and Highsmith (2001) states that Agile work thrives in exploratory, problem domains usually seen in complex and highly-changing projects and prefers people-centered, collaborative organizational culture. Thus, it is likely that Agile methodologies and the traditional organizational context will not function seamlessly together, but rather, that obstacles exist.

At the organizational level Moe et al. (2009) have identified three important barriers to agility, namely;

- **shared resources**, i.e., when project workers are forced to work on multiple projects simultaneously;
- **organizational control**, i.e., the management is more interested in reporting hours spent than actual project progress; and
- **specialist culture**, i.e., self-managed team needs generalists that can substitute for one-another rather than specialists.

Another common issue for implementation of Agile approaches in large companies especially within projects is discussed by Abrahamson et al. (2009). The authors argue that it is not unlikely that projects can have one or many Agile teams functioning quite successfully at the team level, however, at the same time those teams can find it impossible to follow Agile thinking outside of their boundaries. Similar to Melo et al. (2013), Lindvall et al. (2004) also expresses findings that Agile practices matches the needs for large organizations well, especially for small, collocated teams, possibly within project. Lindvall et al. (2004) then moves on to state that the challenge lies not with introducing Agility to the teams or the project, but with integrating the Agile teams in the organizational environment through better defined interfaces between the Agile team and its environment. Karlström and Runeson (2005) also supports this concept through claiming that
Agile software projects have been executed in traditional project management contexts without allowing the Agile development to affect the overall project management despite showing results on the development level. The authors further claim that traditional project management models are not originally designed for incorporating Agile methodologies, and therefore, fail to benefit from their advantages. Barlow et al. (2011) suggests that large projects and mature organizations are complex and have many interdependencies which could be seen as a reason why introducing the Agile concept could be complex.

3.2.1 Organizational Layout
Even though most organizations have similar needs, large organizations show tendencies of a need to see compelling evidence before adopting new methods. Due to their complexity and the need to combine new technologies and processes with existing ones, the need for persuasion looms greater in large organizations (Lindvall et al., 2004). Furthermore, Sutherland (2011) claims that the entire organization, e.g., operations, infrastructure and other functions need to support Agile practices in order to allow Agile software development projects.

Moe et al. (2009) report five separated actions for overcoming organizational barriers:

- **Organize cross-training:** This is costly, but the alternative cost is even higher. Increasing team flexibility through pair programming and job rotation address the vulnerability of the company’s lack of redundancy.

- **Collocate the team in the same room:** In contrast to distributed teams, collocated teams will improve the general communication within the team. Workers will discuss tasks they are currently working on and problems will be more frequently conferred. However, it is of great importance to balance the individual level and team-level autonomy by letting team members work uninterrupted when needed.

- **Appreciate generalists:** The company should be aware of the benefits of recruiting people with potential of creating redundant competence. Therefore, the company culture must appreciate both generalists and specialists in order to build redundancy into the organization.

- **Build trust and commitment:** The need for data collection should be motivated by the teams’ need for continuous learning, not by the company’s need for control. Management should avoid any control of negative influence on creativity and spontaneity in order to build trust in the entire organization. Trust and commitment are built in the organization through common objectives and cooperation among employees.

- **Assign people to one project at a time:** Employees should, as far as possible, be assigned to only one project at a time. It is a management responsibility to decide whether project or support requests should get resources from other projects, instead of team members making the decision of which projects to work on, in order to secure the potential of the self-managed team.
3.2.2 Combining Agile with a stage gate model

According to Cockburn and Highsmith (2001), adopting a process within an ecosystem can be done in one of two ways. On the one hand, the ecosystem can be altered to accommodate the process or, on the other hand, the process can be tweaked to fit the ecosystem. Furthermore, the authors claim that, in real life, both options coincide.

According to Lindvall et al. (2004) using a hybrid approach could allow the company to maintain existing quality systems while serving their customers better. Agile methodologies promises customer satisfaction, lower defect rates and faster development times, while, plan-driven approaches gives stability, predictability and high assurance (Boehm and Turner, 2003). This aligns with Nerur and Balijepally’s (2007) findings that traditional approaches to project management emphasizes detailed plans up front and heavy documentation throughout the project, while Agile development focuses on Lean processes and dynamic adaption. Although different, Boehm and Turner (2003) further argue that both approaches have their respective advantages and shortcomings. On the one hand, a lack of structure to support Agile development can cause chaos while, on the other hand, structure without agility risk leading to flaws from rigidity especially in projects that are prone to change (Batra et al., 2010). Therefore, a balance of the two has the ability to compensate for weaknesses in the respective models while incorporating their strengths (Boehm and Turner, 2003). Karlström and Runeson (2005) provide additional insight on the matter claiming that the Agile approach can provide tools for daily steering while the gate model supports inter-team and inter-project coordination on a higher organizational level. According to Pernstål et al. (2013) Agile methodologies have their greatest weaknesses in large-scale development, where the authors recommend a combination of Agile and traditional plan-driven methods.

As argued above, organizations could certainly benefit from combining traditional project management models with an Agile methodology, and as stated by Karlström and Runeson (2005) the schools of the Stage Gate-model and Agile methodologies are not contradictory. In fact, Gate-oriented project management models have already successfully been combined with iterative development models, one well-known example being the RUP (Royce, 1998). Although Wallin et al. (2002) agrees that the two types of models are combinable the authors also highlight potential problems that could arise in the process. Karlström and Runeson (2005) comment on Wallin et al.’s (2002) findings claiming that the inter-model interfaces and the description of how the processes need to be modified is lacking in detail. Wang et al. (2012) discuss the term Scrumban, that is, a software production model based on Scrum and Kanban. The model is said to be especially suited for maintenance projects or projects characterized by frequent and unexpected user stories changes or programming errors.

To successfully combine Agile methodologies and traditional project management methods is a complex task that requires considering multiple aspects of the project at hand. For example, Selic (2009) argue that managers should disregard certain Agile practices, e.g. reducing documentation, that does not align with the organizational size and culture. Moreover, Karlström and Runeson (2005) found that perception of control is one important aspect to remember. While engineers at the team-level experienced increasing control with the introduction of Agile
methodologies, managers perceived the opposite as they failed to adopt to the new situation (Karlström and Runeson, 2005). However, if management thoroughly understood the advantages of the improved micro planning, Agile methodologies could lead to better adherence with macro-plans (Karlström and Runeson, 2005). Selecting the appropriate mix of Agile and traditional methods also require Project Managers to identify differences and interdependencies among project tasks (Barlow et al., 2011). Barlow et al., (2010) also argues that managers should attempt to modularize the project processes into independent tasks, which aligns with Karlström and Runeson’s (2005) findings that successful integration of project management models and XP could be performed through clearly isolating the included teams and try to make them function like regular teams. A possible explanation to why this works is that Agile methodologies is mainly concerned with team level activities and practices for software development and not with the surrounding organizational context in which the development team lives (Hibbs et al., 2009).

In Product Development, one of the most crucial factors is early involvement of manufacturing engineers. Active participation from a manufacturing engineer reduces the risk of pushing through manufacturing prerequisites in late phases and, by that, possibly avoid costly changes and, ultimately, not jeopardize the launch of the product. (Sobek et al., 1999; Wheelwright and Clark, 1994) (Pernstål et al., 2013)

3.2.2.1 Decision making
In extension to what is mentioned in van Waardenburg and van Vliet (2013), Alleman (2002) and Lindstrom and Jeffries (2004) report that the project managers role as decision maker is greatly reduced when adopting Agile methodologies. Instead project workers, i.e., developers, may take decisions outside their normal skill areas, and product owners are involved in many decisions through their continuous and embedded role (Beck, 2000). However, Agile software development teams are continuously prevented from a linear decision process, due to factors, e.g., traditional project methodologies, which affect them negatively (Drury and McHugh, 2011). Generally, experience is a driving force in Agile software development project management decisions (Drury et al., 2011).

3.2.3 Project portfolio management and resource allocation
In any multi-project environment, efficient resource allocation between projects is key to the organization. Engwall and Jerbrant (2003) stress that multi-project environments are known to cause competitive relationships between project managers, who fight for management attention and resources in an environment that is highly political.

Cooper and Edgett (2003) present nine common excuses for a project portfolio containing too many projects. The excuses are listed below:

- Pressure to get anything to market: Managers express their unwillingness to kill projects, since they need to get something to market.
- Sunk cost reasoning: Difficulty to kill a project, due to major expenditures.
- We're almost there: The project team is begging for just a little more time.
- Executive "pet" projects: Projects that management believes in and have committed to.
Nothing better to work on: No other great ideas. In other words, no other projects is waiting to start.

No killing mechanism: No formal "Go/Kill" decision points.

No portfolio management: Decisions are made on individual projects. However, decisions on the entire portfolio is missing.

Can't say "no" to a key customer account: Unable to say no to any customer request, even poorly thought-out ones.

Too difficult to say "no": Management do not have the courage to turn down project proposals.

Research has found several tools addressing the resource allocation syndrome; Cooper and Edgett (2003) divide them in three categories outlined below; and Engwall and Jerbrant (2003) proves that the resource allocation syndrome is in fact a result of deeper problems in the multi-project environment.

3.2.3.1 Strategic solutions

Strategic solutions aim at making sure project resource allocation follows the organization’s business strategy, goals and competencies. A well-known method for determining the value of a business unit is the BCG-matrix, which with a minor modification can be applied to determine how well projects align with overall corporate goals. (Cooper and Edgett, 2003)

The authors argues that organizations should allocate resources as follows; Stars motivate more spending than the average business unit; Dogs usually meet a harvest and/or divest strategy; Cash Cows is given moderate resources normally aimed at updating the current product line rather than for New Product Development projects; Questions-marks are given resources based on magnitude of opportunity and previous history. (Cooper and Edgett, 2003)

Another strategic approach is to ring-fence resources. This method means to assign resources full time to product development, i.e., making them dedicated resources, rather than assign them part time to several other duties (Cooper and Edgett, 2003). The idea is similar to the project scatter factor described by Hendriks et al. (1999), which intend to explain how divided the project workers are between the different tasks they are assigned to. A lower project scatter factor is, normally, desired (Hendriks et al., 1999). The project scatter factor is further discussed under section 3.3.2.

3.2.3.2 Portfolio management solutions

Cooper and Edgett (2003) discuss that resource shortage problems may occur from other reasons than lack of resources, e.g., how resources are distributed between simultaneously ongoing projects. The authors present three different approaches that all assist in prioritize the current project portfolio and, hence, show where resources are most desirable. Firstly, a strategic bucket method categorizing projects into groups (or buckets) among whom they are to be prioritized; secondly, a formal portfolio management process implying continuous review of all ongoing projects; and lastly, accept only the projects that create value for the company.
3.2.3.3 Tactical solutions

In short, tactical solutions aim at creating a common understanding for the multi-project setting and sharing information within, in-between and around ongoing projects. Cooper and Edgett (2003) propose three possible approaches. Firstly, assist management in understanding how resources are deployed through displaying how resources are utilized and divided between projects. As a result, possible improvements are found to the current resource allocation. Secondly, project teams should specify milestones, but also the team’s resource demand throughout the project. That way, fluctuations in resource demand among projects are visualized, and necessary resources could be assigned to those who need it. Lastly, projects should aim at adopting a true team approach to handle possible resource gaps.

3.2.4 Knowledge about Agile methodologies

To increase commitment throughout the organization, the company need to take advantage of the people already committed to the topic. The common interest among these persons, and their will to deepen their knowledge, can possibly serve as foundation for Communities of Practice (CoPs). At Ericsson, for example, CoPs became a central mechanism in the large-scale Agile implementation that helped mitigate problems of the Agile transformation. (Paasivaara and Lassenius, 2013) Similarly, Takeuchi and Nonaka (1986) state that utilizing internal knowledge throughout the organization and creating a learning organization is a sign of a mature company.

The task of involving business management and reach agility at company level is a challenge (Rodriguez et al., 2014). Project managers’ understanding of the challenges in applying Agile methodologies in their entirety and the need of tailoring and choosing specific tenets to balance development needs and top management priorities is of great importance. In addition, top management needs to be aware of the importance of their roles in the extent to which Agile or adapted Agile methodologies are incorporated. (Cao et al., 2009)

One of the major concerns when adopting Agile software development methods is the possible loss of management control (Mahnic and Zabkar, 2012). van Waardenburg and van Vliet (2013) states that when a project manager denies the team his time and interest, a risk of being left out of the process emerges.

3.2.5 Working with requirements

At its core, agility refers to the ability to rapidly and flexibly create and respond to change (Henderson-Sellers and Serour, 2005; Hibbs et al., 2009. Referred to by Wang et al., 2012). If the Agile teams are faced with a non-negotiable scope, the requirements are set before the Agile process starts, which is a hinder for becoming prone to embrace change. Because of the team being unable to respond to change, the team builds features not corresponding to customers’ requirements. This results in rework and, thus, additional costs for the business. (van Waardenburg and van Vliet, 2013) In other words, failure to respond to change will result in creating waste. It is worth mentioning that the expression waste can be broad and context dependent. In software development, waste can be interpreted as: extra features, waiting, task switching, extra processes, partially done work, movement, defects and unused employee
creativity. (Poppendieck and Poppendieck, 2003, Hibbs et al., 2009. Referred to by Wang et al., 2012)

3.2.5.1 Time estimation
The sprint planning is one of the four formal events defined by The Scrum Guide™ (Schwaber and Sutherland, 2013) and seeks to decide what work shall be performed during the upcoming Sprint. Schwaber and Sutherland (2013) further states that the sprint planning should answer the following:

- what can be delivered in the increment resulting from the upcoming Sprint?
- How will the work needed to deliver the Increment be achieved?

The authors further argue that the first bullet-point requires the team to forecast the product backlog items that they will commit to delivering after the sprint. Although, not a prescribed practice by Scrum or the Agile manifesto, time estimation through story points is a popular way of forecasting the backlog items.

Story points is a high-level estimation of complexity, preferrably made before every sprint planning. The story points is later during each sprint planning used as basis for the low-level task-hour estimation, which in turn represents the actual effort needed to complete all tasks within a certain story. Thus, the story point estimation and task-hour estimation serve different purposes at different times and generally one should avoid talking hours to early in the estimation phase. (Scrum Alliance A, published 2012-08-23, accessed 2015-05-21)

3.2.5.2 Product owner and customer
At every iteration in Agile software development, business representatives are responsible of providing user stories to the project teams. Furthermore, business representatives are in charge of clarification of each user story. However, Agile software development projects experience that business stakeholders rarely provide user stories, which eventually results in unnecessary delays and loss of productivity. (van Waardenburg and van Vliet, 2013)

The case study conducted in van Waardenburg and van Vliet (2013) provides information which indicates that business stakeholders still are used to old habits. As the requirements are elicited and delivered to the development team, the business stakeholders perception is that their work is done and expect to receive a product answering to these requirements at the end of the project (van Waardenburg and van Vliet, 2013). Furthermore, the case study in van Waardenburg and van Vliet (2013) reports that the business still employs traditional thinking.

3.2.6 Agile teams in an organizational context
Sutherland et al. (2006) provides three different strategies for distributed Scrum teams; type A, B and C according to Project Management Body of Knowledge.
Figure 3.1: Strategies for distributed Scrum teams adapted from Sutherland et al. (2006)

Type A describes an isolated Scrum team approach applied onto teams which are not cross-functional. The work is partitioned across teams, creates teams with expertise within a specific area, and eventually violate the Lean development principles. Type B represents a model for relatively autonomous cross-functional teams, connected by a Scrum-of-Scrums where Scrum Masters meet regularly. This creates a co-equal environment and encourages communication, cooperation, and cross-fertilization. The type C model is characterized by fully distributed teams, where each team has members at multiple locations. At first, this model appears to create communication hinders, but the daily meetings help to break down cultural barriers and disparities in way of working. When applied to large enterprise implementations, the model can organize the project into a single whole. The nature of this approach provides location transparency, which creates performance characteristics similar to a small collocated team. (Sutherland et al., 2006)

According to Moe et al. (2009), work teams have several advantages, such as increased productivity, innovation, and employee satisfaction. However, the implementation of work teams does not always result in organizational success. Aspects of the organizational context such as reward systems, supervision, training, resources, and organizational structure are most likely affecting the team. Likewise, the communication with external key stakeholders can influence task performance (Moe et al., 2009). The authors found that, on team level, barriers to self-management are related to commitment, failure to learn, and individual leadership. In the presence of critical barriers at organizational level, implementing self-managed teams can be difficult, if not impossible. Misalignment between team structure and organizational structure can be counterproductive, and attempts to implement self-managed teams is a potential source of frustration for both developers and management. (Moe et al., 2009) Additionally, the authors state that self-managed teams’ performance is determined by the organizational context management provides as well, not only the teams’ competence in managing and executing its work. Self-managed teams is not guaranteed success and is highly dependent of appropriate situations and sufficient leadership and support (Moe et al., 2009).
The case study conducted in Melo et al. (2013) reports that the three primary factors affecting Agile team productivity are (1) team composition and allocation, (2) external dependencies, and (3) staff turnover. Moreover, inter-team coordination impacts Agile team productivity (Melo et al., 2013). Shared resources, prerequisite constraints, simultaneity constraints, and the relationship between tasks and subtasks are often used examples of dependencies in a project (Malone et al., 1993).

According to Tang et al., (2010), software development, and especially Agile software development, relies predominantly on teamwork. Salas et al. (2004) further define teamwork as “a set of interrelated thoughts, actions and feelings that combine to facilitate coordinated, adaptive performance and the completion of taskwork objectives”.

Team productivity is, according to Melo et al. (2013), affected negatively by team member turnover. However, the authors also found proof of that staff turnover can have a positive effect, thanks to the energy and the new knowledge new people brought to the group.

### 3.2.6.1 Organization of teams

Agile development is preferably carried out in smaller teams (Cockburn and Highsmith, 2001; Dybå and Dingsöyr, 2008).

Lyttinen and Rose (2006) report that self-organizing teams with local control are allowed to be open to innovative ideas. Team autonomy and team diversity is seen as important principles and practices in Agile development, since they foster software development agility (Larman, 2004). Autonomous teams thrive from decentralized decision making, and seem to be effective in sensing and responding to environmental changes (McGrath, 2011). A higher degree of autonomy allow teams to make decisions at a higher speed with lower cost, thanks to reduced bureaucratic organizational hierarchy which often is time-consuming and costly (Clark and Fujimoto, 1991). However, a higher degree of team autonomy infer greater risk taking and experimentation (Tushman and O’Reilly, 1996). (Lee and Xia, 2010)

Breaugh (1985) reports that team autonomy refers to the degree of discretion and independence allowed for the team in scheduling their work, determining the procedures and methods to be used, selecting and deploying resources, hiring and firing team members, assigning tasks to team members, and carrying out assigned tasks. According to Larman (2004) and Tata and Prasad (2004), autonomy brings authority of decision making to the people who faces the problems on a daily basis, thus, it increases the speed and effectiveness of problem solving. (Lee and Xia, 2010)

According to Melo et al. (2013) there are advantages and disadvantages in using collocation in software development. Lack of privacy, interruptions during work, lack of individual recognition, and disconnection from other teams were mentioned as shortcomings of collocation (Melo et al., 2013).

### 3.2.6.2 Team composition

Team diversity is, according to Williams and O’Reilly (1998), defined as the heterogeneity within the team in terms of individual attributes, such as age, gender, ethnic background,
education, functional background, tenure, and technical abilities (Lee and Xia, 2010). Campion et al. (1993) and Watson et al. (1993) state that learning and innovation increases with diverse competencies within a team. Furthermore, more alternative solutions for complex problems are more likely to arise in such teams (Lee and Xia, 2010). Although team diversity often is seen as a positive attribute, some authors propose that it may negatively affect the efficiency of team response process. Turner et al. (1987) suggest in Social Identity theory and self-categorization theory that, because of intergroup categorizations and different identities among workgroup members, team cohesion and integration suffer from diversity (Webber and Donahue, 2001). Additionally, diversity cause communication failures (Miller et al., 1998) and increases task-related conflicts (Pelled et al., 1999). (Lee and Xia, 2010)

Agile literature endorse that the internal variety of a software development team should meet the variety and complexity of the environment. Furthermore, the literature states that the internal variety is amplified by the diversity of skills and, additionally, is a key factor for the team to be able to respond to the changing environment. (Highsmith, 2004; Nerur and Balijepally, 2007) (Lee and Xia, 2010)

According to Bell (2007), the relation between team design and team performance is not farfetched, since the design affects the amount of knowledge demanded from each team member. Assigning people with appropriate skills and empowering them are key factors for success in Agile software development. Furthermore, Melo et al. (2013) report that the choice of team design is a factor impacting team productivity. Additionally, the authors state that desirable team design attributes are full-time allocation, diversity, team member skills, team size, and collocation as well as different profiles and knowledge levels was looked upon as positive for team productivity. (Melo et al., 2013)

3.2.6.3 Communication

According to Barlow et al. (2011), which is illustrated in figure 3.2, a large development project where the number of project members increases, causes an exponentially increasing amount of communication channels. Similarly, when the number of teams increases, the number of inter-team communication channels increases exponentially. This results in a complex collaboration situation.

![Figure 3.2: Communication channels within a team adapted from Barlow et al. (2011)]
A common success factor relating to software development projects and teams are communication, since it is an important factor in software development (Stelzer and Mellis, 1998. Referred to by Pikkarainen et al., 2008).

With intention to increase software development project’s agility, Agile methodologies aims to collocate the project team. As a consequence, communication and coordination between multiple teams are left unaddressed, which eventually results in an upturn regarding meetings and documentation. (Lindvall et al., 2004) In keeping with this, Barlow et al. (2011) state that large, complex projects naturally contain extensive interdependencies and, thus, require a large amount of coordination. Additionally, Pikkarainen et al. (2008) state that in larger development situations involving multiple external stakeholders communication might be hindered by the mismatch in adequate communication mechanisms. However, the authors report that Agile practices improve both informal and formal communication.

At Nokia, for example, the approach of solving the challenge with increasing documentation and number of meetings is to reduce cross-team communication through independent subsystems. However, this can be an even greater challenge in large organizations as the architecture evolves. With this in mind, the question on how cross-team communication can be minimized without losing perceived synergies within teams remains unanswered. (Lindvall et al., 2004)
3.3 Measures
This section will introduce the reader to the concept of Measures in general, and Software Metrics in particular.

Brown et al. (2013) found that meaningful measurement and instrumentation can result in a higher level of agility and improved predictability of outcomes. This aligns with Zhang and Zhang’s (2014) ideas that metrics can increase the controllability and manageability of software projects through providing professionalism and standardized and scientific methods. Furthermore, having useful metrics requires utilization of both qualitative and quantitative methods (Marchesi, 1998).

3.3.1 Software Metrics
Software metrics is, according to Zhang and Zhang (2014), a continuous quantitative process of software projects and products that includes data definition, collection, and analysis. The authors further claim that the purpose of software metrics is to understand the situation and state of the current project. The critical factors of software metrics is the data collection, i.e. quantification of the software project subject to being measured and controlled, and analysis of the measurement process (Zhang and Zhang, 2014). Fenton and Neil (2000) state that software metrics is a mutual term for describing the area concerning measurement in software engineering. With respect to this, the authors divide the subject area into two components: (1) The way of defining the actual measures and (2) the process of how to collect, manage and use the measures.

Zhang and Zhang (2014) divide the software metrics into three areas; project measurement, product measurement and process metrics, visualized in figure 3.3. Software project metrics are related to e.g., size, cost, risk and degree of customer satisfaction of the specific project. Software product metrics are primarily quality attributes measurement, e.g., reliability, usability and maintainability. Software process measurements often include, e.g., software development maturity, management and productivity. (Zhang and Zhang, 2014). Below are the categories of process and product metrics described further, however, project metrics is not covered in depth due to the scope of this study.

3.3.1.1 Process metrics
The software team response extensiveness measure is defined by Lee and Xia (2010) as the ratio between the amount of change requests received and the amount of actually incorporated change
requests. The higher response extensiveness, the higher software development agility. Software team response efficiency is, according to Lee and Xia (2010), defined as the minimal time, cost, and resources needed for a team to incorporate a particular change request. The authors summarize by stating that the relation between the measurements is of great interest when determining the agility of a software development team. Aligned with Lee and Xia (2010), Lindvall et al. (2004) report that agility can be reflected in the speed with which the development team implements change requests.

Mahnic and Zabkar (2012) observe several measures of progress, e.g., velocity, amount of work remaining, and schedule and cost performance indexes. Analysis of the velocity measure revealed two mistakes that are commonly committed that should be avoided in Scrum projects: estimation of velocity should be issued on previous sprint’s velocity and there should be no changes in the development team during the project. By measuring the work remaining, which is commonly done through burn-down charts, it is easier to monitor the impact on completion date of added or removed functionality which simplifies the communication between customer and development team. (Mahnic and Zabkar, 2012)

3.3.1.2 Product Metrics
In literature product metrics are usually further divided in three sub-categories that are briefly described below according to the description given by Assmann et al. (2013).

- Firstly, source code metrics describe the general characteristics of source code and can be applied to any programming language. E.g., lines of code, function points etc.
- Secondly, object-oriented metrics are strongly related to object-oriented programming, and hence, is not applicable to all programming languages. For example, inheritance coupling is commonly used, which measures the number of parent classes to which a given class is related.
- Finally, consists of six further object-oriented metrics established by Chidamber and Kemerer, referred to as the CK metric set, and in comparison to the object-oriented metrics have been validated by research many times. Two examples are weighted method count and depth of inheritance tree.

As was argued in the limitations of research-section this report will not dive further into the field of software metrics.

3.3.2 Project Scatter Factor
The Project scatter factor is a measurement for the number of staff needed to complete a one year task (Hendriks et al., 1999). The background of the measure is that allocation of specialists in, e.g., a multi-project environment can be difficult because of their specific knowledge being needed in several projects simultaneously (Hendriks et al., 1999). The larger the scatter factor; the more people are involved in the project; and the less of the project work will be accomplished by the experts (Hendriks et al., 1999). Moreover, devotion to the project decreases (Hendriks et al., 1999). Figure 3.3 show the effect of the scatter factor.
From the graph in figure 3.4 it is evident that a low project scatter factor would facilitate efficiency and optimal work within the single project, while, a high project scatter factor allows for flexible allocation of resources, especially when they are scarce. Similarly, the complexity of resource allocation also increases with the project scatter factor. According to Hammer and Champy (1993) a project team shall be a small multi-functional team, which Hendriks et al. (1999) interpret to say that a lower project scatter factor is always better and 1 is most desirable.

### 3.3.3 How to use measures effectively

According to Concas et al. (2012), the purpose of software metrics is to measure the effort needed to develop the software, or to measure its quality. According to Zhang and Zhang (2014), measures can assist in identifying problem areas, find out the root cause and eventually improve product quality and process efficiency. Furthermore, the authors report that the measures provide necessary information for project decision makers, in order for them to prioritize in the aspects of comprehensive cost, rate of progress, ability and quality, with the intention of adopting feasible actions.

Fenton and Neil (2000) report that two activities, i.e., resource prediction and quality prediction, are the reason for the evolvement of the area of software metrics. Additionally, the authors state that the majority of all metrics has been motivated by one of two activities: (1) The wish to estimate cost of development processes; (2) The wish to predict quality of software products. However, the history of software metrics shows that there is a need for company-specific data input, in order for greater accuracy and relevance (Fenton and Neil, 2000).

Staron (2012) reports that determine at what organizational level a measure should be located, which can be done through measure of the elapsed time between a decision point and the point where its results can be observed via measures. In other words, the length of the feedback loop is a crucial aspect, according to the authors.
3.3.3.1 Measures for decision making
According to Fenton and Neil (2000), software metrics have yet to address its most useful aspect, namely to provide managers with information that would support quantitative decision making. Thus, despite more than 30 years of study in the field, software metrics is scarcely used in everyday Software engineering (Fenton and Neil, 2000). In addition, Staron (2012) states that the metrics that most effectively trigger decision processes are those that warn about potential problems rather than monitoring status. However, Staron (2012) also acknowledges that status monitoring has its upsides, especially related to decision implementation.

Traditionally, metrics have usually been based on regression models that estimate cost and defects, but provide little support for managers that are interested in managing risk (Fenton and Neil, 2000). Another aspect is measuring project status and progress, i.e., provide managers with daily updated information about relevant areas of the project, e.g., requirement coverage or test progress (Staron, 2012). Thus, the field of software metrics should strive to build management decision support tools that combine various aspects of software engineering to help managers predict and assess outcome of software projects (Fenton and Neil, 2000).

3.3.3.2 Measure what counts
In the team, on the one hand, excessive measurement processes quickly risk creating a heavy workload without contributing much value (Staron, 2012). Brown et al. (2013) highlight that individual practitioners tend to favor improvement if the number of metrics are finite with a clear interpretation, and hence, an organization’s ability to find those accurate, timely measures is critical. Moreover, Mahnic and Zabkar (2012) state that an organization’s agility can be harmed if proposed measures does not add value or require too much heavy administrative work to collect the necessary data. Managers, on the other hand, risk ending up in a situation where they are drowning in data, while still not getting the feeling of being well-informed as a result of failing to understand and identify the vital few metrics that would actually be useful but rather measuring everything possible (Marr, 2012).

In contrast, Staron (2012) states that an effective utilization of measures actually require a quite small number of measures, i.e. 20 at the top management level. Staron (2012) further concludes that mature software development organizations should have a limited number of measures showing relevant trends and statistics communicated throughout the whole company. In accordance, much research supports the idea of preferring fewer measurements: Kaplan and Norton (1996), Hope and Fraser (2003) and Parmenter (2010) to mention a few. As for ensuring that selected measures add value, Staron (2012) suggest that development teams themselves are given freedom to choose relevant metrics as long as they can be linked to the overall corporate strategy as this flexibility could provide management with the possibility of observing problems where they originate.

3.3.3.3 Performance and Result indicators
This section discusses the topic of Performance Measures. As this is a field of relatively little research, most of the content is gathered from David Parmenter’s study on the matter.
According to Marr (2012, p XXV), Key Performance Indicators are a vital navigation instrument that managers use to understand whether their business is voyaging successfully or if its skidding off track. Marr (2012), on the one hand, does not distinguish between various types of Performance Measures but refers to them all as Key Performance Indicators, whether it is a financial measure of last quarter’s revenue or it is a measure showing the corporate carbon footprint. Parmenter (2010), on the other hand, while agreeing with the vitality of Key Performance Indicators in organizations, claims that the term is often used incorrectly, and that Key Performance Indicators are really one of four types of Performance Measures. The authors of this report have not been able to decide what side has more support in literature and have, henceforth, chosen to use Parmenter’s (2010) definition of Performance Measures in general, and Key Performance Indicators in particular, as it was deemed more appropriate to this study. Therefore, the controversy on the topic will not be further investigated in this report, however, the disambiguation is presented to clarify the possibility of different viewpoints on this topic.

According to Parmenter (2010) there are four types of Performance Measures, namely:

- Key result indicators (KRI);
- Result indicators (RI);
- Performance indicators (PI); and
- Key performance indicators (KPI).

Parmenter (2010) further argues that the term KPI is mistakenly used to denote all four types above. Using the onion in Figure 3.5 as an analogy, the KRIs are represented by the uttermost layer that tells of the current condition of the onion; i.e., the amount of sun and nutrients it has received while growing; and the core of the onion represents the KPIs (Parmenter, 2010). Between the KRIs and the KPIs lie the RIs and PIs; i.e., measures that share the same characteristics as KRIs and KPIs respectively but are of less importance to the business (Parmenter, 2010). However, a PI or RI may very well be of high importance to one department or team and can, thus, be considered a KPI or KRI on a lower organizational level (Parmenter, 2010).

![Figure 3.5: An analogy for the four types of Performance measures adapted from Parmenter (2010).](image-url)
Customer satisfaction, Net profit before tax, Profitability of customers, Employee satisfaction and Return on capital employed are all typical KRIIs (Parmenter, 2010). Those measures share one common characteristic; they are the result of many actions (Parmenter, 2010). Parmenter (2010) further argues that a KRI tells what you have done in the past, however, it does not tell you what you need to do to improve the result in the future. In contrast, KPIs can be defined to be “a set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization” (Parmenter, 2010, p.4). Parmenter (2010) also outlines a few key differences between KRIIs and KPIs that are summarized in table 3.1.

<table>
<thead>
<tr>
<th>KRI</th>
<th>KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can be both financial and nonfinancial</td>
<td>Nonfinancial measures</td>
</tr>
<tr>
<td>Measured mainly monthly or quarterly</td>
<td>Measured frequently (at least weekly)</td>
</tr>
<tr>
<td>Ideal for reporting progress to the board</td>
<td>Acted on by the CEO and senior management team</td>
</tr>
<tr>
<td>Of little or no use to staff or management</td>
<td>All staff understand the measure and what corrective action is required</td>
</tr>
<tr>
<td>The CEO is often responsible</td>
<td>Responsibility can be tied down to the individual or team</td>
</tr>
<tr>
<td>Designed to summarize activities within one Critical Success Factor</td>
<td>Significant impact on more than one of top Critical Success Factors.</td>
</tr>
<tr>
<td>A result of many activities</td>
<td>Has positive impact, i.e., it affects all other KPIs positively.</td>
</tr>
<tr>
<td>Can successfully be reported by means of a trend graph</td>
<td>Normally reported by way of an intranet screen indicating activity, person responsible and past history.</td>
</tr>
</tbody>
</table>

Table 3.1: Differences between KRI and KPI (Parmenter, 2010)

Within the study field of KPIs, many books cover the concept of Lead and Lag indicators (Parmenter, 2010). Lead (proactive performance measurement) and lag (outcome measurement) indicators (Dyreborg, 2008) are according to Parmenter (2010) clouding the usage of KPIs, as a KPI can be both lead and lag, while a KRI is always a lag. Thus, the concept of lead and lag will not be further discussed in this report.
4. Empirical Findings

4.1 Background to Scania IT

This first section gives a brief overview of the case company and its raison d'être. The information in this section is aggregated from informal interviews with employees at the company and internal documentation unless otherwise stated.

4.1.1 The parent company Scania CV AB

This section aims to give the reader an overview of Scania CV AB, the parent company of Scania IT. The information herein are, unless otherwise stated, fetched from interviews with personnel at Scania IT. It is presented here to give the reader a brief background to Scania IT.

Scania CV AB, herein referred to as Scania, is an international manufacturer of a wide range of products in the heavy vehicle industry; including distribution vehicles, long haulage trucks and special purpose vehicles like fire trucks; coaches and city busses; and engines.

The major parts of Scania are located in Södertälje in Sweden, however, with several larger sites, e.g. Zwolle in the Netherlands, and many smaller service affiliates around the globe. Being a traditional industrial manufacturing company Scania is generally permeated by Lean methodologies. Moreover, Scania strives to have a normal situation that is focusing on a standardized working method.

Although, Scania traditionally is a manufacturing company; IT is playing a crucial role and trucks are incorporating ever more software, not to mention the vast systems needed to run production, service centres and research and development of future products. Therefore, Scania has an increasing need for IT services which led to the creation of a IT department that later was demerged into an IT subsidiary, which are described in more detail below.

4.1.2 Overview of Scania IT

Scania IT is a subsidiary of Scania, with the purpose of providing its parent company with IT solutions, be it hardware and storage; bought or rented pre-built applications; or software development. The services are mostly for facilitating Scania's daily activities, while, software for the trucks and buses are handled by departments at the parent company. That is, Scania IT provides business supporting solutions, e.g., tele-conference applications. Nevertheless, many services are vital for business; among others Mona, the production controlling system, causes extreme costs on every system fault.

On the one hand, Scania IT has only one customer company, namely Scania itself. One the other hand, Scania is free to choose any IT services-provider it sees fit, which means Scania IT is experiencing competition, if however small. Furthermore, being the customer, Scania is directly or indirectly paying for any provided service and for any project, internal as well as external.
Regarding end-users of systems, it is not as clear. Although, most systems are ordered for internal Scania purposes, there are some applications where the end-users are actually the customers of the parent company. To conclude, the orderer and requirements specifier is always a department at Scania, while the end-users can be both employees at Scania, and Scania IT, as well as, customers or business partners of Scania.

4.1.2.1 Internal organization
When looking at the organizational chart for Scania, although, being a separate company Scania IT is represented just like any other department, e.g., Research and Development Depart and Finance Department at Scania. The figure 4.1 below shows an overview of the Scania IT organizational chart at the office level.

![Organizational chart for Scania IT at the office level](image)

Scania IT is notably affected by its parent company with regards to internal organizational structure. At the bottom level, resides the group with their respective group manager. A group is typically a collection of five to fifteen people with the same type of work. Thus, e.g., a group at the Project Management Office consists of only project managers, and hence, when working with their projects, the group members interact with members of other groups in a cross-functional manner. i.e., group members do not necessarily spend most of their time in their home group.

Together a number of groups form a section with a section manager, once again gathering people with similar areas of responsibilities, but in a broader range than at the group level. Similarly, at the next level, sections are grouped together forming offices again with an office manager and wider but similar areas of responsibilities. Figure 4.2 below exemplifies the structure.
4.1.2.2 **Intended way of working**

The IT@Scania pamphlet is an internal document describing the core values and principles of the organization. The section below is unless otherwise stated, aggregated from information therein.

Figure 4.3 shows the Scania IT house that visualizes all core values and principles that are an important part of the daily activities at Scania IT. At the very bottom resides the three core values; customer first, i.e., to always focus on the customer and target the customer when decisions are made; respect for the individual, i.e., to always acknowledge everybody and also allow anyone to suggest changes at any time; and elimination of waste, i.e., both as a mean to lower environmental impact, but also, avoiding to make products that the customer do not wish to buy. Those are actually the same as for the parent company and speaks of a mind-set within the company.
The two rows above, in Figure 4.3 represent the principles that are important at Scania IT, where the "Normal situation - focus on flow orientation" is the most important part. The rest of the house is not explained as a part of this report.

4.1.2.3 PPS - Practical Project Steering

Although, this report focuses on Agile software development projects, PPS is the overall project steering model used at Scania IT, why a brief introduction will help the reader understand the context that the Agile software development projects live in.

Practical Project Steering, henceforth referred to as PPS, is a project management model developed by Tieto providing a complete support for companies that carry out any or all of their activities in projects. Central in the model is the eight decision points, that can be compared to the gates in the stage-gate model and that define the different stages of a project's execution. (TietoEnator, 2002) Table 4.1 outlines the decision points in PPS.

The PPS model includes instructions, role descriptions, process descriptions and document templates, as well as, checklist for deliverables and documentations that should be completed at each decision point in order to move forward (Project methodology designer A, 2014).

<table>
<thead>
<tr>
<th>DP</th>
<th>Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP1</td>
<td>Decision to initiate project</td>
<td>Pre-study</td>
</tr>
<tr>
<td>DP2</td>
<td>Decision to continue, change or interrupt preparation work</td>
<td>Pre-study</td>
</tr>
<tr>
<td>DP4p</td>
<td>Decision to start part of the execution before DP3</td>
<td>Execution</td>
</tr>
<tr>
<td>DP3</td>
<td>Decision to approve the project plan</td>
<td>Pre-study</td>
</tr>
<tr>
<td>DP4</td>
<td>Decision to start execution</td>
<td>Execution</td>
</tr>
<tr>
<td>DP5</td>
<td>Decision to continue, change or interrupt the project</td>
<td>Execution</td>
</tr>
<tr>
<td>DP6</td>
<td>Decision to approve the result of a delivery</td>
<td>Execution</td>
</tr>
<tr>
<td>DP7</td>
<td>Decision to transfer responsibility for a delivery</td>
<td>Review</td>
</tr>
<tr>
<td>DP8</td>
<td>Decision to close the project</td>
<td>Review</td>
</tr>
</tbody>
</table>

Table 4.1: Decision points in PPS (TietoEnator, 2002)

Scania IT holds a PULS-meeting once a week, where all project managers and relevant line managers gather to discuss the status of all ongoing projects in the portfolio relative to the PPS model. The process is visualized on a white board and each project manager reports the status of the project as follows; green, i.e., no deviations; yellow, i.e., deviations that are manageable with the current resources and knowledge and/or will not delay the project; and red, i.e., deviations that will delay the projects and/or the project needs external help. The board will also visualize the current utilization of resources for each project, similarly green, yellow or red depending on the status of that resource. The reason for the PULS-meeting is to allow sharing of lessons-learned between different projects, and when a project reports a problem, it usually leads to discussion among project managers and line managers how to best solve the issue at hand. However, the discussion is most often about the distribution of resources.

The project PULSE

The project PULSE is held once a week at Scania IT, and its mandatory for all project managers to attend. The reason behind the PULSE is to visualize status and progress in on-going projects,
as well as, escalating deviations from the project plan. In short each project manager reports if
his or her project(s) is (are) green, yellow or red according to what is described in the table 4.2
below.

<table>
<thead>
<tr>
<th>Red</th>
<th>Defined targets will be met according to plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>Deviations with planned corrective actions exists</td>
</tr>
<tr>
<td>Green</td>
<td>Project targets will not be met and no corrective actions are known or available.</td>
</tr>
</tbody>
</table>

Table 4.2 project statuses at the PULSE-meeting

4.1.3 Project Success and the project parameters

The grounds that a project’s success is evaluated on are, among the interviewees, commonly
referred to as the project’s accuracy on meeting initial goals regarding scope, time and cost. As
stated by Agile Coach A (2015-03-11), a project is deemed successful if the product is delivered
in time with the agreed upon functionality. However, as Scrum Master C (2015-03-03)
expresses, project parameters are often reevaluated during the project, which makes the
comparison between initial goals and actual result rather misleading:

“If you compare a project’s actual result with initial plan and solely evaluate the success of the
project on this, the project is likely to be doomed unsuccessful”.

Furthermore, Scrum Master C (2015) states that changing the project’s steering parameters in
order to reach customer satisfaction, the project should naturally be evaluated on the degree of
customer satisfaction instead of initial steering parameters.

Compared to traditional stage gate models, Agile methodologies provide the possibility to
control the development of the project, in a more flexible way. As stated by Agile Coach A
(2015), Agile software development projects should be goal-oriented instead of time-oriented,
meaning that the product owner is able to steer the project to correspond with changing
requirement needs. Group Manager B (2015) expresses that management feels a loss of control
when introducing Agile projects, however, Agile Coach A (2015) says:

"Agile methodologies make it possible to control and steer a project, while traditional project
steering methodologies create an illusion of being in control."

In contrast, several interviewees have expressed that if the product owner is aware of any
changes made to the project steering parameters during the project and the customer is happy
with the project result the project is usually deemed successful even if initial goals are not met
(Project Manager C, 2015-03-24; Project Manager A, 2015-03-06; Project Manager B, 2015-03-
13; Agile Coach A, 2015).

4.1.4 Agile at Scania IT

According to Steering Group Representative B (2015) Scania IT manages to incorporate Agile
methodologies quite well today, however, each individual team may have completed more or
less of the journey. Project Manager B (2015) agrees as he states that it differs significantly from
project to project how much of the Agile approach that have been implemented, which could also
be because it is up to each team to decide their own way of working. Moreover, Project Manager A (2015) states that Scania IT has a clearly set goal to increase the use of Agile practices, which is also confirmed by Group Manager B (2015). Project Manager A (2015) also continues by saying that a change in mindset with most of the project managers is necessary to increase agility.

Furthermore, several interviewees claim that Scania IT would benefit from increasing its use of Agile practices. To exemplify in what ways three quotes are given below.

“I believe there are several advantages form increasing agility, but it is also necessary to have a corporate strategy for what we mean by ‘Agile’. That is, we need to decide what is the right level for, e.g., documentation. To perform a daily stand-up every morning has been practice a long time, but we have to decide what’s next. How should we work with requirements analysis? Etc.” (Steering Group Representative B, 2015)

“To make the whole setup more Agile would help us a lot. For example, the product ownership. It is one thing that we work as Agile as possible, but we have the customer too. Scania does not have the tradition of working according to Agile principles, why continuous customer education is important.” (Scrum Master B, 2015)

“From previous experience it has been a great improvement to work according to Scrum. However, that does not mean it is the solution to all problems, although the potential for success increases. I would also like to add that Scrum is sometimes mistakenly compared to waterfall-models. It has been quite a long time since we used waterfall-models and frankly it is not hard to prove that Scrum is better. One should rather compare Scrum to more recent methodologies, e.g, Rational Rose.” (Developer B, 2015)

A relevant question regarding Agile at an organization relates to whom decides whether a project or team should work according to Agile principles or not. According to Project Manager C (2015) Scania IT has no official guideline that a project must or must not work according to Agile or waterfall, but that it is rather a decision made by the customer at Scania. Project Manager A (2015) and Project Manager B (2015) both tell the same story that whether or not a project works according to Agile methodologies is a decision made by the customer. However, both also express that it is usually possible for the project manager to suggest a way of working, and that the customer rarely will disagree as long as the project delivers the promised results. In contrast, Product Owner A (2015) claim that as a customer you like the idea of an Agile way of working, but Scania IT usually have to take the first initiative to introduce Agile practices in a project as we at Scania rarely have any knowledge in, e.g., Scrum.

4.1.5 A new project arrives

At the customer side, i.e. the parent company Scania, IT needs are handled by IT demand organization within Scania’s organization. That is, research and development, purchasing, HR and other functions at Scania all have their separate IT demand organizations that aggregates the function’s total need for IT support and channels it to Scania IT’s Customer Office. Depending on the size of the offices it may sometimes have several IT demand organizations that function at
various depths in the organizational structure. (Group Manager B, 2015-04-29) The image below is meant as a sketch to help the reader understand the set up and does not represent a true organizational chart.

![Organizational Chart](image)

**Figure 4.4: IT Demand Organizations at Scania**

The way projects arrive at Scania IT was modified upon the release of Scania IT’s new internal way of working BAFDIT (Business and Flow Driven IT) a few years ago (Group Manager B, 2015). According to Group Manager B (2015), a new IT-project should by default arrive at Scania IT’s Customer Office, where all software development people at Scania IT work. At this resource allocation forum it is decided whether the customer demand should be started as an IT project, generally if it is more than 400 hours, or whether it should be passed off to development in the maintenance organization at the Solution Office. The forum also considers whether all necessary core resources for the project is available, if not a project can not be started. (Group Manager B, 2015). After this stage and when an Assignment Directive, that clarifies the project scope, goals and resources, a Project Manager is appointed and the project is given a go (Group Manager B, 2015).
4.2 Barriers when combining Agile with the organization

4.2.1 Organizational layout

According to Maintenance Manager A (2015-03-12), Scania is used to measuring hours instead of story points. In addition, Project Manager A (2015) states that Scania would benefit from not attempting to pre-determine project parameters, such as time and cost. Maintenance Manager A (2015) expresses that, instead of ordering a precise amount of hours, ordering functionality would be a better fit for both Scania and Scania IT. Scrum Master C (2015) expresses this as an effect of internal invoicing, and would rather prefer receiving an order for functionality for a certain cost and successively add functionality if necessary.

Agile Coach B (2015-03-24) states that project workers spend too much time documenting under the impression that, e.g., the maintenance organization demands this information. According to Agile Coach B (2015), documentation is necessary at Scania IT, but you have to make sure that the information is needed. Agile Coach B (2015) describes that the maintenance organization has expressed that they study the code, which implies that the documentation might be redundant. The extensive documentation leads to several handovers, i.e., from the business organization to the solution architect, from solution architect to the development team, instead of communicating and collaborating in person.

Maintenance Manager (2015) states that combining maintenance with projects is hindering the Agile way of working, since maintenance always have the highest priority. Similarly, (Scrum Master B, 2015-03-18) expresses that maintenance troubles the project work:

“I try to protect the development team from external factors. However, maintenance is always prioritized, which makes it difficult to plan ahead more than a day."

Scrum Master B (2015) expresses that the main reason of the problem is the fact that Scania IT, and each office within, is divided in functional legs. This increases the distance between stakeholders and, eventually, forces the Scrum Master to prioritize among the stakeholders’ requirements (Scrum Master B, 2015). Similarly, Group Manager B (2015) states that the organizational layout combined with lacking communication forces Scania IT to solve someone else’s problems. Project Manager (2015) experiences the same situation and reflects upon the possibility of redesigning the organizational layout:

“I would prefer a team oriented organization, containing larger teams where each team is in charge of their situation. The team should, e.g., own the rights of deployment, and be able to steer their own way of working without influence from external factors. However, there would still be an infrastructure office, but the office should focus on servicing the teams instead of the other way around.”

Group Manager B (2015) states that the organizational layout is of great influence on Scania IT’s way of working. Maintenance Manager A (2015), Scrum Master A (2015-03-25) and Project Manager C (2015) state that the organizational structure, in particular the Infrastructure Office,
hinders the project’s agility, since they own the right of deployment. Scrum Master A (2015) describes the deployment procedure:

“We do not own the right to deploy our solutions. Instead, we order a resource from the Infrastructure Office one week in advance every time we want to deploy something. Then we call them an request to reschedule the order for today. This is extremely bureaucratic and not in any way an acceptable situation.”

As a result, the deployments suffer from extensive lead times, according to Maintenance Manager A (2015). Similarly, Project Manager C (2015) states that it is time-consuming and causes a waste of resources. The fact that the projects are forced to order a resource in advance is an attempt of time optimization, which causes delays for the projects, according to Project Manager C (2015). Additionally, the situation prevents project teams from deploying in the end of each sprint, according to Project Manager B (2015). Maintenance Manager A (2015) states that, according to the infrastructure office, software development and software testing are two separated phases. However, this is not aligned with the project’s way of working, according to Maintenance Manager A (2015). However, the situation would have been bearable if Scania IT attempted to learn and improve, but no initiatives have been taken, according to Project Manager C (2015). In short, Maintenance Manager A (2015) expresses:

“We are extremely upset with those poor guys at the Infrastructure Office.”

Scrum Master A (2015) states that, even though the Infrastructure Office is hindering the Agile way of working, frequently ordering resources with short notice may convince the Infrastructure Office that the project needs to own the right of deployment. As a result, the team now is able to set up automated deploy environments which will eventually improve agility (Scrum Master A, 2015).

The organizational layout at Scania IT makes documentation important especially when information or tasks are handed over between offices Project Manager C (2015). However, Scania has a habit of documenting the wrong things simply because one office thinks it is needed by another office:

"The developers document the application because they say it is needed by the maintenance team. However, the maintenance team say the do not need it but rather look at the code."

Thus, the habit of documenting the wrong things, in this case, hinder the communication, although, it is a change under progress that hopefully will pay off in the near future Project Manager C (2015). Moreover, Project Manager C (2015) expresses that lack of communication is a major barrier to successfully incorporate Agile methodologies.

Group Manager B (2015) tells that Scania IT has a Customer Office that is responsible for all customer contact and channeling the customer's needs back to the appropriate office at Scania IT. However, Group Manager B (2015) also says that the Customer Office is a quite new installation, and hence, several departments at Scania still have not realized that they need to contact the Customer Office first, but goes straight to the development team as they are used to.
4.2.2 Combining Agile with a stage-gate model

4.2.2.1 Interpretation of the stage gate model
Steering Group Representative A (2015-04-09) expresses that every project manager should work according to the PPS model, whether you want to or not. However, Project Manager A (2015) states that the project has not followed the PPS model, more than him having the model in mind. Project Manager B (2015) states that the PPS model serves as support for the project managers. Similarly, Project Manager C (2015) describes that the PPS model is more of a suggestion of a way of working and that the current decision point is of questionable importance:

“The model contains a checklist of some relevant tasks along the way, but you are not forced to strictly follow it. It is of greater importance to communicate the actual decisions made and to keep stakeholders informed.”

Agile Coach A (2015) states that he does not know how the project relates to the PPS model, but he assumes that the project manager takes responsibility for that. Similarly, Developer A (2015) and Developer B (2015-03-03) express that the PPS model does not affect the developers in their way of working. Moreover, Maintenance Manager A (2015) is satisfied with being unaffected by the PPS model.

Agile Coach B (2015) states that Project 2 is not affected by the PPS model at the moment, since they have been between two decision points for more than a year. Project Manager B (2015) confirms:

“We are trying to follow the PPS model, but since we are stuck in a development phase the relation between the project and the PPS model is far-fetched.”

According to Project Manager A (2015), the same applies to Project 1. Moreover, Project Manager B (2015) describes that the project plan should be updated as the project evolves, but that has not been done due to other priorities and lack of time. As a result, both Maintenance Manager A (2015) and Scrum Master B (2015) state that the PPS model does not affect the day-to-day work.

According to Steering Group Representative A (2015), it is common to modify the PPS model to match each project’s specific needs. Similarly, Project Manager A (2015) describes that each project must have an assignment directive and a project definition, but beyond that you modify the PPS model to satisfy your certain needs. However, the attempt to adapt the model to Agile methodologies has not been a success, according to Steering Group Representative A (2015). Likewise, Developer B (2015) states that the PPS model does not run well with Agile methodologies:

“The PPS model demands that it is clearly specified what use-cases will be delivered at what point in time. This has been a problem for us, since we usually avoid planning ahead more than one sprint.”
However, Project Manager C (2015) expresses that testing and learning and eventually finding a way of working that suits your organization is one of the advantages with Agile methodologies. Furthermore, Project Manager C (2015) states:

“Project managers can assist each other, since the same questions should be answered by each project manager at the decision points. From that point of view the PPS model provides great support as a communication tool.”

According to Project Manager C (2015), it seems as the project steering decreases as the size of the project increases. Similarly, Project Manager B (2015) states that the PPS model is suited for smaller projects. On the contrary, Steering Group Representative A (2015) states that the PPS model is more evident in larger projects, since the different phases are more distinct.

Agile Coach B (2015) discusses that projects run in the context of large organizations stresses some kind of structure, which is in conflict with the basic idea of Agile methodologies. This structure is commonly similar to traditional project steering methods, according to Agile Coach B (2015). As a result, strictly following the PPS model with continuous attempts of working according to Agile methodologies could cause problems, according to Agile Coach B (2015).

4.2.2.2 Pre-study
Several of the interviewees have claimed projects have a hard time living up to early plans set forth during the pre-study phase of the project, for example Project Manager B (2015) tells:

"We are generally late compared to plan. But that is because you cannot know beforehand how complex a software is going to be, and sometimes the customer also increases the scope of the project."

Another interviewee, Group Manager A (2015-04-09), agrees that time estimates from the pre-study are too tight and relies on the assumption that nothing unexpected will happen. Further the Group Manager A (2015) states:

"It is not unreasonable that the pre-study is accurate, however, it is unlikely."

A problem with Agile software development projects at Scania IT is that the result of the pre-study is considered to be the truth and that every possible angle has been covered, and hence, it is possible to set fix deadlines and budget for the project (Group Manager A, 2015). The Group Manager continues:

"If the pre-study suggests 458 000 hours for the project to complete, then everybody shakes hand and agrees on that number. But nobody really knows. And then two years later we stand there and realize that it did not hold. It is perhaps a bit rough to call it a guess, but it is a long shot for sure."

Estimating time is a major issue within software development and if the customer then believes in the early estimates they will most likely be disappointed (Group Manager A, 2015). Whenever a project does not complete on time, the customer would generally comment on the delay, and then more money is given to complete the project (Group Manager A, 2015)
Another issue with pre-studies and agility is that they are normally a long-drawn process (Group Manager A, 2015). Scrum Master B (2015) comments:

"It is a real disadvantage to tell half a year in advance exactly what is supposed to be delivered in six months. It is very likely that something has indeed change and then the delivery is either off or will be late."

Moreover, since Scania seeks to continuously improve its business a long pre-study could mean that more or less of what has been decided in the process is no longer relevant due to changes at the customer-side (Group Manager A, 2015). Group Manager A (2015) tells:

"Sometimes when one realizes that the initial pre-study is off, one simply performs a second pre-study to solve the problem."

When a pre-study is performed at Scania IT, it is normally representatives from the customer at Scania, an analyst and perhaps a solution architect that are involved in the process (Group Manager A, 2015). Once the pre-study is done the result is handed over the the team of developers who then have to live by whatever the pre-study decided (Group Manager A, 2015). The business side rarely has good knowledge of IT and the architects have probably not developed software in a long time if ever (Group Manager A, 2015).

"If you start a project with those preconditions and then realize that it will not be possible as the pre-study said we will get criticized... When we realized the early estimates in our last project we threw them away because we as a team could not stand by those estimations." (Group Manager A, 2015)

That the developers are involved too late, if at all, in the pre-study process causes problems for Agile project work at Scania (Group Manager A, 2015). Project Manager A (2015) also states that it is important that all development-teams are represented. In one project, only the development-team of one application was represented which meant that the development team had a significantly easier journey then the teams of the other applications (Project Manager A, 2015). Involving the developers earlier could also mean that the early time estimates would not have to be redone by the development team later on (Group Manager A).

According to Group Manager A (2015), one of the reasons to work Agile is to avoid long pre-study phases that end up being irrelevant in the end, he says:

"Often it is a short pre-study can be performed quite rapidly to decide the main headlines of what the customer wishes to achieve. Any details are okay to deal with later. If you have established a business case for the customer, which should always be done, then you have enough material to get started."

In addition, Project Manager C (2015) also says that a short pre-study phase is necessary, and one have to manage the greatest risks, with the project. Thereafter, however, the project should start by building something, only then can you get an understanding of which risks will be encountered later on (Project Manager C, 2015).
4.2.2.3 Steering group and decision making

Steering Group Representative A (2015) and Project Manager B (2015) express that the PPS model is beneficial as a support when making decisions. Similarly, Scrum Master B (2015) states that it provides an explicit plan for expected deliverables over time. Aligned with this, the model serves as a framework to use when interacting with stakeholders, according to Project Manager A (2015).

To facilitate decision making in the steering group they have to be fed the right information. For example, Project Manager C (2015) describes that the most important information to communicate to the steering group is the amount of value added and the occurred cost since last meeting. However, Agile Coach A (2015) expresses that they are lacking ways of communicating status to the steering group or to management, except the communication of status on Scania IT’s weekly pulse meeting.

A steering group at Scania IT consists of stakeholders for that particular project (Steering Group Representative B, 2015). The chairman of the steering group is normally representative of the customer at Scania, and other representatives could typically be from Customer Office, Solutions Office, Infrastructure Office and also the project manager for the project (Project Manager B, 2015; Steering Group Representative, 2015). Sometimes the customer has several representative in case multiple markets are involved (Project Manager C, 2015).

The steering group is responsible for making decisions related to the project and the PPS-model (Steering Group Representative B, 2015). Mostly, the role of the representatives in the steering group are that of a Group Manager which means that their ability to make decisions without asking their respective bosses may be limited, according to Scrum Master C (2015) who says:

"Because the steering group consists of Group Managers their decision-making ability is limited and sometimes they have to escalate the issue to their respective managers to reach a verdict."

Moreover, Scrum Master C (2015), claims that it is important that the steering group allows the development team to be part of decisions that affect them. For example if a discussion is held about allocating more people to the team, the team has to be asked since it affects their work (Scrum Master C, 2015). Also Agile Coach A (2015) claim that a steering group that interfere too much with the work of the development team, i.e., through demanding a way of working or following a tight budget, will negatively affect agility within the team.

4.2.3 Project portfolio management and resource allocation

4.2.3.1 Prioritizing the project portfolio

Steering Group Representative A (2015) claims that the reason why this project was started at Scania IT is that it serves a great business case at the parent company Scania. Steering Group Representative A (2015) continues stating that Scania IT never turns down a project proposal from Scania:
“We never tell Scania we do not have the resources to carry out a project. If Scania covers the finances, then we find the resources. Sometimes I have to ask my Manager for more resources, but as long as the money problem is taken care of it is usually not a problem. However, I sometimes have to warn the customer that finding special skills may take a while, or that it will be some time before a new hired person performs well in a project. The customer sometimes has the illusion that more people will be more efficient in an IT-project, but this is not like changing the tyres on a truck.”

The idea is that Scania IT’s customer office should be able to tell Scania that a project is not mature enough to be started or that resources are too scarce at the moment, but the customer office are probably feeling a pressure from Scania that all projects should be started nevertheless (Group Manager B, 2015). Similarly, Group Manager B (2015) states that project portfolio management at Scania IT is insufficient:

“Of course we have a list of all ongoing projects and potential new projects to be started in the future along with the project manager’s name, but we lack the ability to prioritize among our ongoing projects.”

As described previously in this section all projects at Scania IT belongs to an office at Scania, and more specifically to a process, e.g. anything from selling a new truck to repair and maintenance contracts, owned by that office. According to BAFDIT each demand organization shall have a process forums gathering all stakeholders for all parts, sub-processes, of the process. At this forum the IT needs of all sub-processes are discussed and prioritized and the result is then handed over to Scania IT’s customer office by the demand organization. (Group Manager B, 2015) Thus, the projects that Scania IT receive are, by some means, prioritized, however, Group Manager B (2015) expresses that it is not enough:

“The problem of prioritizing is passed on to the Customer Office that now needs to have a dialog with demand organizations from different functional legs at Scania and get them to prioritize in-between themselves and their different projects. The problem is that the demand organizations are unwilling or unable to do so. Instead they claim that this is Scania IT’s problem not ours.”

Also Maintenance Manager A (2015) states that this is an issue as the demand organizations of different offices at Scania usually say that their projects cannot be compared with a project of another demand organization and that it should be prioritized anyway. Group Manager B (2015) says that it feels like whoever shouts the loudest gets the resources they need. As an example, Product Owner A (2015-03-10) tells that Scania sold the first products related to this IT-project based on the initial assumption that the IT-support will be ready at a particular date, which now puts pressure at Scania IT to deliver the IT-support system by this date or it will cost Scania money.

Furthermore, Group Manager B (2015) claims that it would be beneficial if all projects had a priority, especially when it comes to resource allocation. Sometimes when a new project arrives the Project Management Office has to relieve a project manager from another project in order to start up the new project. Since the customer office cannot themselves prioritize which projects
are important, this has to be done in dialogue with the demand organizations. Group Manager B (2015) says:

“Sometimes when we have to remove a project manager from a project we, at Scania IT, have to decide where at the customer it hurts the least. No matter what we do someone is going to get angry, and it is not unlikely that they shout at us saying that ‘this is your problem, solve it’.”

On this topic, Project Manager C (2015) tells that, since Scania IT has no ability to prioritize projects on its own, the flow of work through the company varies drastically from one week to the other, which is a major problem according to Lean principles. A suggestion would be to put a fixed budget for Scania IT, which means that it is not allowed to grow or diminish in size to handle Scania’s demand, and then Scania would have to prioritize what is most important, and that will get done (Project Manager C, 2015).

**4.2.3.2 Resource allocation**

As mentioned above the lack of a working project portfolio management at Scania IT leads to issues in resource allocation. Group Manager B (2015) says:

“The Customer Office sets a roadmap for IT, to show what is in the loop and what projects are expected to start in the near future. For me it gives a small hint of what is about to come. However, it is usually with short notice, and the messages from them are more like ‘this project will start in two weeks’.”

Moreover the Group Manager B (2015) claims that the ideal situation would be to have the most suitable project manager for every project, however, it is unreachable due to resource shortages. In some cases we do hire a consultant rather than appointing one of our own project managers, but that is when we definitely do not have the right knowledgebase amongst our own (Group Manager B, 2015). Group Manager B (2015) also expresses that resource allocation is an issue, and continues that particularly when someone asks for specialized skills or rare competencies is it hard to find suitable personnel for the task.

**Resources and Agile teams**

At Scania IT projects were traditionally given appointed resources through the project manager asking the organization for, e.g., 20 hours a week from a Java developer to complete the project. This request would then end up at the desk of a Group Manager for a Java team at the Solutions Office. The Group Manager would then respond with the name of the developer that was appointed to the project. (Group Manager A, 2015) According to Maintenance Manager A (2015) this is because the PPS model requires named resources at a certain decision point, which according to her is not good Agile practice:

“When I perform a sprint planning with my team, then we are planning what tasks we as a team are to perform in the upcoming sprint. I cannot tell if it is Anders or Jon who will carry out the task.”
As more of the organization has moved to an Agile way of working in the development teams Group Managers have tilted towards denying project managers who ask for a specific person to complete a task. Group Manager A (2015) at the Solutions Office tells:

“I am usually a bit more persistent and tell the project manager he cannot have any resources unless he takes a whole team, because that is our way or working. But I know of other Group Managers who eventually will given in, and then you still end up with the old set ups. There is no one single person who can make this decision. The project managers have their Office which is separate from ours. Thus, not even the most senior manager at the Solution Office can rule in this question.”

Maintenance Manager A (2015) also claims that some project managers who have been dealing with this contradiction between PPS and Agile teams for a while have put in practice to name random persons in the papers since they know that is only matters to the process and has nothing to do with who actually will carry out the task in the end. Steering Group Representative A (2015) also claims that the more project managers get turned down when they ask for specific resources, the less they try, why it gets easier and easier for the Group Managers.

**Allocating hours**

According to project manager Project Manager C (2015) many developers are constantly annoyed having to report their hours. Scania IT employees has to report all hours worked on any project since the cost is invoiced to Scania, and has to be addressed to the right Office at the parent company (Group Manager B, 2015). Project Manager A (2015) comments on this saying:

“This is not something typical for Scania. IT-support-business are generally set up this way with time reporting and internal billing. Of course it is debatable whether this helps creating an Agile environment or not.”

In contrast, Project Manager C (2015), claim that time reporting does not take a lot of time, however, he also says that:

“One is concerned with the wrong things if one thinks it is important to report hours. We have a budget and a time plan so that we can handle deviations.”

Moreover, Project Manager C (2015) continue by arguing that since we report time on all our projects we should use those numbers to learn something and improve, but since we do not do that it feels unimportant. Maintenance Manager A (2015) raises another issue with the focus on hours spent on a project, she says that:

“If the project has been allocated, e.g. an architect for 16 hours, but only 12h every other week it gets tricky for me when planning. If the planned hours are not enough because something changes then the situation becomes trickier. Thus, I talked to the project manager and instead we asked for half a resource without talking about hours. This made my world so much easier. Of course, the project manager will still follow up on how many hours we spend compared to the budget, but for me it makes all the difference.” (Maintenance Manager A, 2015)
Task Switching

Several of the interviewees in the case study have expressed that task switching is a major issue for agility at Scania IT. For example, one developer said the following:

“Although we work only with one application, we receive change requests from the customer that use our applications and also the project that intend to develop an entirely new system. Thus, we receive both incidents that have to be fixed and new features to implement and this causes task switching. I think it is quite a big problem really.” (Developer A, 2015)

Moreover, Group Manager B (2015), also claims that it is difficult for a team that has many stakeholders since each stakeholder may mean a different way of working form the others. Project Manager C (2015) also claims that the duration of projects at Scania IT are affected by task switching:

“If one were to carry out the project with one team of four to five full-time resources one would be able to complete the project in a few months. Instead, because the team has other tasks to attend to, the project is stretched in time.”

Furthermore, Maintenance Manager A (2015) tells that she constantly has to explain to the customer that increasing the budget to allocate more hours to the project is not necessarily a solution, it is more important that the allocated resources are full-time. Similarly, Scrum Master C (2015), says that he thinks it is extremely important that no team member has concurrent tasks in other projects or maintenance organizations:

“I have had to fight with tooth and nail to avoid context shifting in the team. That is one of the most important things when working Agile.”

To solve the issue Scania IT sometimes try to reallocate people from their own organization to a dedicated project team set to work only on that project for a limited time frame, however developer A (2015) claim this is not necessarily a solution:

“If we were to allocate some people to only work on the project we would get the advantages of lesser task switching and more a more dedicated project team. However, since they would be working on the same application where the maintenance organization is correcting incidents and developing smaller features which means the project may miss out on information about updates that affect them. Moreover, once the project is over it will be necessary to merge the knowledge of the project back into the maintenance organization so that we can support its functionality. When we are working on the same application it is actually better that we work together despite task switching.” (Group Manager A, 2015)

Finally, Scrum Master B (2015) raises an issue with task switching when it comes to planning. i.e., since the team has several tasks to carry out in a sprint it is more difficult to estimate how many of the project backlog items will be reasonable to complete. Project Manager C (2015) continues on this topic saying that Scania IT never allocates the full sprint for project work, but actually reserves a certain amount of time for incidents and support:
“We usually deduct about 20% for each full time team member that is reserved for ordinary work in his/her functional unit. i.e., a developer works 40 hours in a full week, out of those about 32 hours are allocated for project work, but out of those only about 18 to 20 hours are actually related to delivering features. The rest is integrations, communication, improvement work etc. Most project managers and managers that do not understand software development do not like hearing that they will only get value corresponding to about 20 hours per week from a full-time allocated developer.” (Project Manager C, 2015)

4.2.4 Knowledge about Agile methodologies

According to Maintenance Manager A (2015), Scania IT has a habit of educate itself in new methodologies. Instead, Maintenance Manager A (2015) would prefer assign the task to someone with experience, in order to speed up the process of organizational change and avoid unnecessary complications.

Project Manager C (2015) states that the fact that Agile methodologies are not implemented throughout the entire organization complicates the work for those who try working according to Agile methodologies. Project Manager B (2015) describes that, Scania is interested in determine a delivery date for certain functionality:

“If I give them a date, we would not be following the Agile way of working. This is testimony of the lack of understanding for Agile methodologies at Scania”.

Similarly, Agile Coach A (2015) expresses that even though Scania IT attempts to work according to Agile methodologies, it is also Scania IT’s responsibility to educate Scania in the way of working. In contrast, Scania IT would also need to educate themselves, particularly project managers and managers in Agile methodologies (Group Manager B, 2015; Project Manager C, 2015). Moreover, Scrum Master C (2015) states:

“The habit of working according to traditional project steering methods is hard to replace, and it is going to take time before everyone is convinced of the benefits of Agile methodologies. Currently, Scania sees Agile methodologies as a Scania IT way of working that does not affect Scania, since they are a truck manufacturer and not an IT company”.

Since a project often has several stakeholders and is depending on multiple teams throughout the organization where most likely one or more is not working according to Agile methodologies, the agility of the project is interfered. For example, the project team has to make reservations of resources located in other departments, causing extensive waiting periods and uncertainties when planning, according to Project Manager C (2015). Aligned with this, Steering Group Representative B (2015) expresses that Scania IT has to convince the customer of the benefits of Agile methodologies. By producing results aligned with the customer’s expectations, Scania IT can overcome the barrier of lacking commitment from Scania, according to Steering Group Representative B (2015).
Furthermore, Project Manager B (2015) believes Scania IT would benefit from a forum for Agile methodologies, where employees could share their knowledge and experience of common areas of problem within Agile methodologies at Scania IT.

4.2.5 Working with requirements

Agile Coach A (2015) describes that the requirements process starts with a business analyst communicating Scania’s requirements to Scania IT. Usually the requirements are handed over to a solution architect and a system analyst who define the requirements during the pre-study without involving the development team, according to Agile Coach A (2015). The requirements are then handed over to the development team, which breaks down the requirements into specific tasks. Agile Coach A (2015) states that the breakdown procedure might cause changes in the requirements, but at this stage the business analyst has been disconnected from the project and are not present to agree upon the changes. Steering Group Representative A (2015) describes that even though the solution architect is involved to some extent during the project, the projects would benefit from a higher degree of participation. Similarly, Scrum Master A (2015) expresses a wish of continuous involvement from the Scania representative in order to improve the requirements specification process. Moreover, Agile Coach A (2015) states that the process needs to be shortened, by reducing the amount of handovers. However, in order to shorten the process a change in organizational layout is needed, which is going to be time consuming, according to Agile Coach A (2015). Aligned with this, Scrum Master A (2015) states that involvement of Scania representatives needs to improve, although it is a massive change. Furthermore, Maintenance Manager A (2015) reflects on the importance of involving the solution architect and the system analyst in the development team, in order to improve agility.

Project Manager B (2015) describes that the organization of the project contains a dedicated requirements team, which is separated from the development team. Similar to Agile Coach A (2015), Maintenance Manager A (2015) and Scrum Master A’s (2015) reflections, Project Manager B (2015) states that this organizational structure increases the number of handovers and, as a result, limits the project’s potential of working according to Agile methodologies. Instead, Project Manager B (2015) wishes to incorporate the requirements team with the development team. Moreover, the current project structure combined with requirement changes from the product owner causes a lot of rework for the requirements team, according to Project Manager B (2015):

“A requirement that has been broken down might be pushed out of the current sprint, since a requirement that has not been broken down is added to the sprint.”

In other words, the requirements team is unable to know if their priorities are correct, according to Project Manager B (2015). Additionally, the requirements being broken down and estimated might be poorly specified or in other ways incomplete:

“It happens every day that the requirements are sent back, because of them being incomplete. This is a problem we have to manage.” (Project Manager B, 2015)
In the case when a project has multiple stakeholders, few of the stakeholders take responsibility for the requirements, according to Agile Coach A (2015). As a result, misinterpretations of requirements are detected late in the project. According to Product Owner A (2015), it is important to be present and be able to agree upon changes throughout the project.

### 4.2.5.1 Time estimation

Agile Coach A (2015) describes that time estimation is at first done by a solution architect and a system analyst without involvement from developers or testers. The estimates are large and unspecified, and seem to be rough time approximations of epics rather than valid time estimations of tasks. These estimates are delivered to the development team and are broken down into tasks and eventually each estimate’s content is specified in detail. According to Agile Coach A (2015), it is not unusual that the two estimates differ. Group Manager A (2015) states that it seems unnecessary that two different teams estimate separately, since it causes double work. Because of this, Agile Coach A (2015) would rather see that the development team is involved in the estimation phase from the beginning. According to Agile Coach A (2015), the fact that the two estimates differ is not communicated to the solution architect and the system analyst, meaning that they still believe their estimates were correct.

The break down of estimates done by the development team is carried out in a way that suits each team separately. Developer A (2015) describes that the team has started using planning poker, which allows every team member to participate and express their thoughts, instead of the expert deciding the estimates, which easily is the case otherwise and which potentially can cause problems delivering what was planned.

Agile Coach (2015) describes that the project manager has applied for a specific amount of hours to be able to complete the project. But since the difference in the estimates is not communicated, the amount of hours is somewhat misleading. Furthermore, it occurs that the customer wants to add tasks during the project, causing the development team producing more than agreed and makes the amount of hours even more misleading. Agile Coach A (2015) expresses that this is not a problem, as long as there is awareness of its potential consequences. Scrum Master B (2015) states the importance of reevaluating the estimates during the project and communicating the status to the product owner, in order to reach a common understanding.

### 4.2.5.2 Product owner

Group Manager A (2015) states that the product owner should be a Scania representative, since Scania IT is not capable of prioritizing Scania’s needs. Additionally, it is common that a project is ongoing for a longer period of time, meaning that the business evolves and, thus, causes requirement changes. According to Group Manager A (2015), these changes need to be communicated from Scania to Scania IT, in order for Scania IT to avoid developing a product based on initially specified requirements. Similarly, Scrum Master C (2015) and Steering Group Representative B (2015) state that the project needs to be driven by Scania. According to Group Manager A (2015) and Scrum Master C (2015), involving the product owner in the planning of each sprint can ensure that the project is driven by Scania.
Scrum Master B (2015) expresses the need of a dedicated product owner who is aware of the effort that Scania IT expects. Group Manager A (2015) describes the issue with the product owner lacking commitment, which results in the product owner being absent from, i.e., sprint planning:

“*It is a problem that the product owner does not participate as much as we would like. At the start of the project we agree upon what is expected, but during the project it is obvious he or she has not understood its significance*”.

Furthermore, Steering Group Representative B (2015-03-16) describes the risk of the product owner not spending the expected amount of hours in the project, which implies that Scania IT has to express its demands to be able to work according to Agile methodologies.

Similarly, Steering Group Representative B (2015) states that one of the expectations Scania IT has, is that Scania provides a product owner who has agreed upon Scania IT’s expectations and can answer for Scania’s requirements for the product. Moreover, Scrum Master B (2015) and Group Manager A (2015) state the importance of the product owner being present to be able to communicate changes and provide continuous feedback.

However, the fact that the product owner is not familiar with software development projects might cause some communication issues, according to Group Manager A (2015):

“If the product owner is not a Scania IT employee, it might be troublesome to explain, e.g., the importance of code refactoring, because of the difference in terminology”.

As a result, it is important for Scania IT to be pedagogical towards the product owner, according to Group Manager A (2015).

Steering Group Representative B (2015) describes the challenge in finding a product owner who is able to prioritize requirements, since an application usually has multiple stakeholders. The stakeholders are necessarily not organizationally related, which implies that the product owner needs to prioritize between different stakeholders’ requirements. According to Steering Group Representative B (2015), this is a common problem for Scania IT.

The product owner is responsible for the requirements changes made, according to Scrum Master A (2015). Similarly, Product Owner A (2015) describes the responsibilities as participating in the sprint planning and accepting the list of priorities, which aligns with Scrum Master A’s (2015) statement that the development team wishes to receive a prioritized backlog from the product owner. Agile Coach B (2015) describes that changes in the backlog during a sprint are acceptable, as long as the product owner is aware of the consequences. Similarly, Scrum Master A (2015) expresses that changes during a sprint is possible, although the product owner is expected to remove something from the sprint when something else is added. Additionally, Project Manager C (2015) describes that the development team usually is flexible, and as long as the team are given clear directions it is possible to reprioritize their tasks. However, Group Manager A (2015) states that the product owner is not always aware that adding requirements will cause delays unless something else is removed.
4.2.6 Agile teams in an organizational context

Agile Coach A (2015) describes the importance of sticking to the sprint planning. Team members are often distracted by questions from external co-workers, which eventually results in the team being unable to deliver what was planned. Agile Coach A (2015) has tried to overcome this by shortening each sprint from five weeks to two weeks, in order to raise awareness of the effect of accepting tasks that are not a part of the sprint and hopefully teach the team of the importance of sticking to the sprint planning.

Project Manager C (2015) expresses that the solution architect and requirements analysts should be integrated with the team, to reduce the number of handovers. Agile Coach A (2015) also discusses the importance of involving the development team in the estimation phase and eventually increase commitment. Another benefit of involving the team in the estimation phase is, according to Agile Coach B (2015), to get an earlier connection to reality and possibly identify problem areas that otherwise would have remained unidentified.

The teams in one of the studied projects are nowadays collocated, which, according to Project Manager B (2015) and Scrum Master A (2015), has increased cooperation and productivity, and both agree going back to distributed teams is not an option. Scrum Master C (2015) states that, even though costs might increase, the benefits of collocation are undeniable. However, Scrum Master A (2015) describes that while there are benefits of collocation, a possible option is to have an entirely distributed team, meaning that no members are located together:

“*If a part of the team are collocated there is a risk that the rest of the team misses out on discussions, since the collocated part does not share information through, e.g., E-mail*”.

Scrum Master C (2015) expresses that the internal communication has benefitted from the team being collocated, thanks to shortened communication channels. As a result, the purpose of the daily meetings has been able to focus on formal topics. Similarly, Developer A (2015) confirms the advantages of collocation:

“*Discussions are held when needed, and you get a feel for the project’s status during the daily meetings*”.

Scrum Master A (2015) tells that the project teams recently have been divided into teams with fewer members in each team, due to the fact that the project has grown in size and, thus, caused the teams to grow. Similarly, Maintenance Manager A (2015) expresses that her projects team recently went through a similar refactoring.

According to Project Manager B (2015) and Scrum Master A (2015) the size of the teams has increased compared to the size at the start of the project. Scrum Master A (2015) describes the negative impact of this and states that each team’s productivity has decreased due to the increasing number of members. Project Manager B (2015) expresses that management is aware that familiarization of new members decreases productivity, but states that productivity will increase in the long run. Scrum Master A (2015) describes that the lowered productivity has caused uncertainties when planning a sprint, since the velocity of each team is unknown. Group Manager A (2015) describes the difficulties with communicating the impact of new team
members to the business organization, since it is common to believe that an additional team member equals an immediate increase of productivity. However, this is not the case in software development, according to Group Manager A (2015). Similarly, Maintenance Manager A (2015) states that there is not a one-to-one ratio between adding a team member and increase in productivity. However, Project Manager B (2015) states that increasing team sizes often are inevitable due to extended project scope.

Project Manager B (2015) says that in Agile projects the role of the project managers is reduced, and that responsibilities include communicating the project's goals and ensuring that the team(s) has the necessary prerequisites to succeed. Moreover, Project Manager B (2015) claims that this is a precondition for autonomous, self-organizing teams.
4.3 Measures
As argued for in section 4.1.3, project success at Scania IT is generally defined solely by customer satisfaction. Related to this, Scrum Master C (2015) believes that the use of Measures within Scania IT should be closely linked to customer satisfaction, and in particular to ensure and forecast customer value as well as estimate cost and time for delivery. Agile Coach B (2015) presents similar ideas when he claims that all teams have a need to measure their quality of work; their complexity, to ensure they do not create technical debt; and their velocity, to facilitate planning of future sprints.

4.3.1 Software Metrics
A few interviewees have expressed that their particular team uses Software Metrics, however, they are not generally reported to anyone outside their team. Moreover, several of the interviewees, all working in different project teams, have expressed a lack of use of Software Metrics within the organization (Developer B, 2015; Developer A, 2015; Scrum Master A, 2015; Agile Coach A, 2015), for example one interviewee stated:

*We do not use Software Metrics, although, it does happen that we manually calculate some metrics related to lines of code or the number of store procedures used by the application.*

(Developer A, 2015).

4.3.1.1 Product Metrics
Project Manager C (2015), a former Java developer, claims that the only software metrics he know to ever be used at Scania IT is SONAR, which is a metrics tool measuring code complexity, warnings, test failures and many more numbers. Project Manager C (2015) said they used it to follow the trends as the application grew in size, and that generally most measures will grow worse as application size increases unless something is done to correct it. However, Project Manager C (2015) further claims that they hade no strict policy when corrective actions where needed, but rather the developer on the team with the lowest tolerance level eventually got tired of all the red flags and tended to it. In contrast Scrum Master B (2015), a new scrum master in the .Net department of Scania IT, also reconciles with Project Manager C (2015) statement about SONAR being used for Java development, but that he does not know of any similar tools being used for .Net development at Scania.

Although, Software Metrics is not a mandatory part of software development at Scania IT, several of the interviewees express thoughts on how to improve in the area in the future. For example, Agile Coach A (2015) suggests that Scania should have a bug policy for how many, non-crucial, bugs are allowed in the code before the team has to correct them all before implementing any new features. On this topic Scrum Master A (2015) speaks about experience from a previous employer that utilised a bug policy that had evidently lead to zero known bugs in the application during the last release. Scrum Master A (2015) further comments that the numbers would probably be a wake up call for his current project, and that they generally do not have the time to correct bugs if they wish to keep up with the project plan.
Another metric that is used in some development teams at Scania IT is code coverage, unit testing and regression tests. For example one developer stated that:

“We use unit tests and code coverage locally on our team... Lacking coverage is bad, however, just because the coverage is 100% does not mean the code quality is, since coverage does not tell about the quality of the tests. More important is that we test priority features with regression tests.” (Group Manager A, 2015)

Maintenance Manager A (2015), presents some other measures that are frequently used for applications once they are released, namely up-time and direct run. The latter refers to the number of bugs and change requests comes from customers when a new release of an application is made and is meant to signal the quality of the release. However, Maintenance Manager A (2015), claims that a major issue with the direct run measure is that, due to limitations in her application, not all markets are reached simultaneously by a new release, meaning that the measure is misleading.

4.3.1.2 Process Metrics

Agile Coach A (2015) claims that measuring progress in software development at Scania IT is complicated since it depends on the team’s level of maturity:

“In Scrum it is generally a good idea to measure team velocity. Normally you would measure on story points, but if the team is really mature you can even measure on number of stories since the team will probably be able to break down most stories to be of similar size.” (Agile Coach A, 2015)

In one project, Project Manager B (2015) tells that progress measurement at the team level works quite well. The scrums perform sprint planning and commits to a certain amount of story points corresponding to several backlog items, and at the sprint review the result of the sprint is evaluated. The Agile Coach B (2015) in the same project agrees with the progress measurement on the team level, but tells that the project lacks an overall progress measure for the system roadmap, i.e., they do not measure whether they are likely to meet the next big application release to a new market, something that is of uttermost importance to the customer. In one of the other studied project, Project Manager A (2015) uses a project wide progress measures:

“I measure how many activities each subsystem is set to perform and how many they have completed so far. However, this measure excludes activities related to integration of the subsystems and system tests.”

In this project, there is indeed a measurement for the overall project progress. However, as the project manager mentioned this excludes activities, and some very important activities according to the Agile Coach A (2015):

“When we now start the system integration and system tests we are probably going to find a lot of bugs, at least, that is my experience from similar projects.”

Just as in the first project where the SOS-leader claimed that each team measured their progress quite well, does the project manager of this project state that each team has their own progress...
measures. However, in this case the measures are almost impossible to compare due to different way of working in the different subsystem teams.

“Since all the subsystem-teams work differently, it is hard to compare the progress measures. Although, each team have their separate measures, e.g., burn-down charts etc. But since one team works Scrum-like and another more Waterfall-like it is like comparing apples and pears with pumpkins.”. (Project Manager A, 2015)

Furthermore, when measuring and comparing among several development teams Agile Coach A (2015) states that it is important to clearly define the measure to avoid misunderstandings. As an example, Agile Coach A (2015) tells that at Scania different teams have different levels of maturity regarding automated testing and delivery. There are some teams that deliver to the end users at the end of a Sprint, while other teams barely deliver to test. Thus, if one were to compare the lead time for change requests of those teams the measure may easily be misleading. (Agile Coach A, 2015)

Moreover, progress measurement in projects at Scania IT are made more difficult by the fact that support and change requests related to bugs on systems already in production are a priority. Thus, a project team can sometimes be forced to quit their current sprint to focus on something related to their functional unit. Although, buffer time for such interruptions are added during the spring planning it is hard to tell whether it is good or bad that a team completed none or all of the items on the sprint backlog during the last sprint. This complicates progress measurement. (Developer A, 2015; Agile Coach A, 2015)

Scania IT measures lead time for the applications, that is the time from when a new change request is received from a customer to that it is given the status closed, i.e., implementation is finished (Maintenance Manager A, 2015). For this measure the release procedure at Scania IT interferes with the validity of the measure, and Maintenance Manager A (2015) claims:

“In the past Sprint my team delivered 17 change requests however they will not be given the closed status until they are released to the customer which only happens quarterly... My team has an average internal lead time of 250 hours, but the actual lead time is much longer... Moreover, since the team sometimes gets change requests that have higher priority, i.e. support errands, any unfinished change requests are put on hold unless resources are increased.”

Although, progress measurement are complicated, many of the interviewees explained that their team is using burn down charts to keep track of their own progress during the sprint (Developer A, 2015; Developer B, 2015; Agile Coach A, 2015; Steering Group Representative, 2015; Scrum Master A, 2015; Project Manager C, 2015). One of the developers expressed the following about using burn down charts in the team:

“I have experience from using burn down charts in previous projects, which was successful. In this project we do not use it, which could be because we do not perform time estimation of tasks in the project. I like using burn down charts, and the graph is very informative.”

Agile Coach A (2015) states that in the project he is currently working on they are using burn down charts related to hours spent, although, he would rather see release-related burn downs using
story points. A project manager in another project stated that the reason to why hours are frequently measured in projects are because those are the base for the internal invoicing from Scania IT to Scania, and it is important that all time spent on a project can be financed by the project budget approved by the customer (Project Manager A, 2015).

### 4.3.2 Project Scatter Factor

In order to calculate the Project Scatter Factor some data was gathered from randomly selected projects at Scania IT. The gathered data is presented in table 4.4. For each project the estimated time (in hours) to complete the entire project, the project duration (in months) from start date to estimated end date, as well as, the number of resources, i.e., people, regardless of the magnitude of their participation was collected.

<table>
<thead>
<tr>
<th>Project</th>
<th>Estimated Time (Hours)</th>
<th>Duration (Months)</th>
<th>Number of resources involved (People)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>766</td>
<td>19,0</td>
<td>6</td>
</tr>
<tr>
<td>P2</td>
<td>1200</td>
<td>6,9</td>
<td>15</td>
</tr>
<tr>
<td>P3</td>
<td>18000</td>
<td>17,0</td>
<td>60</td>
</tr>
<tr>
<td>P4</td>
<td>3000</td>
<td>15,0</td>
<td>16</td>
</tr>
<tr>
<td>P5</td>
<td>4000</td>
<td>11,0</td>
<td>8</td>
</tr>
<tr>
<td>P6</td>
<td>25005</td>
<td>26,0</td>
<td>50</td>
</tr>
<tr>
<td>P7</td>
<td>3500</td>
<td>10,0</td>
<td>15</td>
</tr>
<tr>
<td>P8</td>
<td>2500</td>
<td>36,0</td>
<td>10</td>
</tr>
<tr>
<td>P9</td>
<td>13000</td>
<td>36,0</td>
<td>100</td>
</tr>
<tr>
<td>P10</td>
<td>8000</td>
<td>11,0</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 4.4: Data for ten randomly selected projects at Scania IT IDI.

The data presented in table 4.4 above are used for calculating the Project Scatter Factor for the ten selected projects. The results are presented in section 5.2.2 along with a discussion of what the results indicate. The actual calculation of the Project Scatter Factor is outlined in more detail in appendix 8.1.

### 4.3.3 How to use measures effectively

Scrum Master B (2015), claim that in his experience project stakeholders are generally only interested in cost estimation and how it changes during the project. First when a project is closed do the stakeholders start to question the result of the project and compare it to initial demands placed on project start-up (Scrum Master B, 2015).

Steering Group Representative A (2015) claim Software Metrics are not generally communicated to management or customers since they are thought to be too technical and not informative at their respective level. However, Group Manager A (2015), believe the measures could be used to communicate quality of the product to the product owner. On this topic, Scrum Master A (2015), believes that a good use of Software Metrics could be to clarify to the customer what potential
quality issues exist in the code and motivate magnitude of project risks. Scrum Master A (2015) continues saying:

“In this project focus has been on new features rather than quality and this will cost the customer money eventually. Through using Software Metrics the customer could possibly get a better understanding of the potential risk the priorities imply. Software metrics may help us to visualize certain code related issues, to motivate that we have to work with this to avoid future problems. This project’s main focus is to deliver functionality without respect to quality, which might be costly in the long run.”

In contrast, Developer B (2015), claims that the use of software metrics as a way to communicate status to the steering group is not beneficial. Instead states that burn-down charts may be a more appropriate way of communicating status. In addition, Project Manager B (2015) describes that the use of burn-down charts has made it easier to communicate the overall status of the project. However, Scrum Master A (2015) states that the burn-down chart is not commonly visualized internally, even though the project potentially would benefit from using it at a regular basis. Project Manager A (2015) reflects on the potential impact of visualizing software metrics internally.

“Visualizing status internally could sometimes lower the morale, which is why I have chosen to keep the visualization to myself”.

Similarly, Scrum Master B (2015) describes that visualizing status could be a stress factor internally, in case of being behind schedule. Therefore, communication of status should be done with caution, according to Scrum Master B (2015). In contrast, Agile Coach B (2015) claims that more visualisation of measures and information radiators would be beneficial, since today the product owner has no way of knowing where to look for information about the project’s current status. Project Manager B (2015), in the same project, admits much work remains with communicating status to the customer. However, he claims that the project is constantly coming up with new ways of communicating their status to the customer who then provide feedback whether it is useful or not, and hence, the team continuously improve their status communication. Project Manager B (2015) believes that since the team has only been working according to Agile methodologies for a few months, that they are not yet mature in their status communication.

Project Manager A (2015), tells that he uses another measure in his communication with project stakeholders, namely change request intensity, i.e., how many change requests are received from the customers and how many are completed by the teams. According to Project Manager A (2015), this measure speaks of the current work load within the project and can tell the stakeholders whether the project is expected to be on track or not.

Product Owner A (2015) states that from Scania’s perspective it is relevant to receive status during the project, in order to get a preview of the project’s result. However, Project Manager B (2015) states that they lack experience in what information the customer has interest in and how this information should be presented.
5. Analysis

5.1 Barriers when combining Agile with the organization

5.1.1 Organizational layout
Maintenance and support tasks are assigned to each office. When projects are added on top of this the project work will be interrupted, since the organizational layout at Scania IT implies prioritizing maintenance and support. This will be investigated further in section 5.2.3.2 Resource allocation.

According to Melo et al. (2013), one of the three primary factors affecting Agile team productivity is external dependencies, which can be compared to the Infrastructure Office’s influence on the project teams. The Infrastructure Office owns by default the right of deployment, which in fact makes the project teams highly dependent on the Infrastructure Office. As a result, the development team has no possibility of controlling the actual time of deployment, which hinders the agility (Maintenance Manager A, 2015). Moreover, this interferes with Breaugh’s (1985) team autonomy discussion, where a team is allowed to schedule their work and determine the procedures and methods to be used.

According to Maintenance Manager A (2015), the Infrastructure Office deploys to production once every quarter of a year. In the case of a team using sprints of, e.g., two weeks, they will not receive any feedback from users until the Infrastructure Office deploys, which in fact reduces the benefits of sprints and continuous integration. This aligns with Sutherland's (2011) findings that operations and infrastructure need to support Agile practices in order to allow Agile projects. Moreover, Project Manager B (2015) describes that the project studies the amount of support errands received as feedback. However, if there is a lead-time of three months the feedback is less useful (Project Manager B, 2015). Similarly, an advantage with short feedback loops is the possibility to correct errors quickly, but due to lead-time the risk of an increasing amount of technical debt is inevitable (Agile Coach B, 2015).

As presented in section 3.2.2.1, Beck (2000) describes that Agile methodologies allows project workers to take decisions outside their normal skill areas. However, at Scania IT the organizational layout creates a need of escalating decision making, which prevents development teams from affecting the Infrastructure Office and, eventually, manage their external dependencies on their own.

The authors of this report believes that giving each development team the right of deployment internally would benefit both the development teams and the Infrastructure Office. As mentioned by Maintenance Manager A (2015), the development team has the need of deploying several times each day, which results in a lot of rescheduling and wasted resources when the Infrastructure Office has to be involved. As stated by Tushman and O’Reilly (1996), a higher
degree of team autonomy infers greater risk, however, internal deployment will not jeopardize systems already in production, but provides better conditions for testing within the project. Furthermore, a low frequency of deploying to production leads to long feedback loops with the end-users which increases the importance of automated tests.

5.1.2 Combining Agile with a stage gate model

5.1.2.1 Interpretation of the stage gate model

As stated by Cockburn and Highsmith (2001), either the ecosystem should be adapted to the process or the other way around. Compared to the case study conducted at Scania IT, the PPS model is viewed at as the ecosystem and Agile methodologies as the process. Cockburn and Highsmith (2001) express that it is of great importance not attempting to merge the process with the ecosystem, which is aligned with Group Manager A’s (2015) statement that the PPS model should not, and in fact could not, be modified to serve as an Agile model. However, as described by Project Manager A (2015) and Group Manager A (2015), it is common to modify the PPS model aiming to adapt it to the Agile approach, which differs from Group Manager A’s (2015) earlier statement. Instead, a hybrid approach would be a better fit at Scania IT, which aligns with Lindvall et al. (2004). Similarly, Boehm and Turner (2003) state that Agile methodologies promises customer satisfaction and faster development, while plan-driven approaches provides high assurance and stability. Agile Coach A (2015) describes the need of structure embracing the Agile approach, especially in the context of a large organization, which aligns with Pernstål et al. (2013) declaring that the greatest weakness of Agile methodologies resides in large-scale development. Additionally, Boehm and Turner (2003) state that both approaches have their benefits and drawbacks. Moreover, Batra et al. (2010) state that, on the one hand, lack of structure to support Agile methodologies can cause chaos, while, on the other hand, structure without agility risk leading to flaws from rigidity.

The case study conducted at Scania IT fails to bring clarity to what extent the PPS model needs to be followed. As stated in section 4.2.2.1, there is ambiguity whether the PPS model is suited for larger or smaller project at Scania IT, which stresses the need of clarification of how to use the model. As stated by Group Manager A, it is mandatory to work according the PPS model, no matter which project. However, on the one hand, Agile Coach B (2015) describes that the extensive period of time between the decision points in a large project creates a perception of being disconnected from the PPS model. On the other hand, the use of the PPS model in small projects seems rather overly regulated, according to Project Manager C. In sum, the question of why a mandatory process is unofficially treated optional remains.

Agile Coach B (2015) express that the PPS model provides a checklist of decisions, for those who feel a need of assistance. However, the most important factor is to inform each stakeholder of the decisions made to reach a common understanding, instead of making decisions without communicating the information, according to Agile Coach B (2015). This aligns with the Agile Manifesto (Beck et al., 2001), which argues that people trump processes.
As stated by Maintenance Manager A (2015), Developer A (2015) and Developer B (2015), the PPS model does not affect the team’s day-to-day work. Similarly, Project Manager B (2015) and Project Manager C (2015), the PPS model is uppermost used by project managers. This aligns with Karlström and Runeson’s (2005) statement that a gate model supports inter-team and inter-project coordination on a higher organizational level, while the Agile approach serves as daily steering within a team. Additionally, Mahnic and Zabkar (2012) discuss management’s perception of losing control when adopting Agile methodologies. However, in combination with Karlström and Runeson’s (2005) statement above, the presence of the PPS model could ease the perception of losing control.

5.1.2.2 Pre-study
As stated in section 4.2.2.3, the current setup of the pre-study phase at Scania IT is questionable. Firstly, the outcome of the pre-study is thought of as the truth, which makes it hard to adjust to changes during an on-going project. Secondly, the pre-study is time-consuming and is of a high level of detail. Lastly, the development team is not involved early enough.

Barlow et al. (2011) state that a plan-based method implies setting the requirements early in the process, resulting in reduced flexibility. At Scania IT, the requirements set in the pre-study limits the responsiveness to change in subsequent phases of the project. Similarly, Beck et al. (2001) state in the Agile Manifesto that customer collaboration should be prioritized instead of contract negotiation, which Barlow et al. (2001) interpret as the need of allowing the development team to make changes and adjust to project needs. Additionally, Boehm and Turner (2003) express that rework of initial plan due to rapidly evolving markets is likely to be slow and expensive, which motivates shortening the duration of the pre-study phase, since early set requirements risk being outdated in subsequent phases. Moreover, the interpretation of the Agile Manifesto in Barlow et al. (2011) stresses the importance of reducing formalities in order to start and finish faster, with a strong focus on the customer. With respect to this, Scania IT would potentially benefit from starting the development phase earlier and avoid focusing on details in the pre-study, especially since customer satisfaction is of great importance.

Requirements changes in subsequent phases are mainly a challenge for the development team, according to Austin and Devin (2009). This, in combination with the fact that the development team is rarely involved in the pre-study phase, implies that the persons responsible for the requirements set in the pre-study are not faced with the challenge of adapting to the evolving markets. As a result, they are unaware of what problems the development team encounters during the project. Additionally, Sobek et al. (1999) and Clark and Wheelwright (1994) express that early involvement of manufacturing engineers in product development could assist in possibly avoiding costly changes. The authors of this report believe the connection to software development is not far-fetched and, hence, relates this to the case study as well.

However, it should be noted that a rigid pre-study phase is sometimes needed when utilizing a stage gate model, which is confirmed by Project Manager A (2015).
5.1.2.3 Steering group and decision-making
Section 4.2.2.3 describes how the steering group relates to project and particularly regarding decision-making. In summary, the steering group is responsible for making decisions related to the project and the PPS-model. Compared to Agile methodologies, that generally prescribe delegating decisions to the team in order to facilitating autonomous and self-organizing teams, the PPS-model prescribes that decisions should be escalated. Moreover, as the steering group generally consists of stakeholders that have the authority of a Group Manager in the organizational layout, even the representatives of the steering group may sometimes have to escalate decisions to their respective bosses, especially regarding cost and time.

Furthermore, that decision-making within the teams according to Agile methodologies are somewhat hindered by Scania IT’s use of the PPS-model is not a unique situation. The literature study, e.g., Drury and McHugh (2011) also suggest that traditional project methodologies restrict the more linear decision process of Agile methodologies. Drury et al. (2011) continues the reasoning by stating that experience generally is a major driving force for how to structure decision-making in set-ups that combine Agile methodologies and traditional stage gate models.

Therefore, the decision-making hierarchy of the PPS-model could be a hinder for software team agility. For example, Scrum Master C (2015), claims that it is important that the steering group allows the development team to be part of decisions that affect them. Similarly, Agile Coach A (2015) claim that a steering group that interfere too much with the work of the development team, i.e., through demanding a way of working or following a tight budget, will negatively affect agility within the team.

However, as explained in section 4.2.2.3 several interviewees claim that the PPS-model serves as a useful support when making decisions related to project, and also regarding what deliverables to expect from the project at what time. In relation to Drury et al.’s (2011) findings, that steering group experience is important to successfully combine agility and a traditional project methodology, several interviewees have also expressed that the project have to feed the right information to the steering group to allow them to make the right decisions.

5.1.3 Project portfolio management and resource allocation

5.1.3.1 Prioritizing the project portfolio
As stated in section 4.2.3.1, Scania IT lacks the ability of prioritizing its project portfolio. As listed in section 3.2.3, Cooper and Edgett (2003) have found nine reasons for organizations having too many ongoing projects, where the reasons applying to Scania IT’s situation are listed below:

- Pressure to get anything to market. As stated by Product Owner A (2015), Scania has already sold the first products related to the IT-project, which for obvious reasons puts pressure on Scania IT.
- Sunk cost reasoning and We’re almost there. Additional cost is, at Scania IT, translated to additional hours, hence, the two reasons are combined. Group Manager A (2015)
describes that, in case of a project needing more time, it is not unusual to receive extra funding.

- No killing mechanism. The decision points in the PPS model is “Go/No go”. Instead, a “Go/Kill” approach would be more appropriate.
- No portfolio management. Group Manager B (2015) states that Scania IT lacks project portfolio management.
- Can’t say “no” to a key customer account and Too difficult to say “no”. Group Manager B (2015) describes that projects handed to Scania IT are expected to be completed, without respect to the current amount of ongoing projects or to Scania IT’s current overall situation. Additionally, Group Manager A (2015) expresses that a project proposal is never turned down, as long as Scania covers the finances.

Since the majority of the reasons listed by Cooper and Edgett (2003) is possible to relate to Scania IT’s situation, the authors of this report find it reasonable to believe that Scania IT is suffering from too many projects.

With respect to the abovementioned empirical findings, it is reasonable to state that Scania IT lacks project portfolio management. As stated by Group Manager B (2015), projects may be prioritized at Scania before being handed to Scania IT. However, the prioritization is not communicated and, hence, causes problems internally at Scania IT. In other words, without knowledge of how projects are prioritized by Scania it is unlikely to believe that Scania IT is capable of prioritizing the projects. As a result, Scania IT lacks data to communicate internally and, thus, the employees at Scania IT do not know which tasks to focus on. As stated by Group Manager B (2015), it seems as whoever shouts the loudest get the resources, which is a likely consequence of insufficient project portfolio management. This aligns with Engwall and Jerbrant (2003) who stress that multi-project environments are prone to cause competitive relationships among project managers.

### 5.1.3.2 Resource allocation

As established in section 4.2.3 and discussed above, Scania IT lacks a working project portfolio management which puts pressure on the ongoing projects in specifically regarding resource allocation, however, the experienced problem is not unique to Scania IT. For example, Engwall and Jerbrant (2003) claim that multi-project environments are known to cause competitive relationships between projects and project managers which describe the situation at Scania IT very well. According to Group Manager B (2015), sometimes the project that shouts the loudest gets the resources needed. In the following section the issues of resource allocation will be discussed further.

Firstly, an issue raised by among others Group Manager A (2015), is the inability to appoint the most suitable resource to a given task. i.e., within the Solutions Office if a specialized skill is required for the new project, Scania IT often have to scatter a resources between multiple projects since no already ongoing project can be deprioritized (Group Manager A, 2015). Moreover, Group Manager B (2015) claims that a similar problem exists at the Project Management Office, whenever a new project is started and no suitable project manager is
available. The customer, Scania, would usually complain loudly whenever a project manager is removed from an ongoing project, and hence, Scania IT have to either settle for a less suitable option or recruit a new one. However, as prescribed by the Agile approach an organization should favor generalists and avoid specialist competence, which means that the issue of finding the most suitable resource may lie deeper in the organization than with the problem of allocating resources.

Secondly, an issue observed by the authors of this report is that during the time we performed the case study at the company, Scania IT has hired several new project manager and significantly grown the size of the Project Management Office. Group Manager B (2015) confirms this through saying that the group has grown a lot lately due to the start-up of many new projects. The authors of this report would thus like to question whether, e.g., the Solutions Office where all the developers work and who will eventually carry out the actual work in many of the newly started projects have grown anywhere near as much. Group Manager B (2015) claims this is outside of her knowledge but she further claims that this is not something that is considered when hiring a new project manager. Thus, Scania IT is pushing the resource allocation problem around internally.

Thirdly, an issue of resource allocation is related to the PPS-model. As several interviewees have expressed the model requires that resources are named at a certain decision point which puts pressure on the organization’s wish too increase their agility. However, most of the interviewees have also expressed that this problem has diminished significantly over the past decade as more and more project managers get accustomed to the new way of allocating whole teams rather than individual resources. Thus, the report focuses no further on this issue. Although it should be noted that the practice that some project managers have put in place of using random names in the PPS-documentation as a solution to the problem suggest that the PPS-model definitely could benefit from being updated in this aspect, especially if the organization wishes to combine it with Agile methodologies.

Fourthly and lastly, an issue of resource allocation at Scania IT relates to the fact that the organization has to report all its hours to facilitate internal invoicing to the parent company. Since, this is a fundamental part of Scania IT’s being this is not further discussed as it is not likely to change. However, as Project Manager C (2015) describes the organization should strive to learn from the time reporting. For example, Project Manager C (2015) mentions that a full-time allocated developer will not add value corresponding to 40 hours to the project because of project overhead related to, e.g., communication and team work improvement. Thus, if Scania IT were to use the data gathered when time reporting the organization would be able to estimate how many hours of functionality will be delivered for each hour spent on a project which would greatly help estimating duration of future projects.

After having outlined the main problems that Scania IT faces regarding resource allocation some possible solutions to the issue will be discussed below. As described in section 3.2.3, Cooper and Edgett (2003) divide the resource allocation practices in three categories, namely; strategic solutions, portfolio management solutions and tactical solutions.
Firstly, strategic solutions are about ensuring that project resource allocation is aligned with the overall business strategy, goals and competencies. Cooper and Edgett (2003) claim that organizations should divide their projects into stars, dogs, cash cows and question marks, where stars are given majority of resources. Group Manager B (2015) claim that Scania IT used to have a notion named Gold Arrow a few years ago. A gold arrow was a project that would be prioritized above all others, which aligns with Cooper and Edgett’s (2003) idea about Stars. However, Group Manager B (2015) also tell that it has been a few years since it was last used.

Another strategic approach mentioned by Cooper and Edgett (2003) is ring-fencing which means allocating resources full-time to projects rather than dividing them among many other duties. Also Moe et al. (2009) reports that shared resources is a major barrier when incorporating Agile methodologies into an organization. At Scania IT this refers to the situation that many projects are run within the functional organization through using the existing development teams. This solution also relate to the problems that arise since project work is often deprioritized compared to support and maintenance within those teams. Project Manager C (2015) also claims that if he would get to choose he would create a team based organization where teams would not have to work on projects and support simultaneously. However, as Scania IT seems to have a clearly set priority that maintenance and support is always prioritized above project work, ring-fencing as Cooper and Edgett (2003) describes it is not a possible solution as the authors claim that ring-fenced resources should only work in projects rather than maintenance. Still the idea of assigning one people to one project at a time rather than several projects is supported by other authors, e.g., Moe et al. (2009) and Melo et al. (2013) who both argue that scattered resources severely affect software development agility. Scattered resources is also further discussed in section 5.3.2 Project Scatter Factor.

Secondly, portfolio management solutions argue that resource shortage is rather a problem of how the project portfolio is managed, and hence, is not an option for Scania IT due to the lack of ability to prioritize its project portfolio as described above. However, the approach of strategic buckets (Cooper and Edgett, 2003) could be useful for Scania IT since it would be easy to categorize all projects into buckets, e.g., the R&D bucket for all projects pertaining to Scania's Research and Development Department. The issue, though, arises since Cooper and Edgett (2003) further argue that project's within the same bucket could now be prioritized, which again is difficult as was established in section 4.2.3.

Thirdly and lastly, tactical solutions are about creating a common understanding for the multi-project setting (Cooper and Edgett, 2003). The first approach discussed is about visualizing how resources is allocated between ongoing projects. The idea is that this would enlighten management that the current resource allocation is an issue and, moreover, help identify possible improvements. For example, if two projects share the same two developers on half-time, the resources could be re-allocated between the projects so that each get one full-time developer without lowering each projects resources. When resource are visualized it would also be reasonable to visualize project priority so that at least all project managers and their respective group managers know which projects have the highest priority. For example, Group Manager B (2015) claim that it is an issue that the management board for the Project Management Office
does not discuss project priorities and that one manager may know that a project is prioritized without this being communicated to all the managers.

The second approach that the authors mention is somewhat similar to Scania IT’s PPS-model, and tells that resources should be specified for each part of the project. That way when a project reaches a toll-gate in the stage-gate model it should only be allowed to move forward once all required resource can be secured. However, this does not align with the Agile practices of not naming specific resources.

The third approach is basically about adopting a true team approach to handle resource gaps. This relates to Project Manager C’s (2015) wishes to create a team based organization where projects are prioritized within the teams. However, as have been argued for above, this is not likely to happen given Scania IT’s current priorities why this discussion is cut short.

**Task switching**

A result of the issues in resource allocation described above is task switching, i.e., that a single resource is forced to work on one or more project simultaneously, will also having to juggle work in his or her office, e.g., related to support or maintenance. As have been established in section 4.2.3.2 many of the interviewees, particularly developers, claim that task switching is a big problem. This aligns with the literature review where, among others, Moe et al. (2009) and Melo et al. (2013) argue that task switching significantly affect team agility.

However, as one of the developers explained it is not necessary that assigning a dedicated project team would solve the problem at Scania IT. Since, the project and the maintenance organization within the office would then work in the same application too many dependencies between the project team and the office will decrease productivity severely. Thus, it is likely that some task switching has to be accepted when Agile methodologies is combined with a large organization.

### 5.1.4 Knowledge about Agile methodologies

The Agile way of working is a newly adopted concept at Scania IT, which naturally implies that the knowledge in the area is yet to spread throughout the organization. As described by Maintenance Manager A (2015) Scania IT has a habit of educating itself, which according to Takeuchi and Nonaka (1986) is beneficial and a step towards becoming a mature organization. However, it is of great importance to assure that the information being spread is correct from a Scania IT perspective.

As described by Cao et al. (2009) in section 3.2.4, project managers’ understanding of the challenges in applying Agile methodologies and top management priorities is of great importance. Group Manager A (2015) state that Scania IT needs to educate Scania in the general idea behind Agile methodologies, in order to motivate their way of working. Accordingly, Scrum Master C express that skeptics need to be convinced of the benefits of Agile methodologies. With respect to Group Manager B’s (2015) statement that management lacks experience of Agile methodologies and to Project Manager C’s (2015) statement that there is a need of educating project managers at Scania IT in becoming Agile project managers, the authors of this report find
it reasonable to believe that the overall knowledge of Agile methodologies needs to increase at Scania IT.

As stated by Group Manager B (2015), management feel they are losing control, which is aligned with the statement by Mahnic and Zabkar (2012) and Karlström and Runeson (2005). This might be a consequence of lacking understanding of Agile methodologies among management, since Agile Coach A (2015) states that Agile methodologies allows steering a project, while traditional project steering methodologies create an illusion of being in control. By extension, the lack of understanding among management may prevent the incorporation of Agile methodologies at Scania IT. In sum, Karlström and Runeson (2005) state that if management thoroughly understood the advantages of the improved micro planning, Agile methodologies could lead to better adherence with macro-plans.

Group Manager B (2015) describes that an adjacent group at Scania IT has chosen to focus on sharing knowledge of Agile methodologies on their weekly meetings, in order to increase the overall knowledge among group members. The authors of this report find them likely to succeed, although it is of great importance to share the right information. In other words, on the one hand, this can be beneficial. On the other hand, if the information is misleading, it is a risk of unknown magnitude. Furthermore, it seems reasonable to state that those who have experience from successfully completed Agile software development projects should share the knowledge, instead of persons managing an ongoing project without proof of success.

Project Manager B (2015) expresses a need for a discipline forum, in particular a forum for Scrum Masters, which would serve as communication channel for share knowledge of Agile methodologies. This aligns with Paasivaara and Lassenius (2013), who discuss the benefits of Communities of Practice when aiming for increased commitment throughout the organization. Paasivaara and Lassenius (2013) exemplify this with the Communities of Practice at Ericsson, which helped mitigate problems of the Agile transformation.

### 5.1.5 Working with requirements

Van Waardenburg and van Vliet (2013) discuss that a non-negotiable scope hinders the development team from becoming prone to embrace change. Even though the project scopes at Scania IT commonly is negotiable, the fact that the product owner lacks commitment (Group Manager A, 2015) limits the possibility of negotiating requirements and, thus, the risk of incorrect time estimations and rework of features not corresponding to the product owner’s requirements increases.

As described in section 4.2.5, the process of requirement elicitation and time estimation contains several handovers and commonly leads to misinterpretations. A possible enhancement would be to include the solution architect in the project team, which is suggested by Maintenance Manager A (2015), Agile Coach A (2015) and Project Manager B (2015). However, as stated by Agile Coach A (2015), the current organizational layout at Scania IT does not allow this kind of setup and, thus, would require a redesign of the organizational layout. Additionally, the underlying reasons for the current organizational layout is
5.1.5.1 Time estimation

Working with requirements has a strong relation to time estimation, and hence, the following section intervenes tightly with the previous one.

As was established in section 4.2.5, the case study identified several issues related to time estimations at Scania IT, i.e., time estimates being performed on fuzzy requirements before they have been broken down appropriately, time estimates being re-performed once the development team is involved in the process, differences in initial estimates and estimates on refined requirements are not communicated to stakeholders nor to the people responsible for the initial estimates.

Firstly, the early estimates that are performed on the fuzzy requirements are meant to give the product owner an initial idea of the length of the project, which is fine, however, the issue arises since requirements and the related estimations are then handed over to the developers who have to adjust to someone else’s estimates. This negatively affects the team’s autonomy, which according to the performed literature review will further negatively affect team agility. Moreover, Agile Coach A (2015) states that it is not unusual that the initial estimates are off compared to the truth.

Secondly, the developers would usually break down the fuzzy requirements into more specified ones and re-estimate the initial time estimations. On the one hand, this causes double work, which is a waste of time, but on the other hand, the initial estimates are probably needed to give the customer an initial idea of the project length or it will not be accepted to start at all.

Thirdly, and more importantly, Scania IT currently fails to learn from the second round of estimates as they are not communicated back to the people who performed the first ones. According to Agile Coach A (2015) this is an issue since the architects and system analysts that perform the initial estimates are less likely to improve if they never are informed that their previous work was wrong. An improvement suggestion that have been made by several interviewees, among others Project Manager C (2015), is to include the architects and system analysts in the project teams and thus the communication of time estimates would be improved. Moreover, if the developers are allowed to be part of making the time estimates their commitment to the project will, according to theory, increase as they feel they can affect their own work situation. For example, Developer A (2015) claim that planning poker is a good practice as it involves the entire development team more and that everybody can have a say about time estimation for tasks and that not only the experts perform the time estimations as they are not the ones who will implement all features.

Finally, Scrum Master B (2015) claims that it is also important to continuously re-evaluate earlier estimates. i.e., the further the project proceeds the more is known about the software and potential issues in development, and hence, the team should be able to make estimates that are closer and closer to the truth.
5.1.5.2 Product Owner

As stated by Scrum Master B (2015), it is of great importance of having a dedicated product owner who is continuously involved in the project. However, Group Manager A (2015) states that it is common that the product owner lacks commitment. Accordingly, Van Waardenburg and van Vliet (2013) describe that business stakeholders common perception is that their work is done after delivering the initial requirements, and expect to receive a product answering to these requirements at the end of the project. This has several consequences, as the product owner misses out on status reports during the project and, conversely, the project team misses out on feedback. As a result, the continuous communication of requirements is hindered and the backlog might become outdated and, thus, the risk of misinterpretations increases.

According to Schwaber and Sutherland (2011), the product owner should be the owner of, and the one prioritizing, the product backlog. However, projects at Scania IT find it troublesome to assign this responsibility to one single person, due to the organizational layout in general, and the numerous stakeholders in different offices in particular. As stated by Steering Group Representative B (2015), one person is rarely able to answer for other office’s requirements, and thus lacks the possibility of prioritizing the entire product backlog.

As stated by Group Manager A (2015), the product owner should, and commonly are, represented by a Scania employee. However, the degree of knowledge in Agile methodologies is generally lower among Scania employees compared to Scania IT employees. As the product owner is involved in many decisions through their continuous and embedded role (Beck, 2000), there is a need for Scania IT to continuously educate Scania in the responsibilities of a product owner (Group Manager A, 2015). It is of great importance that the product owner understands the consequences brought by decisions. For instance, adding extra features to an existing product backlog might delay the overall duration of the project, unless something else is removed from the product backlog (Scrum Master A, 2015). Therefore, clear communication between Scania IT and Scania regarding the product owner’s responsibilities and the expectations on the product owner is key for a successful incorporation of the role of product owner.

5.1.6 Agile teams in an organizational context

In section 3.2.6 three types project team distribution are described. Compared to what has been described in section 4.1.6 about the four projects subject to the performed case study the first part of this section will attempt to categorize the projects according the types of teams. Table 5.1 summarizes the team distribution of the four projects.

<table>
<thead>
<tr>
<th>Project</th>
<th>Project 1</th>
<th>Project 2</th>
<th>Project 3</th>
<th>Project 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>A</td>
<td>B</td>
<td>-</td>
<td>B</td>
</tr>
</tbody>
</table>

Table 5.1: Pairing the studied projects with the A-B-C-types.

Firstly, Project 1 relates to type A, i.e., the project consists of isolated teams which are not cross-functional and the project work is partitioned across the teams. According to theory, this will eventually create team with expertise within a specific area, which violates Lean and Agile principles. Secondly, Project 2 relates to type B, i.e., the project is organized in relatively
autonomous cross-functional teams connected by a Scrum-of-Scrums where Scrum Master of the teams meet regularly. According to theory, this creates a co-equal environment and encourages communication, cooperation, and cross-fertilization. Thirdly, Project 3 has been hard to categorize considering one team carries it out alone. Fourthly and lastly, Project 4, like Project 2 relates to type B, as the two project teams are partly separated and communication in-between the teams is performed via the Scrum Master.

Although, type C does not relate to any of the project setups found in the case study, it can be related to what Scrum Master A (2015) describes about partly separated team. i.e., separating the members of a team among several locations could work as long as all team members separated. The issues with non-collocated teams, according to Scrum Master A (2015), arise typically when only a few team members are not collocated. Scrum Master A (2015) further explains that a completely separated team will have to find other channels for communicating, while a partly collocated team can communicate orally, meaning the distributed team members will be forgotten. Similarly type C, in literature, is defined by each team having members at multiple location.

5.1.6.1 Organization of teams
As Moe et al. (2009) between team structure and organizational layout is a potential cause of frustration for both developers and management. Related to this several interviewees have expressed that Scania IT’s organizational layout affects team agility negatively, and moreover, Group Manager B (2015) claims that management can sometimes feel they are loosing control in Agile software projects.

Collocation was one major issue that was identified during the case study, and several interviewees among other Project Manager B (2015) claims that the advantages are through the roof. According to Melo et al. (2013), there are surely advantages that come from collocated teams, and Moe et al. (2009) exemplifies that collocated teams will generally improve communication which will greatly help problem solving. Moreover, Lindvall et al. (2004) claim that software development project agility will be enhanced by collocating development teams, i.e., collocation of teams would help increasing the project’s ability to respond to changes. However, Moe et al. (2009) and Melo et al. (2013) both agree that collocation also comes with some disadvantages, namely; lack of privacy, interruptions during work, lack of individual recognition, and disconnection from other teams. The latter disadvantage was also identified in the case study as, for example, Developer A (2015) claim that separating the project team from other teams working on the same software would cause communication issues since the teams have a need for integrating their respective work. On this topic, Moe et al. (2009) argues that the organization has a need to balance the individual level and team-level autonomy in software development teams.

5.1.6.2 Team composition
At Scania IT developers are usually organized in teams but architects, system-analysts and similar roles are generally not a part of the team (Maintenance Manager A, 2015). Development teams are normally not involved in the pre-study phase (Group Manager A, 2015; Agile Coach
A, 2015), why people outside the team perform initial requirements analysis and time estimates, the setup affects team autonomy. i.e., the team lacks the opportunity of making its own destiny, which will also decrease commitment. Thus, Scania IT would benefit from including the architects and similar roles in the project teams, which also was expressed by Agile Coach A (2015). This aligns with the theory outlined in section 3.4.6 that, e.g., claims that cross-functional teams will increase team learning and productivity. Literature further prescribes that team diversity does not only relate to having different roles and competencies within the team, but also other individual attributes such as gender and ethnic background.

Moreover, Developer B (2015) tells that within his team all developers have different backgrounds and competencies, however, stresses that no one has a different role from the others. Related to literature this setup helps an organization avoid creating single competencies and enhances team learning. For example, Campion et al. (1993) and Watson et al. (1993) argue that learning increases within multi-knowledged teams. Project Manager C (2015) also claims that single competencies are an obstacle for real Agile software teams.

Another issue related to team composition at Scania IT is that the customer does not understand that adding more developers to a project does not necessarily shorten time-to-market (Maintenance Manager A, 2015). This relates to Melo et al.’s (2013) findings that staff turnover have an impact on Agile team productivity. i.e., adding or removing people to a development team causes disturbances within the team which means that productivity will most likely drop, and hence, more people is not always better. Scrum Master C (2015), for example, tells that it is important to listen to the team when thinking of introducing another developer. However, Melo et al. (2013) also claim that there are positive effects from adding new staff, e.g., introducing new thoughts to the team through increasing diversity. On this topic, Project Manager B (2015) claim that in their project some developers like that the project is growing while others hate it.

### 5.1.6.3 Communication

Good communication is always an issue in organizations and in projects, why e.g., Stelzer and Mellis (1998) identify it as a critical success factor in software development. As reported in section 3.2.6.3 the number of communication channels is an important aspect to remember when it comes to teams. That is, the number of communication channels increases fast as team size increases. This could be one of the reasons why one of the studied project had felt a need to split there two development teams into three teams because the teams hade grown to big, as expressed by Scrum Master A (2015). The corrective action of splitting the teams into smaller teams have meant that the number of communication channels for each developer within a team has decreased, and communication in-between the teams are handled by the Scrum Masters. Ensuring that software development teams are limited in size is also supported by, e.g., Cockburn and Highsmith (2001).
5.2 Measures

When the authors studied project measures in general, and software metrics in particular, at Scania IT, it was found that the use of measures are limited or circumstantial at best which was argued for in section 4.3 Measures. However, as was also established in the empirical findings many employees at Scania IT have good ideas of how to use measures effectively to communicate project success and status.

For example, Project Manager C (2015) expresses that all teams have a need to measure quality of work and their velocity. This aligns with what Zhang and Zhang (2014) and Brown et al. (2013) describes that metrics can indeed be used to improve predictability of outcome and increase manageability of software projects.

Continued in this section the authors will analyse Scania IT’s use of project measures, related to what theory describes as good practices, and also try to identify potential improvement areas where suggestions will be made in the conclusions section.

5.2.1 Software metrics

As explained in the background, the study of this scope does not cover what software metrics to be used by an IT-organization but rather how selected metrics can be used to communicate status and progress in software development projects. This section thus briefly explains how software metrics are used at the case company today and how this relates to prescribed practices in literature.

As explained in section 4.3.1 Software Metrics many interviewees express that software metrics are not used to a satisfactory extent at Scania IT. Compared to literature this issue is not limited to Scania IT. For example, Fenton and Neil (2000) state that software metrics are yet to reach its full potential when used on software projects in organizations. The paragraphs below, hence, tries to explain what potential benefits Scania IT could receive from increasing the use of software metrics.

Firstly, Scrum Master C (2015) express that he believes the most useful aspect of software metrics would be attempting to forecast cost, risk and degree of customer satisfaction. According to Zhang and Zhang (2014) this is one of the benefits of utilizing software project metrics in organizations. Secondly, both Zhang and Zhang (2014) and Fenton and Neil (2000) argue that software metrics can assist in identifying risks, estimate cost of development and predict software quality. This aligns with the wishes of Project Manager C (2015) that all teams should want to measure their quality and velocity, which would according could be done with software metrics.

As described in section 4.3.1 Software Metrics, software metrics are generally further categorized into project metrics, product metrics, and process metrics. The use of the latter two at Scania IT is further discussed in the following two sections.
5.2.1.1 Product Metrics

As Zhang and Zhang (2014) explain, software product metrics mainly aim at calculating quality attributes for software products, i.e., the code itself. The authors exemplify that metrics for reliability, usability and maintainability are categorized as software product metrics.

As traditional software metrics goes, regarding code metrics, most interviewees tell that it is used far too little at Scania IT. For example one of the interviewees, a former developer, claim that he only knows of one use of Software Metrics at Scania IT, referring to the use of a software metrics tool called SONAR that the Java development teams sometimes use. If compared to Zhang and Zhang’s (2014) definition of software product metrics, the lack of use of code related metrics, e.g. lines of code, would mean that maintainability of the software product is unknown. i.e., the development teams at Scania IT lacks understanding of how code produced today affect the ability to extend the software in the future.

Related to Zhang and Zhang’s (2014) sub-category of reliability, several interviewees have expressed that testing, and specifically automated testing, of software is an issue at Scania IT.

The case study has, hence, shown that Scania IT does not use software metrics to a satisfactory extend and more importantly that there is an eminent risk that software quality is severely disregarded because of lack of knowledge of the magnitudes of the risks taken when quality of the applications are sometimes deprioritized in favor of features and functionality.

5.2.1.2 Process Metrics

If Scania IT lacks effective use of software product metrics the situation regarding software process metrics does appear better although there are room for improvement.

Firstly, many interviewees have expressed that their team uses burn-down charts to visualize status and work remaining to be done, although, most teams keep the charts internally. According to Mahnic and Zabkar (2012) a major advantage of measuring remaining work is to visualize and communicate to the product owner how adding or removing functionality impacts completion date. As, for example, Group Manager A (2015) has expressed that the product owner sometimes does not understand that adding requirements will affect the time and cost of the project, why using the burn-down charts in communication with the customer could be a suitable solution.

Moreover, burn-down charts at Scania IT are often related to hours rather than story points. This is an effect of the time reporting structure of Scania IT, which makes it natural for the project to estimate tasks in hours rather than the more fictive story point unit. Although, as the authors have understood from the literature review that story points is the prescribed base for burn-down charts theory does not claim that hours necessarily worse to use. However, Mahnic and Zabkar (2012) state that a common mistake is that velocity is not predicted by experience from previous sprints and that team composition is often changed throughout the project. Thus, using a burn-down chart related to hours rather than story points could very well work as long as the team velocity is predicted form the average number of hours it takes to complete a requirement for the current team.
Secondly, some interviewees have also spoken about the use of a measure known as change request intensity. According to Lee and Xia (2010) the ratio between the amount of change requests received and the amount actually implemented are a measure of the software team response extensiveness. The higher the response extensiveness the higher the software development agility (Lee and Xia, 2010; Lindvall et al., 2004). However, at Scania IT change request intensity seems to rather be used to estimate the current workload of a development team. For example, Project Manager A (2015) tells that he measures the number of received and the number of completed change requests to show the project steering group work load of the respective development team related to the project. Thus, although, the measure is utilized at Scania IT, and the interpretation of the measure seems reasonable to the authors of this report, the measure could, hence, be used to also signal how Agile a project actually is. Furthermore, related to response extensiveness, theory also suggests measuring software team response efficiency (Lee and Xia, 2010). Those two measures in combination would be helpful to determine agility of the project team (Lee and Xia, 2010).

A few interviewees also mentioned another metric, that is closely related to change request intensity, namely a measure of team lead-time. i.e., a measure of the time form a new change request arrives to the team to it has been fully implemented and delivered to the customer. However, several issues with this measure has also been mentioned by the interviewees. The issues are described in 4.3.1.2 Process Metrics and, briefly, relate to:

- organizational layout, that affects the lead time more than the efficiency within the team;
- way of working in various teams, it is difficult to compare the measure between various teams since they have different set ups and prerequisites;
- organizational priorities, i.e., support errands are mostly prioritized why lead time for change requests related to new features could be misleading; and
- maturity of the team, i.e., some teams automatically deliver new features directly to the users at the end of a sprint, while other teams hardly deliver automatically to test.

Thus, the lead time measure, while having potential, is probably not a good measure to use given the current situation at Scania IT.

5.2.2 Project Scatter Factor

Figure 5.1 below shows the calculated project scatter factor for ten randomly selected projects at Scania IT. As the scatter-diagram shows most projects have a project scatter factor that is way above one (1) which According to Hendriks et al. (1999) is described as the target value for the project scatter factor.
The dashed line in the figure shows the average project scatter factor for all ten projects and has a value of about 15, which is far from the optimal value. Moreover, the project that has the lowest project scatter factor out of the ten is project 5 with a scatter factor of 3.7 which is closing in on the optimal value, however, still off. Thus, it is reasonable to assume that most projects at Scania will have a project scatter factor that is above the optimal value.

One the one hand, as Scrum Master B (2015) claims most projects at Scania IT are carried out through being dispatched to the development teams in the respective office. Each project is divided into requirements that are translated into change requests that are handed out to a team to be implemented (Scrum Master B, 2015). This set up is one possible explanation to why Scania IT experiences such high project scatter factor. On the other hand, even when a designated project team is created the developers of the team usually still have responsibilities besides the project. For example, Scrum Master C (2015) has expressed that he had to fight for a project team that was dedicated solely to the project without any external responsibilities, e.g., support.

As figure 3.3 in the Theory chapter shows, an increased Project Scatter Factor leads to lowered team commitment and decreased efficiency in the work carried out by the project team. On the other hand, if a company allows a high project scatter factor it is also easier to allocate scarce resources between projects as it is possible to move part-timers around. However, this in turn makes for a more complicated resource allocation structure. (Hendriks et al., 1999)

5.2.2.1 Discipline Scatter Factor

As was argued in section 5.2.3.2 Resource allocation Scania IT generally does not consider how hiring resources in one office will affect resource allocation in other offices. Moreover, Cooper and Edgett (2003) advocate that organizations can benefit from using tactical solution to the resource allocation crunch, i.e., visualize the current situation in order to create a common understanding of multi-project environment. The Project Scatter Factor is one way of visualizing
the resource problem, but the authors of this report would like to introduce a slightly modified approach that help explain how hiring philosophy of one office affects other offices.

The Discipline Scatter Factor is defined as the average number of assignments per employee of a certain discipline, i.e., at Scania IT due to its functional organizational layout this becomes the average number of assignments per employee pertaining to a particular office. The figure 5.2 shows an example that is explained in the subsequent paragraph.

![Figure 5.2: Discipline Scatter Factor](image)

The figure clearly shows that the number of projects is increased from Month 1 to Month 2, but the Discipline Scatter Factor of the Project Management Office is unchanged, i.e., the office has increased its number of employees accordingly. The Solutions office, on the other hand, has experienced an increase in the Discipline Scatter Factor due to the increased number of projects and not enough recruits to cover up. When comparing Month 2 and 3 at the Project Management Office the number of projects have once again increased drastically but this time the Discipline Scatter Factor increases with it, i.e., the office has assigned project managers to the new projects without recruiting. Similarly to the change from Month 1 to 2, the Solutions Office once again experiences an increase in the Discipline Scatter Factor from Month 2 to 3, i.e., recruitment frequency is still too low. Finally, from Month 3 to 4, the number of projects does not change, and likewise is the Discipline Scatter Factor for the Project Management Office constant, i.e., the office has no new recruits. This time, the Solutions office have actually recruited more heavily, managing to lower the Discipline Scatter Factor.

The new-defined scatter factor live under the obvious assumption that the different offices in an organization have an impact on one another regarding workload. I.e., in this case it is assumed that a new project arriving at the Project Management Office will increase the workload at the Solutions Office. Without this hypothesis the graphs depicted and described above may still look the same, but it would be impossible to explain the behaviour of one graph by the help of the others. As was established in section 5.2.3 on project portfolio management and resource
allocation Scania IT lacks a functioning project portfolio and solves the accompanying resource crunch by scattering resources as necessary to sufficiently allocate resources to all incoming projects. Thus, the assumption behind the Discipline Scatter Factor appears to be valid at the case company.

The Project Scatter Factor and the Discipline Scatter Factor are most likely correlated as both are alternative measurements for how many simultaneous tasks an average employee have. However, in contrast to the Project Scatter Factor that only account for project related work, the Discipline Scatter Factor also includes other types of work, e.g., work in the functional organization. Therefore, in an organization that carry out most of its work in project form without the underlying functional offices, the two measurements will probably coincide even more extensively. Nevertheless, this report will not dig any further into the nexus of the two.

In conclusion, the Discipline Scatter Factor helps visualize how recruiting in one office affects task switching in another office. In a standard software development project, it is reasonable to assume that the number of developers far exceeds the number of project managers. Thus, if the Project Management Office hires one new project manager to manage one new project, the Solutions Office will most likely have to recruit tenfold in order to not increase the Discipline Scatter Factor, i.e., not increase task switching.

5.2.3 How to use measures effectively
As argued for in section 4.3.3, Scania IT does not use measures enough in their communication with the steering group. A few reasons for why this is not done was also established, and include; project stakeholders are generally only interested in cost estimation during the project and only after the project do they question the result; and software metrics are considered too technical to be of any use to the customer. However, after analysing the empirical data and specifically what the Product Owner A expressed regarding communication of metrics it appears as if the customer would indeed be interested in learning about more than cost during the project and get an overview of the product quality. Staron (2012) also finds that providing managers with daily updated information about relevant areas of the project would be beneficial.

Therefore, it appears that Scania IT should indeed communicate software metrics to the product owner and that he or she is interested. In addition, just as the customer sometimes do not understand agility, several of the interviewees, among others Steering Group Representative A (2015), have expressed that the customer rarely understands the software development process, which could be a potential reason for the misunderstanding expressed above that the customer is not interested in software metrics. i.e., the customer does not understand what benefits lie in reviewing software metrics for the project. Thus, it is likely that it will be Scania IT’s mission to introduce the product owner to the concept of software metrics. Similarly, Project Manager B (2015) claims that Scania IT lack experience in what measures the customer is interested in.

Another aspect is that used measures should be constantly evaluated and improved. For example, Melo et al. (2013) argue that new circumstances, e.g., new software engineering practices may cause measurements to be outdated or irrelevant. On this topic the authors of this report have
found that Scania IT works quite well at the moment. For example, Project Manager B (2015) states that since the team has only been working according to Agile methodologies for a few months they are not yet fully mature in their communication of measures and status. Similarly, Project Manager B (2015) tells that measures should be used to learn and improve as a team, which aligns with the findings from the performed literature review.

Moreover, several interviewees have expressed that developers on the team may perceive too much pressure if metrics are used that visualize how badly the current state of the application is. This is important to bear in mind when adopting new measures. However, the authors of this report agree that there is an evident risk that Scania IT would, hence, disregard flaws in software development simply because they do not want to upset the developers. A more reasonable approach would be to use measures regardless of how bad the situation is, and rather, encourage the developers to improve through giving them time to focus on code quality. It is likely that this will eventually benefit Scania due to less incidents.

5.2.3.1 Measures for decision making

According to Fenton and Neil (2000) the use of software metrics should strive to build management decision support tools combining various aspects of software engineering to help managers predict and assess outcome of software projects. Several interviewees have also expressed similar ideas. For example, Scrum Master A (2015) claims that software metrics could be used to visualize the condition of the software to the product owner and, moreover, the potential risks with prioritizing new features ahead of quality over time. Project Manager C (2015) also says that more visualization of measure is needed since the product owner should be able to simple look at information radiators and make decisions from there. Zhang and Zhang (2014) also state that measures can be used to provide necessary information, related to cost, rate of progress and quality, for project decision makers with the intention of initiating feasible actions to reach the desired output.

Another aspect of communication software metrics to the product owner, and the steering group of PPS, relates to Staron’s (2012) finding that a measure should be located on such organizational level that the elapsed time from when a change in the metric is observed to when a corrective decision can be made is not too long. Thus, since the product owner is responsible for prioritizing the product backlog and, hence, can affect what the development team will spend their time on, it would be beneficial to communicate metrics related to, e.g., quality to the product owner to ensure that a well informed decision is made. For example, Scrum Master A (2015) claim that their project focuses solely on developing new features and disregards quality which will eventually cost the customer a lot of money. Project Manager B (2015) claim that the product owner is aware of the risk and has made the decision that features are that important that quality can wait. However, as no measure of quality for the application is used, nor communicated to the product owner, the decision is made based on a risk of unknown magnitude.

Furthermore, Staron (2012) argue that the most effective measures are those that warn about potential problems rather than monitor status, although, the author also acknowledges that the latter type have their upsides too. What this means for Scania IT is that focus should be on
measures that visualize risk rather than status. However, Staron (2012) also claims that a good use of status-measures are to help follow up that decisions are implemented. This further consolidates that software metrics can and should be used to facilitate decision making in software projects.

5.2.3.2 Measure what counts
As argued for in section 3.3.3.3 Measure what counts an organization can easily have too many metrics. This will cause a heavy workload for the developers and force them to focus on updating data for the measures rather than produce software. Thus, while sufficiently many measures can be used to improve software development processes, software quality and productivity within the teams, too many can likewise decrease performance. Literature further claims that measures can indeed help team agility, albeit, it may destroy the same if overused.

Project Manager C (2015) touches on this topic when he claims that all measures should be used because we intend to learn something from them. i.e., if measures are not used to teach the organization to improve they are most likely dispensable. Project Manager C (2015) further gives the example of reporting time, saying that Scania IT spend to many hours reporting their hours which would be fine if we used the numbers to learn from them. However, the authors of this report find that it is not likely that the time reporting task will ever disappear since it relates to Scania IT’s need to invoice all costs to the parent company Scania.

A good policy for sorting out useful measures from those simply presenting statistics are to review whether the measures are connected to corrective actions. i.e., if the measures are related to actions that will be taken to correct faulty values it is fine, compared to a measure where the organization simply will acknowledge that the measure has gone bad without doing something about it. A former developer (2015), for example, claim that in his development team they used to measure number of bugs and code complexity, however, they had no defined target value, and no corrective actions were taken when the measure went sour.

In addition, if the project wishes to receive the necessary attention from the product owner it is also relevant to reduce the number of used metrics as the product owner doubtlessly has other tasks to attend to besides being product owner. As an example, Project Manager B (2015) state that their project team listen to feedback from the product owner whenever they show him a new measure to ensure that they communicate only relevant information to avoid loosing his interest.

5.2.3.3 Performance and result indicators
The idea of performance and result indicators are briefly introduced in section 3.3.3.4, and to summarize it is about understanding that some measures are harder to affect than others. i.e., Key Result Indicators are the result of many activities why it is harder to realize what actions are needed to improve in the measured area. Similarly, Key Performance Indicators are tied to an individual or a team, be future oriented and only focus on one single area of measurement making it easier to realize how to improve. This can be related to what Staron (2012) states about software metrics. The article argues that the most effective measures are those that warn about potential problems rather than monitor status. That is, measures that warn about problems are easier to relate to corrective actions and are, hence, to be considered KPIs, and while status
measures are KRIs. Furthermore, the author claim that status measures are of good use to follow up how made decisions affect performance, which again relates to Parmenter’s (2010) idea about KRIs.

Scania IT have a clearly set policy to strive for nine out of ten projects delivering on time and according to budget. However, from what has been established in chapter 4. Empirical findings it is reasonable to assume that it is the organization does not manage to live by that goal today. Relating this measure to Parmenter’s (2010) theories on performance measures, it is evident that the measure is a KRI. That is, there are many factors affecting the value of the measure, e.g., initial estimations of time and budget being incorrect; product owner adding requirements during the project; and support errands hindering the development team from focusing on the project. Therefore, this measure is extremely hard to improve for the organization and the responsibility cannot be tied to one individual to ensure that the target value is met. A better measure would be related to customer satisfaction as stated by Scrum Master C (2015), as long as the product owner is aware that the initial estimates were to optimistic; added user-demands causes delay; and that support errands interrupted the project, the customer may still be happy with the project outcome. Related to Parmenter (2010) customer satisfaction is perhaps not a perfect measure since it does depend on multiple aspects, but is definitely better. If Scania IT were to measure the customer’s happiness with an ongoing project, all project workers would solely have to focus on always doing the right things according to the customer.

Finally, Parmenter (2010) states that a KPI should always have corrective actions related to it. i.e., whenever the measure is off its set target value, employees should understand how to act in order to fix it. Therefore, the measure can also be tied to an individual or team. This can be related to Staron’s (2012) findings that a measure should be located on the level of the organization where the people who can decide to take actions against a faulty value work. Thus, a measure that only the CEO can affect should not reside at the developer level of the organization according to Staron (2012), which also aligns with Parmenter’s (2010) idea that most measures relevant to the CEO are KRIs and can not be acted on by the individual employee. One example of a measure at Scania IT that the authors of this report have found lacks corrective actions is the Project PULSE, as explained in section 4.1.2.3, where each project manager report the current status of their project(s). It appears as if a project can report being red for several weeks in a row without anyone reacting apart from asking the project manager if they need help. The authors of this report believe that the measure is, hence, a KRI, but with the potential to be more. It is already reported regularly, why the only thing missing to make it really useful is corrective actions that can be tied to an individual. For example, Scania IT could start keeping score of how many weeks of a project’s total duration that it was reported red. Once a project is completed that has a majority of its duration reported red status, the project should be thoroughly investigated to find out why. After having investigated multiple projects that was more or less red during their execution it is not unlikely that Scania IT will find correlations between those projects that can help explaining why and, hence, corrective actions can be put in place to avoid similar mistakes from happening again in future projects. In conclusion, it is important to always have an action plan for any used measures.
6. Conclusions

This chapter attempts to conclude the discussion in chapter 5. Analysis, aiming for answering the research questions established in chapter 1. Introduction and that have been the base for the entire literature review presented in chapter 3. Theory and the case study performed at Scania IT outlined in chapter 4. Empirical findings. Finally, this chapter ends with a section on how this report relates to the academic literature on the subject and suggests areas that may be of interest for future studies.

6.1 Answering the research questions

6.1.1 How is agility in Agile software development projects affected while the surrounding organization regards IT-services solely as support for its core business?

In chapter 5, Analysis several factors that affect agility at the an IT-company whose raison d'être is supporting a company's main business is identified, namely; its setup for infrastructure; how the company utilizes teams.

Firstly, the IT-organization will be remarkably affected by the organizational layout of the parent company. In the example of the case company the parent company was organized in functional departments, e.g., R&D and Finance. Since each of the departments may have IT needs but the departments lack the ability to prioritize amongst each other's IT needs, all acute needs of each department is passed on to the IT organization. Thus, the workload of the IT organization may significantly fluctuate from month to month and when resources become scarce the prioritization problem of the parent company is effectively passed on to the IT organization. On the one hand, an alternating flow of work will negatively affect agility in software development projects directly. On the other hand, in the case when a project has multiple stakeholders the prioritization problem leads to the project team having difficulties to decide how to prioritize the project backlog which in turn leads to difficulties in sprint planning that directly affects team agility. Several issues related to the prioritization problem also affects resource allocation which will be discussed in section 6.4.

Secondly, at the case company support and maintenance was a prioritized task, which can be related to the fact that the parent company views IT as support for its own business, i.e., IT is something that should always work. Thus, new product development projects is deprioritized in favor of supporting applications already in use. As was discussed in the analysis-chapter projects are prevented from planning the full capacity of the sprint as the teams need to reserve a significant amount of hours in case of any urgent support errands. Once again, the situation at the IT organization related to its parent organization affects the teams' ability to plan their work which according to literature negatively affects software team agility. However, the developers
are usually the best suited to provide support, why a set-up like this is reasonable although it does affect team agility.

Thirdly, the parent company generally has no internal motivation for understanding the way of working in an IT organization, and more specifically the parent company does not incorporate Agile methodologies in its own work. Thus, the parent company may lack knowledge of the Agile approach and Agile practices, e.g., customer involvement which, as was established in the analysis, decreases the IT organization's possibilities to be fully Agile. Moreover, a lacking knowledge of Agile principles was also identified at the IT organization itself, although, less prominent.

Fourthly, the organizational layout of the IT organization was greatly affected by the organizational layout of the parent company. Thus, the IT organization is divided in functional departments rather than a project team based organization, which would be advantageous for project work as was established in the analysis chapter. For example, the infrastructure department has been identified to cause issues for agility in software projects, through, e.g., putting restrictions on deployment to production and testing environment. This, in turn, limit the project teams ability to get valuable feedback at the end of each sprint. i.e., the project can work iteratively, however, one of the main reasons for doing so is to get continuous customer feedback. Therefore, the organizational layout also negatively affects agility.

Finally, as was argued above the organizational layout is significantly impacted by the parent company. Moreover, as was established in section 5.2.6 Agile teams in an organizational context teams are greatly influenced by the surrounding organization. In particular, it was identified that the organizational layout affects the project team types (A, B and C) that was further linked to team agility. Additionally, collocation, which heavily relates to team agility, is potentially restricted by the organizational layout.

In summary, the conclusions above prove that the setup where IT is facilitating an organization's core business, agility is usually negatively affected, and in particular sprint planning and feedback is affected the most.

6.1.2 How does a traditional project stage gate model affect agility within Agile software development projects?

As described in section 5.2.2.1, the stage gate model at the IT-organization mainly serves as a framework for project managers and management through providing assurance and stability to the overall project steering in their daily work. Moreover, it was established that the stage gate model does not affect the project teams' daily work. As a result, there is no obvious reason for an IT organization to not successfully combine Agile methodologies with a traditional stage gate model.

An issue discovered when studying the PPS-model is related to the pre-study phase. Currently, the PPS-model implies a rigid pre-study phase where, e.g., much work is done in defining requirements and time estimations. However, a pre-study phase is necessary but as established
the duration of the pre-study is commonly too long. Consequently, the project teams' ability to respond to change and, thus, their ability to become fully Agile are limited.

The second issue that was identified relating to the PPS-model and team agility is that PPS generally advocates that decisions are escalated to a project steering group rather than distributed to the teams as is common in Agile methodologies. However, the analysis in section 5.2.2.3 Steering group and decision-making also found that the PPS-model provides a useful framework that facilitates decision-making related to projects why it also has advantages. Although, the steering group has to be aware of how badly an order may affect the project team, and as argued for in the analysis the steering group should always strive to include the project team in their decisions.

As was already argued in the theory-chapter, in section 3.2.2. Combining Agile with a stage gate model a stage gate model can be successfully combined with Agile approaches, which was further addressed in section 4.2.2 and 5.2.2, an conclusively it was established that the PPS-model is likewise combinable with Agile practices at the case company. Moreover, combining the two could also prove extremely useful as the organization may benefit from both models’ advantages while compensating for their respective disadvantages. However, in order to combine the two approaches, both have to be adapted to suit one another. Above, two issues with the PPS-model and agility at the case company is outlined which have to be managed in order to combine the models. Firstly, it is recommended that a stage gate model should not prescribe an extensive pre-study as it rhymes badly with agility, and secondly, decision making should include the project workers as much as possible.

6.1.3 How resource allocation at Scania IT affects software project team agility?
As discussed in section 5.2.3.1, the case company lacks the ability to prioritize their project portfolio, which results in too many ongoing projects and, eventually, a complex resource allocation. Additionally, the authors have established that this, through, e.g., task switching, negatively affects the agility of project teams at the case company.

As established in section 5.2.3.2, there are solutions proposed by theory that may not be suited for the current situation at the case company. Firstly, since the ability of prioritizing among projects at the case company remains insufficient, the possibility to divide the projects according to the modified BCG-matrix is not an option. Secondly, portfolio management solutions is not an option, due to identical reasons. Lastly, the utilization of ring-fencing at the case company serves no purpose, due to the fact that maintenance and support always are prioritized.

However, there are two possible solutions that may assist management at the case company in allocate resources more efficiently. Firstly, the IT organization should aim at allocating full-time resources to increase dedication. Secondly, the IT organization should attempt to visualize the situation internally, in order to reach a common understanding about the severity of the topic. The authors of this report suggest doing this through utilization of the Project Scatter Factor. As described, the target value of the Project Scatter Factor is one (1), and as presented, projects at
the case company exceeds this by far. Additionally, in section 5.3.3.2, the authors present the Discipline Scatter Factor that could assist the case company in visualizing how hiring resources in one office affects resource allocation in other offices.

6.1.4 How can performance measures, and particularly software metrics, be used to concurrently monitor and communicate status and facilitate decision making within Agile software development projects?

As was argued for in both section 4.3 an section 5.3, the use of software metrics at the case company is fairly limited and circumstantial today. However, it was also found that the organization could greatly benefit from increasing the use of measures in generally, and software metrics in particular.

The analysis of measures established that project teams have the need for measuring aspects related to team velocity and productivity as well as the product quality. For several reasons, the analysis also points towards quality measures being more important than measures related to velocity and productivity. The first identified reason was that several interviewees express that IT development is not necessarily sped up form measuring team productivity, and that velocity is more relevant when estimating possible delivery dates. The second reason was that, even if productivity and velocity measures could be used to estimate delivery dates and completion time, the case study also found a mismatch between initial time estimates and actual time to complete a feature why any estimated delivery dates will be equally of. Therefore, although productivity and velocity measures could be useful, there are too many related uncertainties to draw any absolute conclusions regarding their usefulness at the case company. However, as the case company continuously improve their way of working it is likely that productivity and velocity measures will be more useful in the future.

On the one hand, it was found that introducing measures related to product quality, e.g., product metrics may be a vast improvement for the case company. For example, it was found that the product owner sometimes prioritized features over product quality with the argument that it is worth the risk. With the current absence of product metrics the project team likely, and the product owner definitely, do not know the magnitude of that risk.

Moreover, as was established in the section 5.3.3. How to use measures effectively, software metrics would often be proven more useful to an organization if used for effective decision-making. The analysis also stressed that any successful metric should come with an action plan to correct a faulty value. Thus, if the case company would start to use product metrics, particularly related to quality, along with set target values, whenever the value is off the measures can be used in communication with the product owner that focus has to be on improving the software code rather than only prioritizing functionality.

On the other hand, although the usefulness of productivity and velocity measures are debatable at the case company it was argued in the analysis that they may be used to raise awareness
internally with management and with the customer. I.e., it was found that the case company fought a battle in explaining to project stakeholders that more resources does not necessarily increase productivity, nor does a full-time allocated developer produce value for a full week of 40 hours in the project. Therefore, productivity and velocity measures may successfully be used to raise awareness of issues related to resource allocation and Agile software development teams.

Although, the scope of this study does not cover what measures to use but only how measures can be used in software projects, the analysis also identified several critical success factors in software projects and also potential ways of monitoring them through measures, why a few suggestions is presented below to prove the point of the usefulness of measures in Agile software development projects. All the measures have been argued for in the analysis-chapter, and hence, only a summary are presented here:

- Firstly, team response extensiveness can be used for measuring team agility, and change request intensity can be used to measures the current work load within a software development team.
- Secondly, as was concluded in section 6.2 above the stage gate model at the case company and its impact on project team agility could be improved by shortening the pre-study. Thus, management should measure the length of the pre-study in all projects to monitor if any improvements are made. Through monitoring this measure it may eventually become evident that the pre-studies cannot be enough shortened due to too many tasks being prescribed by the stage gate model why this measure will signal to management that the model needs to be altered to lower its impact on agility.
- Thirdly, the case company should introduce measures related to visualizing the status of the resource allocation crunch. In the analysis to such measures was identified, namely; Project Scatter Factor and Discipline Scatter Factor
- Fifthly, as have already been covered in this section the case company could benefit from utilizing software product metrics, particularly related to measuring product quality. Identifying exactly which product metrics to use is not part of this report as it is potentially a broad enough scope for a separate master thesis project. For example, theory claims that software metrics usually require company specific data and that not all existing software metrics would suit all types of IT organizations.
- Finally, the case company should measure the frequency of delivery to the customer. As was identified in the analysis, if differs significantly from one project to another how often delivery to the customer is possible. This measure could potentially be a good indicator of the software development team's agility and maturity.
6.2 Academic contribution and further research

The performed case study is by several means rather narrow in its scope. Firstly, the case company has a rather unique position for an IT organization as it serves only to support its parent company with IT support, and hence, has no direct ability to affect its situation. Secondly, the size of the study is limited with regards to the number of held interviews in particular, but also because it included only four projects.

However, most of the findings align well with previous research in the field why its reasonable that the study could be generalized to apply to IT organizations in different contexts and with other setups. Moreover, the conclusions outlined above, particularly for question 2 and 4, all seems quite general to the authors of this report why it may actually be applicable to other IT companies. For example, the conclusion that a lack of knowledge of Agile practices at the customer and that it will affect the project teams agility is most generalizable to other business.

Moreover, a thought for the reader could be that the case company is a separate company does not mean that the conclusions are only useable for other IT organizations that have only their parent company as customer. An example could be the IT department of any small, medium-sized or large company whose main business needs IT support but does not in itself sell IT to its customers.

A particular contribution made by this paper is that of the introduction of the Discipline Scatter Factor, as a spin-off of the Project Scatter Factor. Although, the use of the measure has not been proven in practice, it is relevant in accordance with the performed literature review and also the reasoning behind the measures applicability is likely to hold. Thus, the measure could be considered an academic contribution of this paper, however, further research is necessary to prove exactly how accurate the measure is in real life situations.

6.2.1 Suggestions for further research

As established in section 1.2.1, the research questions mainly aim at study how certain context dependent factors affect the agility in software development projects and, thus, consequences for the rest of the organization from the recommendations are left unevaluated apart from its impact on the agility. i.e., what impact would the shortened pre-study of the stage gate model have on, e.g., resource allocation within the organization. Therefore, further research on how the author's propositions would affect the case company and its parent company is needed.

Another aspect of the study that was definitely limited by the restricted time frame was the number of studied projects. The authors of this report chose to study a lesser number of projects to maintain research quality. This way the authors had time to perform an extensive analysis of the gathered empirical data, e.g., all interviews were transcribed verbatim. An alternative approach would have been to simply summarize the essence of all interviews which would have left time to include several more project. Similarly, specifically regarding research question number 4, the authors would have preferred to perform a test run with selected measures to evaluate their usefulness for the case company.
Finally, the scope of the report, as was explained under research question four above, did not include a study of what software metrics would be most suitable to use for measuring product quality at the case company. i.e., literature prescribes many different metrics that all can serve as estimations of software complexity, testability, maintainability and many more aspects that affect the overall quality, but it also clearly states that company specific data and customization of the metrics is needed to ensure its reliability and credibility.

The list below suggests a few related areas that may be potential fields of study for future master thesis projects, at the case company or elsewhere;

- as mentioned above, the consequences of the recommendations of this study remain unknown and, hence, further research on how the IT organization would be affected by the implementation of the suggested actions is needed;
- as have been briefly touched upon above the authors suggest that a thorough study is conducted regarding what software metrics that would be most suitable for the case company to use to measure the quality of its software products. This aligns well with existing literature that metrics benefit from company specific data;
- whether or not the case company could possibly introduce a working project portfolio management technique that does not change its relation with the parent company; or
- perhaps a deeper study in one of the research questions in this report, but with several, case companies participating would enlighten the problem differently and also make the conclusions more generalizable and/or credible.
7. References

7.1 Academic Journals


Abrahamsson, P., et al., 2009. 'lots done, more to do': the current state of Agile systems development research. EJIS 18, 4, p. 281–284.


Drury, M., et al., 2011. The role of experience in Agile software development decision making. The 6th International Research Workshop on IT Project Management, a pre-conference workshop for the 2011 International Conference on Information Systems (ICIS), hosted by the
Special Interest Group for IT Project Management (SIGITProjMgmt) in the Association for Information Systems (AIS), Shanghai, China.


Sutherland, J., et al., 2006. Distributed Scrum: Agile Project Management with Outsourced Development Teams. (Conference)

Sutherland, J., 2011. Ten year Agile retrospective: how can Agile improve in the next ten years?.

102


7.2 Books


7.3 Online resources


### 8.1 Project Scatter Factor calculations

<table>
<thead>
<tr>
<th></th>
<th>Estimated time (h) *</th>
<th>Duration (months) *</th>
<th>Number of resources involved *</th>
<th>Estimated time (Menyears)</th>
<th>Duration (years)</th>
<th>Resources needed to complete in one year</th>
<th>Project Scatter Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>766</td>
<td>19.0</td>
<td>6</td>
<td>766/(8*250) = 0.4</td>
<td>19.0/12 = 1.6</td>
<td>6*1,6 = 9.5</td>
<td>9.5/0.4 = 24.8</td>
</tr>
<tr>
<td>P2</td>
<td>1200</td>
<td>6.9</td>
<td>15</td>
<td>0.6</td>
<td>0.6</td>
<td>8,6</td>
<td>14,4</td>
</tr>
<tr>
<td>P3</td>
<td>18000</td>
<td>17.0</td>
<td>60</td>
<td>9.0</td>
<td>1,4</td>
<td>85,1</td>
<td>9,5</td>
</tr>
<tr>
<td>P4</td>
<td>3000</td>
<td>15.0</td>
<td>16</td>
<td>1.5</td>
<td>1.3</td>
<td>20,0</td>
<td>13,3</td>
</tr>
<tr>
<td>P5</td>
<td>4000</td>
<td>11.0</td>
<td>8</td>
<td>2.0</td>
<td>0.9</td>
<td>7,3</td>
<td>3,7</td>
</tr>
<tr>
<td>P6</td>
<td>25005</td>
<td>26.0</td>
<td>50</td>
<td>12.5</td>
<td>2.2</td>
<td>108,3</td>
<td>8,7</td>
</tr>
<tr>
<td>P7</td>
<td>3500</td>
<td>10.0</td>
<td>15</td>
<td>1.8</td>
<td>0.8</td>
<td>12,5</td>
<td>7,1</td>
</tr>
<tr>
<td>P8</td>
<td>2500</td>
<td>36.0</td>
<td>10</td>
<td>1,3</td>
<td>3.0</td>
<td>30,0</td>
<td>24,0</td>
</tr>
<tr>
<td>P9</td>
<td>13000</td>
<td>36.0</td>
<td>100</td>
<td>6,5</td>
<td>3.0</td>
<td>300,0</td>
<td>46,2</td>
</tr>
<tr>
<td>P10</td>
<td>8000</td>
<td>11,0</td>
<td>30</td>
<td>4,0</td>
<td>0.9</td>
<td>27,4</td>
<td>6,9</td>
</tr>
</tbody>
</table>

* Data gathered from Scania IT

\[
\text{Estimated time (Menyears)} = \frac{\text{Estimated time (h)}}{\text{Number of working hours in one year}} = \frac{\text{Estimated time (h)}}{8 \text{ h/day} \times 250 \text{ working days}}
\]

\[
\text{Duration (years)} = \frac{\text{Duration (months)}}{12}
\]

\[
\text{Resources needed to complete in one year} = \text{Number of resources involved} \times \text{Duration (years)}
\]

\[
\text{Project Scatter Factor} = \frac{\text{Resources needed to complete in one year}}{\text{Estimated time (menyears)}}
\]
8.2 Interview questions

8.2.1 Project manager and Group manager

General
Project context, what is the idea behind it, problem description? Project organization? The interviewee’s role in the project?

1. Please briefly describe the background of the project.
   a. What is the expected benefit for Scania of carrying out the project?
2. Please briefly describe the organization of the project
   a. How many are you?
   b. How many teams are involved?
   c. How are the teams organized?
3. Please briefly describe your role in the project
   a. What are your responsibilities in the project?
   b. How many hours are you supposed to be working on the project?
   c. Have you got other responsibilities besides the project? How many hours?
   d. How does your role and your work help in accomplishing the overall goal of the project?
   e. How do you work with leadership in your project
   f. Please describe how your role relates to the project manager’s role in the project? Regarding collaboration, responsibilities, daily steering, follow-up, etc.?

Gate Model
Getting an understanding of how well Scania’s incorporated Gate model is known throughout the project organization? Moreover, how the Gate model affects work in the various projects.

4. Please describe how your project is related to the PPS model?
   a. How did you prioritize between time, cost, quality and scope in the project
   b. What was the definition of done in your project? (for the entire project)
5. Please describe how your project is working according to the PPS model?
6. Please describe how your project is affected (good and bad) by the PPS model?
7. How you would you say that the gate model affects you, personally, in your work?
8. How does the steering group affect the way of working in the project?
   a. Who gets to be in the steering group?
   b. How do you work with your steering group? (Communication, results, etc.)
   c. What have the organisation demanded of the project?
   d. On what grounds are the success of the project to be evaluated?

Agile
Mapping how well Agile methodologies is known throughout the project organization and how well the project workers feel they are following Agile practices and roles.

106
9. Who decides if a project is allowed to work according to Agile methodologies?
10. Please briefly describe your previous experience in working with Agile Software Development?
   a. How many of your project workers have experience in Agile methodologies?
11. Please, describe how your project is working according to Agile methodologies?
   a. In what ways do you believe that you are working according to Agile methodologies? Why? (Requirements, Customer, Collocation, Sprints, Scrum Master, Artefacts, Documentation, Working product, Retrospectives, Backlog, Lessons-Learned, Scrum board, Testing)
   b. In what ways do you believe that you are not following Agile methodologies? Why?
   c. Would you say that your goal is to work according to Agile methodologies?
   d. Do you believe the project would benefit from increasing the incorporation of Agile methodologies?
   e. How do you communicate status of your work to your fellow project members?
   f. How do you work with daily meetings? i.e., daily stand-up?
12. What guidelines regarding the way of working have you given to the project workers?
13. What do you see as the major hinder for the team to work according to Agile methodologies?
14. How does the project work with/react on continuous changes throughout the project?
   a. Is there a difference between how changes are adopted within and in-between sprints? Why?
   b. Who is responsible for changes during the project? (Customer, Product Owner, etc.)

**Success factors**

*This part aims at understanding the project knowledge of success factors in Agile Software Development projects and possibly finding weaknesses.*

15. How do you measure the progress of the project?
   a. What was the definition of done in your project?
   b. How is it communicated?
   c. Do you have knowledge of whether you are on track with the overall project plan?
   d. Who is responsible for keeping track of progress?
16. What kind of measures are used in the project and why?
   a. How are the measures communicated?
   b. Do you know why the measures are being used?
   c. How do the used measures affect your daily work?
17. From previous experience, how well would you say that projects are able to meet initial goals related to time, cost, scope and quality? Why?
   a. Do you know how daily activities affect the project goals regarding time, cost, scope and quality?
Improvement suggestions

Try to get suggestions that this particular individual would like to see in the project.

18. How would you like to use measures in a project?
19. Do you have any suggestions of measures that would increase the knowledge on project success? Which?

Ending

20. Is there anything you would like to add?
21. Is it ok for us to ask complimentary questions later via email?

8.2.2 Scrum Master

General

Project context, what is the idea behind it, problem description? Project organization? The interviewee’s role in the project?

1. Please briefly describe the background of the project.
   a. What is the expected benefit for Scania of carrying out the project?
2. Please briefly describe the organization of your team?
   a. How many are you?
3. Please briefly describe your role in the project
   a. What are your responsibilities in the project?
   b. How many hours are you supposed to be working on the project?
   c. Have you got other responsibilities besides the project? How many hours?
   d. How does your role and your work help in accomplishing the overall goal of the project?
   e. How do you work with leadership in your project?
   f. Please describe how your role relates to the project manager’s role in the project? Regarding collaboration, responsibilities, daily steering, follow-up, etc.?

Gate Model

Getting an understanding of how well Scania’s incorporated Gate model is known throughout the project organization? Moreover, how the Gate model affects work in the various projects.

4. Please describe how your project is affected (good and bad) by the PPS model?
   a. How would you say that the gate model affects you, personally, in your work?

Agile

Mapping how well Agile methodologies is known throughout the project organization and how well the project workers feel they are following Agile practices and roles.

5. Please briefly describe your previous experience in working with Agile Software Development?
a. How many of your project workers have experience in Agile methodologies?

6. Please, describe how your project is working according to Agile methodologies?
   b. In what ways do you believe that you are not following Agile methodologies? Why?
   c. Would you say that your goal is to work according to Agile methodologies?
   d. Do you believe the project would benefit from increasing the incorporation of Agile methodologies?
   e. How do you communicate status of your work to your fellow project members?
   f. How do you perform work-package estimation? I.e., planning poker?
   g. How do you work with daily meetings? i.e., daily stand-up?
   h. How do you work with continuous integration?

7. What guidelines regarding the way of working have you given to your team?

8. What do you see as the major hinder for the team to work according to Agile methodologies?

9. How does the project work with/react on continuous changes throughout the project?
   a. Is there a difference between how changes are adopted within and in-between sprints? Why?
   b. Who is responsible for changes during the project? (Customer, Product Owner, etc.)

Success factors

This part aims at understanding the project knowledge of success factors in Agile Software Development projects and possibly finding weaknesses.

10. How do you measure the progress of the project?
    a. What was the definition of done in your project?
    b. How is it communicated?
    c. Do you have knowledge of whether you are on track with the overall project plan?
    d. Who is responsible for keeping track of progress?
    e. How do you work with burn-down charts?

11. What kind of measures are used in the project and why?
    a. How are the measures communicated?
    b. Do you know why the measures are being used?
    c. How do utilized measures affect your daily work?

12. From previous experience, how well would you say that projects are able to meet initial goals related to time, cost, scope and quality? Why?
    a. Do you know how daily activities affect the project goals regarding time, cost, scope and quality?
**Improvement suggestions**

*Try to get suggestions that this particular individual would like to see in the project.*

13. How would you like to use measures in the project? (Burn-down charts, etc.)
14. Do you have any suggestions of measures that would increase the knowledge on project success? Which?

**Ending**

15. Is there anything you would like to add?
16. Is it ok for us to ask complimentary questions later via email?

### 8.2.3 Product owner

#### General

*Project context, what is the idea behind it, problem description? Project organization? The interviewee's role in the project?*

1. Please briefly describe the background of the project.
   a. What is the expected benefit for Scania of carrying out the project?
2. Please briefly describe the organization of the project
   a. How many are you?
   b. How many teams are involved?
   c. How are the teams organized?
3. Please briefly describe your role in the project
   a. What are your responsibilities in the project?
   b. How many hours are you supposed to be working on the project?
   c. Have you got other responsibilities besides the project? How many hours?
   d. How does your role and your work help in accomplishing the overall goal of the project?

#### Gate Model

*Getting an understanding of how well Scania's incorporated Gate model is known throughout the project organization? Moreover, how the Gate model affects work in the various projects.*

4. Please describe how your project is related to the PPS model?
5. Please describe how your project is working according to the PPS model?
6. Please describe how your project is affected (good and bad) by the PPS model?
7. How would you say that the gate model affects you, personally, in your work?
8. How does the steering group affect the way of working in the project?
   a. Who gets to be in the steering group?
   b. How do you work with your steering group? (Communication, results, etc.)
Agile

Mapping how well Agile methodologies is known throughout the project organization and how well the project workers feel they are following Agile practices and roles.

9. Who decides if a project is allowed to work according to Agile methodologies?
10. Please briefly describe your previous experience in working with Agile Software Development?
11. Please, describe how your project is working according to Agile methodologies?
   a. In what ways do you believe that you are working according to Agile methodologies? Why? (Customer, Collocation, Sprints, Scrum Master, Artefacts, Documentation, Working product, Retrospectives, Backlog, Lessons-Learned, Scrum board, Testing)
   b. In what ways do you believe that you are not following Agile methodologies? Why?
   c. Would you say that your goal is to work according to Agile methodologies?
   d. Do you believe the project would benefit from increasing the incorporation of Agile methodologies?
   e. How do you communicate status of your work to your fellow project members?
12. What guidelines regarding the way of working have you given to the project workers?
13. What do you see as the major hinder for the team to work according to Agile methodologies?
14. How does the project work with, react on continuous changes throughout the project?
   a. Is there a difference between how changes are adopted within and in-between sprints? Why?
   b. Who is responsible for changes during the project? (Customer, Product Owner, etc.)

Success factors

This part aims at understanding the project knowledge of success factors in Agile Software Development projects and possibly finding weaknesses.

15. How do you measure the progress of the project?
   a. How is it communicated?
   b. Do you have knowledge of whether you are on track with the overall project plan?
   c. Who is responsible for keeping track of progress?
16. What kind of measures are used in the project and why?
   a. How are the measures communicated?
   b. Do you know why the measures are being used?
   c. How do the used measures affect your daily work?
17. From previous experience, how well would you say that projects are able to meet initial goals related to time, cost, scope and quality? Why?
   a. Do you know how daily activities affect the project goals regarding time, cost, scope and quality?
**Improvement suggestions**

*Try to get suggestions that this particular individual would like to see in the project.*

18. How would you like to use measures in a project?
19. Do you have any suggestions of measures that would increase the knowledge on project success? Which?

**Ending**

20. Is there anything you would like to add?
21. Is it ok for us to ask complimentary questions later via email?

**8.2.4 Developer**

**General**

*Project context, what is the idea behind it, problem description? Project organization? The interviewee's role in the project?*

1. Please briefly describe the background of the project.
   a. What is the expected benefit for Scania of carrying out the project?
2. Please briefly describe the organization of the project
   a. How many are you?
   b. How many teams are involved?
   c. How are the teams organized?
3. Please briefly describe your role in the project
   a. What are your responsibilities in the project?
   b. How many hours are you supposed to be working on the project?
   c. Have you got other responsibilities besides the project? How many hours?
   d. How does your role and your work help in accomplishing the overall goal of the project?
   e. How do you work with leadership in your project
   f. Please describe how your role relates to the Scrum master’s role in the project? Regarding collaboration, responsibilities, daily steering, follow-up, etc.?

**Gate Model**

*Getting an understanding of how well Scania's incorporated Gate model is known throughout the project organization? Moreover, how the Gate model affects work in the various projects.*

4. Please describe how your project is affected (good and bad) by the PPS model?
   a. How you would you say that the gate model affects you, personally, in your work?

**Agile**

*Mapping how well Agile methodologies is known throughout the project organization and how well the project workers feel they are following Agile practices and roles.*
5. Please briefly describe your previous experience in working with Agile Software Development?

6. Please, describe how your project is working according to Agile methodologies?
   a. In what ways do you believe that you are working according to Agile methodologies? Why? (Customer, Collocation, Sprints, Scrum Master, Artefacts, Documentation, Working product, Retrospectives, Backlog, Lessons-Learned, Scrum board, Testing)
   b. In what ways do you believe that you are not following Agile methodologies? Why?
   c. Would you say that your goal is to work according to Agile methodologies?
   d. Do you believe the project would benefit from increasing the incorporation of Agile methodologies?
   e. How do you communicate status of your work to your fellow project members?
   f. How do you perform work-package estimation? I.e., planning poker?
   g. How do you work with daily meetings? I.e., daily stand-up?
   h. How do you work with continuous integration?

7. What guidelines regarding the way of working have you received from PL/SM?

8. What do you see as the major hinder for the team to work according to Agile methodologies?

9. How does the project work with, react on continuous changes throughout the project?
   a. Is there a difference between how changes are adopted within and in-between sprints? Why?
   b. Who is responsible for changes during the project? (Customer, Product Owner, etc.)

Success factors
This part aims at understanding the project knowledge of success factors in Agile Software Development projects and possibly finding weaknesses.

10. How is project success defined at Scania?
    a. How would you like to define it?

11. What factors do you believe is most important for project success?

12. How do you measure the progress of the project?
    a. What was the definition of done in your project?
    b. How is it communicated?
    c. Do you have knowledge of whether you are on track with the overall project plan?
    d. Who is responsible for keeping track of progress?
    e. How do you work with burn-down charts?

13. What kind of measures are used in the project and why?
    a. How are the measures communicated?
    b. Do you know why the measures are being used?
    c. How do the used measures affect your daily work?

14. What kind of software metrics are used in the project and why? E.g., Lines of Code etc.
15. From previous experience, how well would you say that projects are able to meet initial goals related to time, cost, scope and quality? Why?
   a. Do you know how daily activities affect the project goals regarding time, cost, scope and quality?

**Improvement suggestions**
*Try to get suggestions that this particular individual would like to see in the project.*

16. How would you like to use measures in a project?
17. Do you have any suggestions of measures that would increase the knowledge on project success? Which?

**Ending**
18. Is there anything you would like to add?
19. Is it ok for us to ask complimentary questions later via email?
På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekomersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådan sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© David Larsson and Tomas Walander 2015