



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

Success factors and measurements of Agile teams

Master of Science Thesis in Software Engineering

Olliver Mattsson Kim Egenvall

MASTER'S THESIS 2017:06

Success factors and measurements of Agile teams

Master of Science Thesis in Software Engineering

KIM EGENVALL
OLLIVER MATTSSON



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Division of Software Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Success factors and measurements of Agile teams
Master of Science Thesis in Software Engineering
OLLIVER MATTSSON KIM EGENVALL

© OLLIVER MATTSSON, 2017.

© KIM EGENVALL, 2017.

Supervisors: Rickard Nilsson, Knowit; Jennifer Horkoff, Department of Computer Science and Engineering; Hiva Alahyari, Department of Computer Science and Engineering

Examiner: Jan-Philipp Steghöfer, Department of Computer Science and Engineering

Master's Thesis 2017:06
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in L^AT_EX
Printed by [Name of printing company]
Gothenburg, Sweden 2017

Success factors and measurements of Agile teams
Master of Science Thesis in Software Engineering
KIM EGENVALL
OLLIVER MATTSSON
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Discovering the success factors of an Agile team can prove important for teams that are not performing as well as they should. Combined with measurements for these factors, organizations can pinpoint where problems exist and fix these problems in order to help their teams perform better. Previous research into the topic of Agile success factors have examined factors in relation to Agile projects. However, what this work has not taken into account are what factors affect specifically the team, and how these could possibly be measured. To find these success factors and measurements, a survey was performed. The results were then combined with factors identified from previous literature. This was followed by a second survey, as well as four interviews where the participants ranked the identified factors. The final result was a list of ranked success factors, where the factors most important to a teams success had the highest ranking. The results also introduced some new success factors not previously discussed in literature. The results indicate that the majority of the most important factors relate to people and team spirit, with some factors related to organizational as well as technical factors. The factor deemed most important was "Good communication and collaboration within the team". Compared to previous research, the findings in this report indicate an importance of team spirit, team environment and team capability as opposed to previous literature which advocates project management process as well as customer involvement. These findings can help when Agile teams are not performing as well as hoped by identifying if these factors are present and if not, helping the team in that area in order for them to be successful.

Keywords: Success factors, Agile, Agile teams, Measurements, Team, Value, Success

Acknowledgements

We would like to extend our appreciation to our supervisors at Chalmers University of Technology, Jennifer Horkoff and Hiva Alahyari, for giving us help and guidance throughout this thesis until the very end.

We would also like to extend our appreciation to Knowit and their customer for allowing us to survey and interview their employees for data. A special thanks to Rickard Nilsson at Knowit, for quick responses when something was needed, providing people to help us the same day we asked about it, as well as his encouraging enthusiasm.

Kim Egenvall & Olliver Mattsson, Gothenburg, 06 2017

Contents

List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Statement of the problem	2
1.2 Purpose of the study	2
1.3 Delimitations	3
1.4 Thesis outline	3
2 Background	4
2.1 Agile	4
2.1.1 Successes and Failures of Agile	5
2.2 Team Dynamics	6
2.3 Studies of success factors	7
2.4 Agile Success Factor Categories	7
2.4.1 Success Factors	8
2.5 Success dimensions	8
2.6 Agile Metrics	10
3 Methods	12
3.1 Surveys	12
3.1.1 The first survey	13
3.1.1.1 Pilot test	13
3.1.2 The second survey	14
3.1.2.1 Pilot test	15
3.2 Interviews	15
3.2.1 Agile coaches	15
3.2.2 Developers & Scrum Masters	15
3.3 Analysis	16
3.3.1 Iterative Categorization	16
3.3.2 Surveys	16
3.3.2.1 Initial success factors	17
3.3.3 Interviews	18
4 Results	19
4.1 Interview with Agile coaches	19

4.2	Survey One	19
4.2.1	Background Information of Respondents	20
4.2.2	Agility of organization	21
4.2.3	Initial success factors	24
4.2.4	Measurements found	24
4.3	Survey Two	27
4.3.1	Background Information of Respondents	29
4.3.2	Ranked List	29
4.4	Interviews	30
4.4.1	Interviewee A	30
4.4.2	Interviewee B	33
4.4.3	Interviewee C	34
4.4.4	Interviewee D	34
4.4.5	Rating of Measurements	35
5	Discussion	38
5.1	Surveys: Differences between findings	38
5.1.1	Team size	38
5.1.2	Overlap in Survey Respondents	39
5.2	Agility of company	39
5.3	Most important success factors	40
5.4	Comparison of the ranked list of success factors to the literature . . .	40
5.4.1	Team Environment	41
5.4.2	Project management process	43
5.4.3	Agile software techniques	43
5.4.4	Customer involvement	43
5.4.5	Delivery strategy	44
5.4.6	Team capability	44
5.4.7	Why are the important factors different from literature? . . .	44
5.5	Interviews	45
5.6	Measurements	46
5.7	Threats to validity	47
5.7.1	Construct Validity	47
5.7.2	Internal Validity	48
5.7.3	External Validity	48
5.8	Future work	49
6	Conclusion	50
	Bibliography	I
	A Interview Script	II
	B Survey One	III
	C Survey Two	XII
	D Ranking from second survey	XX

E Ranking from interviews	XXII
----------------------------------	-------------

List of Figures

3.1	The research process	12
B.1	Top of first page of survey	III
B.2	Bottom of first page of survey	IV
B.3	Top of second page of survey	V
B.4	Middle of second page of survey	VI
B.5	Bottom of second page of survey	VII
B.6	Factor 1-3 of last page of survey	VIII
B.7	Factor 4-7 of last page of survey	IX
B.8	Factor 8-11 of last page of survey	X
B.9	Factor 12-15 of last page of survey	XI
C.1	Top of first page of survey	XII
C.2	Bottom of first page of survey	XIII
C.3	Team spirit success factors	XIV
C.4	Organizational success factors	XV
C.5	People success factors	XVI
C.6	Process success factors	XVII
C.7	Technical success factors	XVIII
C.8	Project success factors	XIX

List of Tables

2.1	Success factors from literature [1]	9
3.1	Modifications to identified success factors from literature	18
4.1	Agile practices that were not used	19
4.2	Experience of respondents from Survey one	20
4.3	Role of respondents from Survey one	20
4.4	Number of people in team from Survey one	21
4.5	Complexity of product from Survey one	21
4.6	What Agile practices are used (results of survey one), the number presented is Number of respondents	22
4.7	Agile practices where more than 50% of respondents answered that they always use it	23
4.8	Agile practices where more than 50% of respondents answered they use it sometimes or more often	23
4.9	Initial list of categories. The description explains what kind of factors were placed in that category	24
4.10	Success factors from literature (Including the ones modified)	25
4.11	Factors found not present in literature	26
4.12	Measurements derived from surveys	27
4.13	Measurements derived from surveys cont.	28
4.14	Experience of respondents from Survey two	29
4.15	Role of respondents from Survey two	29
4.16	Number of people in team from Survey two	30
4.17	Complexity of product from Survey two	30
4.18	Ranked success factors, italic is newly found. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	31
4.19	Ranked success factors cont., italic is newly found. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	32
4.20	Ranking of categories	33
4.21	Ranked measurements, TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	36
4.22	Ranked measurements cont., TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	37

5.1	Relation between high level factors to low level factors	42
D.1	Ranked success factors. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	XX
D.2	Ranked success factors cont. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	XXI
E.1	Ranked success factors. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	XXII
E.2	Ranked success factors cont. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.	XXIII

1

Introduction

Agile software development is growing and being adopted not only by software development teams but also by entire organizations [2]. There are numerous studies that examine whether Agile as a principle works or not, as well as studies showing what factors are most important when it comes to quality and performance of Agile projects [1],[3]. Most of these studies were performed around 10 years ago and stated that the research should be validated after 8 to 10 years in order to see if the factors remain the same or if they have changed. Chow and Cao, 2007 [1] identified five categories of success factors for an Agile project: Organizational, People, Process, Technical and Project with 36 total success factors, that were refined into 12 factors of which 6 were considered critical. They suggested that new success factors may emerge while others may become obsolete or no longer of critical nature in a span of five to ten years [1].

This study aims to compile a ranked list of success factors and proposed measurements for Agile teams that will be compared to, and extend, existing research found in literature. The approach of this study extends the current body of knowledge, as well as validates it, as it is focused towards success factors on a team level, while existing research and literature most often investigate success factors with regards to organizational or project success. Data gathering was done through surveying members of Agile teams for what they believed were success factors that helped them achieve their goals and how they would go about measuring these factors. Once these factors were collected and processed, the members were asked to rate the factors by importance in order to identify the most important success factors. As a complement, interviews were used to gather more substantial feedback and reasoning behind survey response data. The study looks at success factors from the view of Agile teams and the results are then compared to existing literature and the state of the art.

The research was conducted in close cooperation with Knowit, a company specialized in IT consulting, and one of their customers, which is a large company in the telecom service business. The study was performed on Agile teams in the service companies operation in Gothenburg and the sample was based on individuals that are involved in these Agile teams. Consultants from Knowit are present in these teams but this study does not distinguish between a Knowit consultant and an employee at the service company within a team, both are considered participants in the study.

1.1 Statement of the problem

Organizations and companies that are undergoing an Agile transformation from a traditional waterfall process have the need to measure and monitor similar values as was monitored in the waterfall approach as they transition into an Agile way of working. The problem these organizations are facing in the transition is that there is no obvious way to measure how well an Agile team or organization performs in contrast to the old methodology. When using the old methodology it was easier to say if a project was over budget or delayed, for instance. The project had a fixed deadline and a fixed budget so this was easy to measure, but with Agile it is very dynamic and thus harder to measure. This introduces an issue when constructing Agile teams since there are no clear measures of what factors have an effect on the team's success, while it is also difficult to evaluate how the transition to Agile has affected the success of the company. These problems were stated by the company with which the research has been conducted.

1.2 Purpose of the study

The purpose of the study is to identify and evaluate factors that have a great impact on the success of Agile teams in terms of team and company value. The results of the study will provide a benefit to organizations that are using an Agile approach for software development or organizational structure by identifying and ranking factors that have an impact on the success of an Agile team. This holds especially true for Knowit and their customer who are part of the study. The term success is defined by the organization itself, but as an example success may be defined as: reduced delivery schedules, increased return of investment, increased ability to meet with the current customer requirements, increased flexibility to meet with the changing customer requirements and improved business processes [3]. These factors may be used as a guideline on how to create team compositions that generate more value both for the team and the company. Khurum et. al [4][5] argue that value is more than just profit generation, while Aurum and Wohlin [6] explains that value includes product value, customer's perceived value and relationship value between the customer and the company. What organizations consider as value is also relevant for this topic and this is investigated by Alahyari et. al [7].

The results of this thesis may also be used to evaluate existing Agile teams within an organization to identify potential flaws and areas of improvement within the team and organization as a whole that may enhance the success across the Agile teams. It will also provide an insight for management on how the members of the teams identify important factors for the success and performance of a team. This has led to the research questions:

RQ1 What factors are important for an Agile team's success?

RQ2 How can you measure the success factors of an Agile team?

1.3 Delimitations

Originally, the plan was to conduct interviews with managers at the in order to get their perspective on the perceived success factors identified from the members of the Agile teams. However, due to time constraints and numerous attempts to get in contact with the company to schedule the interviews over three weeks with no response, it was not possible to perform these interviews. So instead of this we had prepared to interview employees that we had direct access to. These were employees from Knowit that were working as consultants at the service company in different roles such as developers and scrum masters.

The problem statement provided by the company and its customer consisted of identifying and evaluating factors for the success of an Agile team throughout the entire organization as the customer in question had rapidly changing requirements and was dependent on rapid decisions in order to keep a competitive advantage. It was also requested to identify a way of measuring these factors as well as developing an application that could measure the effectiveness of an Agile team for a general purpose. In collaboration with the thesis supervisors and the company representatives, the scope of the thesis was narrowed down into identifying and ranking factors of an Agile team's success in terms of how it creates value for the team and the company as well as identifying some possible measurements. This limitation was introduced due to the broad nature of the the case as well as the time constraint and resources available.

There exists other research that looks into the topic of success factors [2], [8], [3], [1], [9] Since there already exists research into success factors with regards to projects and organizations, this thesis chose to look at success factors regarding Agile teams, since this has not previously been researched. This is, however, closely related to a combination of the research mentioned previously. Therefore, this thesis aims to fill this gap by looking at how the success factors of Agile teams differ from the success factors with regards to team development or the success of a project.

1.4 Thesis outline

The thesis is structured as follows, Chapter 2 talks about previous research and definitions in the field. It also provides a discussion about success and failure in relation to Agile projects. Chapter 3 explains how the research was conducted. It contains information about how the surveys, interviews and pilot tests for these were performed. It also explains the processing of the acquired data. Chapter 4 presents the results and Chapter 5 provides a discussion. Finally, Chapter 6 presents the conclusions and suggestions for future work.

2

Background

This section will present some initial theory about Agile in general as well as successes and failures when using it. This will be followed by related work in the form of discussing team dynamics and lastly the background will be presented by talking about success factors in literature as well as Agile measurements.

2.1 Agile

Dybå and Dingsøyrr define Agile software development as "a set of practices that have been created by experienced practitioners" [10]. They state that Agile methods are a reaction to previous methods used in plan-based projects and that these methods emphasized a rationalized engineering-based approach [11]. These types of methods claim that problems are fully specifiable and that optimal solutions who are predictable exist for all problems. Compared to the previous methods, Agile ways of working prepare for the unpredictability of projects by allowing people to use their creativity in order to solve problems rather than using a process [11]. In the review by Dybå and Dingsøyrr, they use a definition of agility stated by Ericksson et al. [12] as follows:

“agility means to strip away as much of the heaviness, commonly associated with the traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like.” (p. 89)

Agile methodologies for have been around formally since the Agile manifesto was created in 2001. The Agile manifesto states:

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

The items on the right are still important just valued less than the items on the left [13].

Agile builds on a number of practices and principles. These all lead back to satisfying the Agile manifesto. According to the Agile Alliance there are in total 12 principles and 66 practices [14],[15]. These practices and principles are the foundation that an Agile way of working should be built upon. All the practices and principles can be

found on the Agile Alliance website (www.agilealliance.org).

Agile practices are not always praised as the holy grail in software development, both practitioners as well as academics have been known to be critical of Agile methods [10]. The main focus points of this criticism are:

- The feeling that Agile development isn't new and that it has been around since the 1960s [16]
- Agile's low focus on architecture will bring sub-optimal decisions [17],[18]
- There exists too little evidence to support all the claims Agile practitioners make [17]
- XP practices are hardly ever applied by the book if even at all possible [19]
- Agile methods don't work for larger projects and are more suited for small teams [20]

As seen from previous research, Agile might not always be a good choice, therefore, the section below will discuss some successes and failures of Agile usage in practice.

2.1.1 Successes and Failures of Agile

There are studies that show how Agile has positively impacted everyday work and how it has been of help in achieving higher quality code among other things. For example Schatz and Abdelshafi report on the company Primavera, which moved from a waterfall model to an Agile model using scrum [21]. When moving to an Agile based approach using scrum, they noticed an improvement in the quality of the code. As a result, the amount of defects that were found was lowered by 30% in comparison to previous projects using the waterfall approach. They also noticed that they completed the projects faster than previously. A project that would have previously taken them 14 months to complete now only took 10 months. This was due to the fact that instead of having two separate releases with separate functionality they could put everything into a shared backlog and work on the most important things first. This resulted in an earlier release that were a combination of the two separate releases previously planned.

Another example of how moving to an Agile way of working improved the success of projects is provided by Karlström and Runeson [22]. In this study an Agile way of working was introduced into projects that were using the stage-gate system. A case study was done on three companies and improvements were seen in multiple areas. With regards to planning and prioritization, improvements seen include getting earlier feedback on features and avoided fixed plans. When looking at the area of communication and follow-up, the internal communication became better, the delivered code was of higher quality and the team members felt more in control. Lastly in the areas of process models and roles as well as project management, improvements were seen with regards to the continuous feedback, delivering relevant features, members feeling motivated as well as better priorities between code and documentation [22].

The study mentions a number of positive effects which came from using Agile within

a stage-gate model, however it also came with some problems [22]. If we start by looking at the area of planning and prioritization, the study found that there was a lack of support for long-term plans when switching to Agile. When looking at communication and follow-up, there were problems with the isolation of the Agile team from the organization. In the area of process models and roles, a problem of conflict between different amounts of documentation arose. Lastly in the area of project management it was found that managers were initially afraid of the transition and required training. There was also a problem with technical questions being raised too early for management. The article by Karlström and Runeson show that while Agile can bring a number of improvements, it is not flawless [22]. The article however does reveal that there are more positive impacts than negative as a result of introducing Agile which should make it worth implementing.

2.2 Team Dynamics

When working in an Agile way the work is almost always conducted in some form of a team constellation. Therefore, the team dynamics can be seen as a critical part of success, but not the only one, for an Agile team. In a study by Gren et al. they investigate how the maturity of a team is connected with being Agile [9]. In the study they compare Agile teams to high-performance teams according to the Group Development Questionnaire (GDQ) presented by Wheelan and Hochberger [23]. The study found many similarities between how Agile teams are described by practitioners and how high performance teams are described by psychologists. It is also revealed that a team that is more mature, as defined by psychology, is also more Agile. It was found that the highest correlation between agility and group maturity was with regards to dedication to teamwork and results, open communication and Agile planning. It is found that the interviewees identified solving group issues as key success factors when building teams. The importance of psychological skills when building teams is however not evident enough in the study. This study's focus is mostly related to building teams that will work well from a psychological standpoint, whereas this study will look at what makes a team successful from an Agile point of view.

In the study by Gren et al., the interviews found interesting results [9]. The interviewees mentioned team spirit as a main reason for the increase in satisfaction when working in a team. It was also stated how team spirit could be helped by having an Agile approach. The interviewees also brought up how both the team working skill and the individual skills were important as well as how having a collocated team certainly helps team development. It is mentioned how Agile project management can't solve all problems, but building high performance teams is a big part. The interviewees also mention some problems in the early stages of the team, when the team is newly formed, where task volunteering was the main problem brought up.

2.3 Studies of success factors

The related work of this study is mostly focused on what values and factors that have been previously identified as important for a company and its employees. The literature review found a number of values and factors identified by multiple sources that will be used for comparison with the results of this study.

According to Alahyari et al., the values identified as most important for the participants of the study performed were delivery process with regards to time, perceived quality and cost [7]. The participants came from different industries and got the opportunity to state what they value most in their organizations.

In a survey conducted by Chow and Cao [1] they identify critical success factors in Agile software development projects, three factors were identified as truly critical while another three were deemed critical in some projects. The factors that were truly critical were: Delivery strategy, Agile software engineering techniques and Team capability. The three factors that were deemed critical in some projects were: Project management process, Team environment and Customer involvement. These factors were derived from multiple success factors that were more narrow. The survey was conducted on 109 Agile projects from a diverse group of organizations of various sizes. The study involved 48 research hypotheses and ten of these were supported by the study. The results of this study helped with the identification of previously studied success factors since the factors identified were used when analyzing the data from survey one.

In a similar study by Misra et al., the success of an Agile software development project is defined by these criteria: reduced delivery schedules, increased return of investment, increased ability to meet with the current customer requirements, increased flexibility to meet with the changing customer requirements and improved business processes [3]. The factors that had a significant relation to the success of an Agile software development project were found to be customer satisfaction, customer collaboration, customer commitment, decision time, corporate culture, control, personal characteristics, societal culture and training and learning.

In a case study performed by Castka et al.[8] on the factors affecting successful implementation of high performance teams the results indicate that some of the human factors affecting the success are: defined focus, need of the individual, group culture and the existence of measures of performance.

2.4 Agile Success Factor Categories

The previous sections focused mostly on which factors that had been previously identified as important for a company and its employees. The researchers aimed to find similar values and factors identified by multiple sources and use this to compare with the results of the research. It was also sought to identify what different factors

are important for different levels of a company.

The different success factors identified can be categorized into five different categories.

- Organizational
- People
- Process
- Technical
- Project

These categories are stated by Chow and Cao and it is these as well as the success factors identified by them that will be used when creating the first survey [1].

2.4.1 Success Factors

In the study performed by Chow and Cao they identified 36 success factors that they later refined into 12 which is used in their hypotheses testing [1]. These 36 success factors are divided into the categories discussed previously and listed below in table 2.1. These are the same factors that were used in survey two combined with the factors gathered from survey one.

Chow and Cao also performed a reliability analysis in order to reduce the number of high level factors they had into 12. These were later translated into hypotheses for testing. The factors identified are listed below in their respective category.

- **Organizational**
 - Management Commitment
 - Organizational Environment
 - Team Environment
- **People**
 - Team Capability
 - Customer Involvement
- **Process**
 - Project Management Process
 - Project Definition Process
- **Technical**
 - Agile Software Techniques
 - Delivery Strategy
- **Project**
 - Project Nature
 - Project Type
 - Project Schedule

2.5 Success dimensions

Another aspect discussed by Chow and Cao [1] are success dimensions. They identify the success dimensions Quality, Scope, Time and Cost. Quality is defined as delivering a good product or a good project outcome, scope is defined as meeting all requirements and objectives, time is defined as delivering on time, cost is defined

Table 2.1: Success factors from literature [1]

<i>Dimension</i>	<i>Factors</i>
<i>Organizational</i>	<ol style="list-style-type: none"> 1. Strong executive support 2. Committed sponsor or manager 3. Cooperative organizational culture instead of hierarchical 4. Oral culture placing high value on face-to-face communication 5. Organizations where Agile methodology is universally accepted 6. Collocation of the whole team 7. Facility with proper Agile-style work environment 8. Reward system appropriate for Agile
<i>People</i>	<ol style="list-style-type: none"> 9. Team members with high competence and expertise 10. Team members with great motivation 11. Managers knowledgeable in Agile process 12. Managers who have light-touch or adaptive management style 13. Coherent, self-organizing teamwork 14. Good customer relationship
<i>Process</i>	<ol style="list-style-type: none"> 15. Following Agile-oriented requirement management process 16. Following Agile-oriented project management process 17. Following Agile-oriented configuration management process 18. Strong communication focus with daily face-to-face meetings 19. Honoring regular working schedule – no overtime 20. Strong customer commitment and presence 21. Customer having full authority
<i>Technical</i>	<ol style="list-style-type: none"> 22. Well-defined coding standards up front 23. Pursuing simple design 24. Rigorous refactoring activities 25. Right amount of documentation 26. Regular delivery of software 27. Delivering most important features first 28. Correct integration testing 29. Appropriate technical training to team
<i>Project</i>	<ol style="list-style-type: none"> 30. Project nature being non-life-critical 31. Project type being of variable scope with emergent requirement 32. Projects with dynamic, accelerated schedule 33. Projects with small team 34. Projects with no multiple independent teams 35. Projects with up-front cost evaluation done 36. Projects with up-front risk analysis done

as delivering within estimated cost and effort [1]. The 12 factors stated previously were tested with regards to these dimensions and listed below are the ones that were deemed to have an impact on the dimension.

- **Quality**
 - Team environment
 - Project management process
 - Agile software techniques
- **Scope**
 - Customer involvement
 - Agile software techniques
 - Delivery strategy
- **Time**
 - Team capability
 - Delivery strategy
- **Cost**
 - Team capability
 - Delivery strategy

If we look at the categories of the factors that were deemed to have an impact, it can be confirmed that Chow and Cao found no factors that affected the success dimensions at all from the Project category. The categories that can be deemed to be most important are Technical and People where both have factors that affect multiple success dimensions [1]. Therefore we will study success factors of teams and not projects.

2.6 Agile Metrics

There are different metrics and measurements used within Agile, some examples can be measuring conformance to Agile practices, value delivered and velocity (how fast you deliver). In a paper by Matthies et al. the authors describe how to measure conformance to Agile practices [24]. The paper presents a list of conformance metrics and lists their severity, effort, a short summary as well as where it can present itself. These metrics are then used as help in a course on Agile practices in order to find violations for the teams. The study promotes this way of measuring since it relies on a data-driven approach and thus avoids bias.

In a study performed by Kupiainen et al. they discuss metrics and measurements on how well Agile performs [25]. The paper studies previous research in order to find metrics used in Agile and compiles a list of these as well as looking at their effectiveness. Examples of metrics provided are: delivered business value, velocity, technical debt and critical defects sent by customer. The paper defines five different categories that the metrics can be used for, it is found that some metrics can be used for several categories. These categories relate to planning, progress, quality, fixing problems and motivation. It is also found that some metrics can be used for all these categories, for example metrics of defects. In total 30 different metrics are

found and investigated how they can be used with regards to these categories.

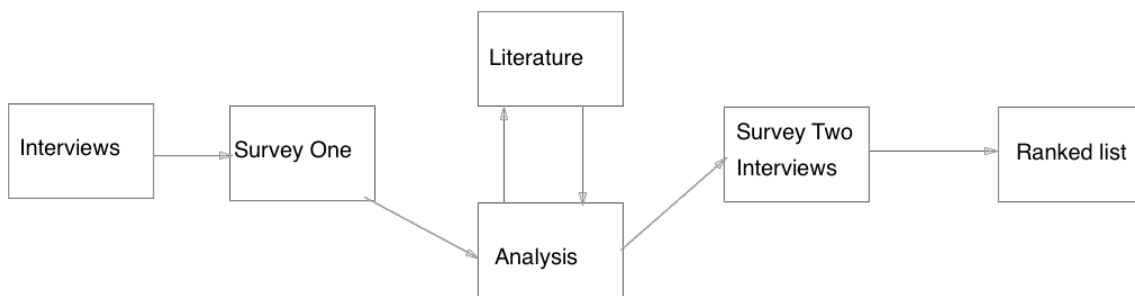
Although this research focuses on measures for Agile team success and these focus on measurements and metrics regarding Agile performance and a team's conformance to the Agile practices some might apply. However, these metrics and measurements are not necessarily the same. We do however note that there is a difference between a metric and a measurement, although relevant sources disagree on the exact definitions [26], [27]. Although this study has asked about measurements, because of the similarity between the terms, there is still some merit to comparing these measures to metrics.

3

Methods

This section presents a description of the methods used to gather data. The research is comprised of two surveys as well as interviews, the description for each and the reasoning behind their application are stated in this chapter. It also presents how the data was analyzed after collection. The research process is visualized, from performing the first interview to compiling the ranked list, below in Figure 3.1.

Figure 3.1: The research process



3.1 Surveys

In order to gather the necessary data to answer the research questions two surveys were conducted. The total population for the surveys were employees that work in Agile teams at the telecom service company. The selection for the surveys was made based on stratified random sampling [28]. This means that the employees were divided into different groups and only specific groups were selected for the sample. It was decided that the most interesting groups to survey was the employees that work in Agile teams that develop code. This is since the study looks at Agile teams in a software engineering context. The sample, as mentioned in the introduction, was based on the consultants that are working for Knowit at the service company and the employees at the service company. Once this distinction was made the sample consisted of 300 people. The employees that were surveyed consisted of people from different disciplines within software engineering, such as developer, UX designer and architect, which gave a broad span of opinions based on the tasks performed in the teams.

The data was collected using online surveys, where links were sent via email that the participants could answer. This link was sent out by an Agile coach at the service company in order to get a higher response rate since the survey would feel more

important if it came from their superiors. Since the surveys were performed online and no personal information was provided by the respondents that could identify them, we could ensure total anonymity of the answers. This was an important credential since the respondents had the possibility to share sensitive information in their responses which they put faith in that no one could trace back to the individual who wrote it. This was crucial since we had to eliminate the risk that the respondents manager or superior would read the responses later on and find out who wrote a certain piece of information [28]. The online surveys were in the form of questionnaires with both closed and open-ended questions. These closed questions dealt with both the participants background such as what types of Agile practices they use as well as information about their role in the company. The open-ended questions were there so that the respondents could suggest success factors. The data was collected at one point in time as is the case with cross-sectional surveys [28], this was due to the fact that there was no need to look at the answers over time.

Since two surveys were performed it is important to highlight the difference between these. The data to be collected were different between the first and second survey as they served different purposes. Both surveys were tested before sending them out to the sample. This was done to identify missing or ambiguous questions [28] in order to get as many valid answers as possible and avoid missing or erroneous data. In the following sections each of the two surveys are described in more detail.

3.1.1 The first survey

The first survey conducted was open-ended, asking questions in order to come up with success factors and measurements. This was done in order to see what the respondents could come up with without any input. This input could be for example showing previously found success factors or giving concrete examples of what a success factor could be. The questions regarding their background information were closed with fixed alternatives for them to pick from. There was also a section asking the respondents about what Agile practices were used in their team. This was done in order to understand the agility of the company. This is in line with the opinions of Kurapati et al. which advocates asking about practices used when evaluating the agility [29]. The survey was used to collect the data necessary in order to create the second survey. The data was later processed into a list of success factors and measurements for these factors which can be seen in Section 3.3. The survey can be viewed in its entirety in Appendix B.

3.1.1.1 Pilot test

For the first survey a pilot test was conducted in order to remove uncertainties within the survey. The survey was sent out to five people in the sample to have them test it. Based on their feedback the survey was altered to make it easier to complete and understand. As part of the pilot an interview with Agile coaches at the company was performed. The interview pilot was done in order to narrow down the list of Agile practices that were going to be asked in the survey. The feedback received was regarding the time it took to complete the survey as well as comments

about the length and design of the Agile principles list. Most felt that the list was too long, unfortunately the list had to be at that length since all of those practices were marked as used in the interview performed with the Agile coaches. Therefore it is necessary to ask about all of them in order to see how Agile they are with regards to practices they use.

3.1.2 The second survey

The second survey contained the same first section as the first survey which consisted of background information about the respondent, such as team name and experience. This part was included in the second survey as well in order to know the experience of the respondents. The second survey was different from the first survey since it consisted of closed questions rather than open ended questions. This survey focused on rating the importance of the success factors found from processing the results of the first survey that were merged with the success factors derived from literature [1]. The success factors from the literature that were used as input in survey two had two slight modifications which are explained in the analysis of the results of survey one, described in Section 3.3.2.1.

Participants of the second survey were asked to rate the importance of the proposed success factors on a one to five likert scale where one represents "not at all important", three is "neutral" and five is "very important". The rating of importance was then used to create a ranked list of success factors, ranked from most important to least important. Rating the factors is deemed an important step since this gave the respondents an opportunity to rate factors that they might not have thought of earlier, but may be relevant to the working scenario. When constructing the ranking of success factors from these results the average of the answers given was used. Although Sullivan and Artino argue that likert scales should not be averaged, since they are ordinal and not nominal [30], this was done anyway in order to get a meaningful ranking of the success factors. Although this could prove to be an issue it was necessary to do in order to get a rough estimate of importance and thus a form of ranking between the factors.

The last part of the second survey consisted of the measurements provided by the respondents of survey one presented under their corresponding success factors' category that was provided through iterative categorization [31] which is described further in section 3.3.1. This approach was chosen in order to be able to shorten down the length of the survey to some extent as keeping the measurements connected to the success factors would complicate the survey and make it too long in our opinion. As a result of iterative categorization the amount of measurements could be reduced from from 151 to 68. However the pilot test revealed that the survey was too long and thus the measurements section of the survey was dropped and only used for the interviews. Finally, the second survey provided an option for the respondents to write a piece of text explaining their decisions, one overall for each category. The survey can be viewed in its entirety in Appendix C.

3.1.2.1 Pilot test

The pilot test for the second survey was performed in the same way as the pilot test for the first survey and was performed to remove ambiguity and missing questions [28]. The survey was distributed to five people that answered the survey and provided feedback. Even during the pilot, the second survey received significantly less interest than the first survey since only three of the five people it was sent to responded. The results of the pilot test showed that the survey was too long, which was anticipated since the list of measurements was introduced. Three people responded to the pilot test of the second survey and they gave similar feedback that the whole survey was too long and that the measurements part was confusing. The success factor section and measurements section was also filled in at the four interviews performed with developers and scrum masters. During these interviews, more information and thoughts about the presentation of the success factors and measurements could be retrieved. Due to all the feedback received about the survey from the pilot test and the interviews, the measurements section of the survey was dropped before sending it out to the sample.

3.2 Interviews

Here a description of the different interviews performed during the study will be discussed.

3.2.1 Agile coaches

To gather more information about the Agile practices and methods that are relevant to this study, an initial interview was carried out with two Agile coaches from the company. This interview provided a thorough background on how the teams utilize Agile practices today and was helpful in narrowing down the list of Agile principles that had to be investigated in the following interviews and surveys. This interview also helped in testing the interview questions that were going to be asked in the following interviews. This interview was performed before sending out the first survey and as a result, weaknesses in the questions could be found while ambiguous and misinterpreted questions could be rephrased or removed.

3.2.2 Developers & Scrum Masters

The interviews were originally planned to be performed with managers but due to time constraints and communication issues with the company this was not possible. Therefore, as a complement to the first and second survey, interviews were performed with two developers and two scrum masters instead. These interviews allowed us to get more in-depth information about the success factors and measurements that had been gathered from the first survey. It also provided a deeper insight in how the organization was functioning from an inside perspective, and how the company's Agile transformation was perceived. The interview script for these interviews can be seen in appendix A.

3.3 Analysis

This section covers how the data from the surveys and interviews was analysed, along with some of the techniques used.

3.3.1 Iterative Categorization

Iterative Categorization, hereafter referred to as IC, is a way to process qualitative data. According to Neale, the value of IC is that it uses standardized procedures to perform the analysis which leaves a clear trail of how the processing was made [31]. This implies that it is possible for anyone to understand how the categorizations were made and follow the thought process behind analysing the findings of the success factors as it provides a way to trace back the steps.

3.3.2 Surveys

Once a basic understanding of the data had been gained, the interpretation started which consisted of looking for patterns and categories in the data that can be related to each other and a broader body of knowledge. In the first iteration, the success factors from the first survey were mapped to rough categories in which they could belong with regards to the domain of the success factor and its proposed measure. As an example, a success factor "low external downtime" is mapped to a rough category "external downtime" as it is an external factor that can influence the success of a team in this scenario. The rough categorization was also made to provide an early and quick overview of how the answers were distributed across several categories. This process was repeated three times in close collaboration between the researchers to ensure that the success factors were mapped in the most accurate way possible and that there would be no ambiguity when re-tracing the steps later on. Some of the success factors were vaguely defined but had a distinctive measurement proposed that described the thought process behind the answer, in scenarios like these it was of great importance to be able to retrace the process.

Each of these categories had success factors which were related to the area of the category according to the survey response. The identified success factors from the survey were then compared to success factors identified in the study by Chow and Cao [1] which includes the categories Organizational, People, Process, Technical and Project and had in total 36 success factors associated with it.

To compare found success factors to success factors from the literature, we mapped the success factors from the survey to the success factors identified by the literature and merged any factors that represented the same concepts. This was done to compile an optimized list of success factors that could be used for the second survey where the resulting success factors were to be ranked by importance by the respondents. The merging was necessary as it would influence the second survey and the final result if there were two or more success factors that essentially were the same

but had different names. It was thus necessary in order to reduce the amount of success factors presented back as we had the intention to construct a concentrated list of factors.

During this step of analysis, the mapping was made from success factors within a category shown in Table 4.9, to a success factor within a category defined by literature. It is important to differentiate between category and success factor in this step as a category has a number of success factors. If a success factor from the survey responses could not be mapped to any success factor from the literature, it was placed under a category from the literature, if it was deemed to fit there.

As an example, a success factor A from the survey responses might be mapped to success factor B from the literature, that is contained in category C. If there was no mapping possible between A and B, A was placed as a success factor contained in C if it was deemed to fit there.

If the success factor did not fit in any of the categories from the literature, it remained in its current category which was then added to the final list of categories. As a result of the processing, the only category from the survey that was appended to the final list was "Team spirit". All other success factors were successfully mapped to existing success factors or placed in an existing category. By appending the new category, it made sure that no success factors were lost by trying to force them into categories. The resulting list of success factors can be seen in the results Section 4.2.3.

When processing the results of survey two, we noticed that there was a single success factor that had already been mapped into another success factor, but was still present in its original format in the second survey. As a result, any responses regarding the success factor "Team affect product" will be ignored since it was mapped into the success factor "Coherent self organizing teamwork" which, in turn, was rated and processed in the second survey.

3.3.2.1 Initial success factors

The list of success factors provided by [1] was modified before using in the second survey. Two factors, 9 and 28, were modified for a slightly more flexible interpretation. There were a number of factors regarding diverse teams retrieved from the first survey and in order to not create too similar factors, such as a factor saying to have "Teams with both experienced developers and new", the existing factor from literature was altered to reflect this. The same applies with the factor regarding testing, the responses from the survey talk about having testable code so, instead of creating a new factor that stated "Testability of the code" this was altered to reflect this and avoid duplicates.

Table 3.1: Modifications to identified success factors from literature

<i>Original</i>	<i>Modified</i>
Team members with high competence and expertise	Team members with high competence and diverse expertise
Correct integration testing	High testability of codebase

3.3.3 Interviews

The data from the interviews with developers and scrum masters consisted of two parts: the answers to the second survey where they rated success factors and gave their opinion on measurements, and their verbal answers to the interview questions.

The answers regarding the second survey was collected on a paper and later on processed in a similar fashion to the survey responses so that these results could be included in the final list of rated success factors. However, the benefit of interviews was utilized and more information was gathered verbally regarding how they chose to rate the factors.

The interviews were recorded with the permission of the interviewee and notes were taken during the interview. When processing the interviews, the recordings were listened to again and cross checked against the notes to identify any potential information that was left out during the initial note taking.

4

Results

This section presents the results from the interview with the Agile coaches, the two surveys and the interviews conducted with the developers and scrum masters. The results are presented in descending order based on time, starting with the interview with Agile coaches and the first survey, followed by the second survey and ending with the final interviews.

4.1 Interview with Agile coaches

The results from the interview revealed that the Agile coaches didn't really monitor which teams were in need of support, it was rather the teams responsibility to ask for help when they needed it. From the original list of Agile practices, with 60 entries at the time of the interview, 50 of the practices were used in the organization. The Agile practices removed can be seen in Table 4.1.

Table 4.1: Agile practices that were **not** used

<i>Number</i>	<i>Agile Practice</i>
<i>1</i>	ATDD
<i>2</i>	Given-When-Then
<i>3</i>	Heartbeat retrospective
<i>4</i>	Project Chartering
<i>5</i>	Role-Feature-Reason
<i>6</i>	Rules of simplicity
<i>7</i>	Three C's
<i>8</i>	Three Questions
<i>9</i>	Three Amigos
<i>10</i>	Ubiquitous language

4.2 Survey One

This section starts by presenting the background of the respondents followed by the degree of agility of the organization. Next, the categories and success factors identified as a result of the first survey are presented. The total number of respondents for this survey was 53. Out of these responses, five provided invalid feedback regarding the success factors and had to be removed when examining success factors.

However, these responses were still taken into account when analyzing the agility and maturity of the organization as a whole. Invalid feedback could be, for example, if someone had just put a dot or a dash and had not answered the question.

4.2.1 Background Information of Respondents

The gathered data from the first survey identifies that the experience of the survey respondents in the organization is rather high, where 74% of the respondents had more than five years of experience in developing software. The results also show that 66% of the respondents have known about Agile practices for more than five years and in total, 90% have known about Agile for 3 years or more. The findings also indicate that 75% of the respondents have been developing software using Agile techniques for more than three years and 60% of these have more than five years of experience with Agile software development. This background information conveys the story that the majority of the employees in the Agile teams of the company have a lot of experience with both software development and Agile in general. In contrast, only three people had less than one year of experience with software development and only one person had been knowledgeable about Agile practices for less than one year. The distribution of the data can be seen in table 4.2.

Table 4.2: Experience of respondents from Survey one

<i>Years</i>	<i><1 years</i>	<i>1-2 years</i>	<i>3-4 years</i>	<i>5+ years</i>	Total
<i>I have developed software for:</i>	3	4	7	39	53
<i>I have known about Agile practices for:</i>	1	4	13	35	53
<i>I have developed software using Agile practices for:</i>	5	8	16	24	53

From the data it is evident that most of our respondents identify themselves as developers and only a small part of them identify as either product owners or managers. A portion of the respondents identify themselves as something else and the answers to these include roles such as: Scrum master, UX designer, architect or a role regarding testing. This is in line with how the sample was envisioned, people working at the telecom service company in an Agile way connected to development. The distribution of this data can be seen in table 4.3.

Table 4.3: Role of respondents from Survey one

<i>Role</i>	<i>Responses</i>
<i>Developer</i>	35
<i>Product Owner</i>	4
<i>Project Manager</i>	1
<i>Other</i>	13
Total	53

The results show that most of the teams are rather large with more than half of the respondents stating that they are eight or more people with more than a fourth stating that they are ten or more in their team. It was surprising to see that so many of the teams are ten or more people since the Agile alliance recommends teams to be a maximum of ten people [32]. The distribution of the data can be seen in table 4.4.

Table 4.4: Number of people in team from Survey one

<i>People in team</i>	<i>Responses</i>
4	4
5	7
6	6
7	8
8	10
9	4
10+	14
Total	53

From the responses it is also clear that the respondents feel that the products they work on have a higher than average complexity with almost half of them stating that they feel that their product has a high or very high complexity. The distribution can be seen in table 4.5.

Table 4.5: Complexity of product from Survey one

<i>Complexity</i>	<i>Responses</i>
<i>Average Complexity</i>	13
<i>More than average Complexity</i>	16
<i>High Complexity</i>	20
<i>Very high Complexity</i>	4
Total	53

4.2.2 Agility of organization

This section presents the results of how Agile the organization appears based on the data gathered from the first survey.

4. Results

Table 4.6: What Agile practices are used (results of survey one), the number presented is **Number of respondents**

<i>Practice</i>	<i>We don't use this</i>	<i>We us it rarely</i>	<i>We us it sometimes</i>	<i>We us it always</i>	<i>I am not familiar with the term</i>	Total
<i>Acceptance Testing</i>	5	6	19	23	0	53
<i>Automated Build</i>	1	1	3	46	2	53
<i>Backlog</i>	0	0	2	51	0	53
<i>Backlog Refinement</i>	0	1	6	46	0	53
<i>BDD</i>	16	15	8	7	7	53
<i>Burndown Chart</i>	22	9	6	13	3	53
<i>Collective Ownership</i>	5	6	9	29	4	53
<i>Continuous Deployment</i>	11	10	21	11	0	53
<i>Continuous Integration</i>	6	2	25	18	2	53
<i>Daily Meeting</i>	1	0	0	52	0	53
<i>Definition Of Done</i>	7	12	19	15	0	53
<i>Definition Of Ready</i>	16	12	10	11	4	53
<i>Design Studio</i>	19	5	4	1	24	53
<i>Estimation</i>	6	9	13	25	0	53
<i>Exploratory Testing</i>	8	15	12	8	10	53
<i>Facilitation</i>	8	7	11	8	19	53
<i>Frequent Releases</i>	5	4	18	26	0	53
<i>Incremental Development</i>	2	2	16	24	9	53
<i>Information Radiators</i>	7	4	4	4	34	53
<i>Integration</i>	2	4	10	29	8	53
<i>Invest</i>	10	1	2	6	34	53
<i>Iteration</i>	2	3	12	32	4	53
<i>Iterative Development</i>	1	3	14	33	2	53
<i>Kanban Board</i>	19	10	8	15	1	53
<i>Lead Time</i>	18	11	5	7	12	53
<i>Milestone Retrospective</i>	23	9	5	11	5	53
<i>Mob Programming</i>	24	13	4	1	11	53
<i>Mock Objects</i>	2	5	23	21	2	53
<i>Pair Programming</i>	3	18	24	7	1	53
<i>Personas</i>	22	18	6	3	4	53
<i>Planning Poker</i>	16	8	7	18	4	53
<i>Points (Estimates In)</i>	11	5	7	28	2	53
<i>Refactoring</i>	0	7	25	19	2	53
<i>Relative Estimation</i>	14	10	3	9	17	53
<i>Scrum Of Scrums</i>	3	5	18	25	2	53
<i>Sign Up For Tasks</i>	6	2	10	19	16	53
<i>Simple Design</i>	7	6	10	5	25	53
<i>Story Mapping</i>	8	15	16	4	10	53
<i>Story Splitting</i>	4	11	25	7	6	53
<i>Sustainable Pace</i>	10	7	8	6	22	53
<i>Task Board</i>	7	1	9	30	6	53
<i>TDD</i>	8	17	20	3	5	53
<i>Team</i>	1	0	1	50	1	53
<i>Team Room</i>	20	6	4	9	14	53
<i>Timebox</i>	3	12	19	9	10	53
<i>Unit Testing</i>	0	3	11	39	0	53
<i>Usability Testing</i>	7	20	15	9	2	53
<i>User Stories</i>	2	7	8	36	0	53
<i>Velocity</i>	21	8	9	11	4	53
<i>Version Control</i>	0	2	5	45	1	53

From these results we can derive two more tables to show which of the Agile practices are used the most. Table 4.7 show practices that more than 50% of the respondents claim to use always. As a complement, Table 4.8 show practices that more than 50% claim to use either always or sometimes. In Table 4.8 the practices from Table 4.7 are excluded.

Table 4.7: Agile practices where more than 50% of respondents answered that they always use it

<i>Practice</i>	<i>Percentage Use it always</i>
<i>Daily Meeting</i>	98%
<i>Backlog</i>	96%
<i>Team</i>	94%
<i>Automated Build</i>	87%
<i>Backlog Refinement</i>	87%
<i>Version Control</i>	85%
<i>Unit Testing</i>	74%
<i>User Stories</i>	68%
<i>Iterative Development</i>	62%
<i>Iteration</i>	60%
<i>Task Board</i>	57%
<i>Collective Ownership</i>	55%
<i>Integration</i>	55%
<i>Points (Estimates In)</i>	53%

Table 4.8: Agile practices where more than 50% of respondents answered they use it sometimes or more often

<i>Practice</i>	<i>Use it sometimes or always</i>
<i>Frequent Releases</i>	83%
<i>Mock Objects</i>	83%
<i>Refactoring</i>	83%
<i>Scrum Of Scrums</i>	81%
<i>Continuous Integration</i>	81%
<i>Acceptance testing</i>	79%
<i>Incremental Development</i>	75%
<i>Estimation</i>	72%
<i>Definition Of Done</i>	64%
<i>Continuous Deployment</i>	60%
<i>Story Splitting</i>	60%
<i>Pair Programming</i>	58%
<i>Sign Up For Tasks</i>	55%
<i>Timebox</i>	53%

4. Results

4.2.3 Initial success factors

The first survey consisted of 243 valid data points which represented success factors suggested by the respondents. After the initial processing of the survey responses according to IC, an initial list of categories were compiled as listed in table 4.9.

Table 4.9: Initial list of categories. The description explains what kind of factors were placed in that category

<i>Number</i>	<i>Category</i>	<i>Description</i>
1	Amount of buffer	Factors related to slack, unscheduled time to try new things
2	Autonomous	Factors related to autonomous teams
3	Backlog priority	Factors related to prioritization of backlog items
4	Collaboration (Team)	Factors related to collaboration within and between teams
5	Communication	Factors related to verbal and written communication within and between teams
6	Customer collaboration	Factors related to customer collaboration
7	Customer satisfaction	Factors related to customer satisfaction
8	Deadlines	Factors related deadlines, e.g flexible deadlines
9	Do what needs to be done	Factors related to team members doing what is needed to progress, even if it's not their area of expertise
10	Documentation	Factors related to documentation, e.g less documentation
11	Environmental	Factors related to work environment, such as desk, office condition
12	External dependencies	Factors related to dependencies on outside resources, e.g frameworks
13	External downtime	Factors related to downtime that isn't caused by the team, eg server downtime, waiting for responses
14	Good architecture	Factors related to software architecture
15	Incremental development	Factors related to incremental changes
16	Iteration	Factors related to iterative processes
17	Learning	Factors related to personal development
18	Low maintenance	Factors related to maintainability of software
19	Openness	Factors related to openness and transparency in the team and organization
20	Organizational	Factors related to organizational culture and processes
21	People	Factors related to people
22	Planning	Factors related to planning and scheduling
23	Proper tools	Factors related to hardware and software tools
24	Quality	Factors related to software quality and quality assurance
25	Refined backlog	Factors related to backlog refinement and making the backlog better
26	Story refinement	Factors related to refining user stories and well defined stories
27	Task focus	Factors related to focused work, less multitasking
28	Team affect product	Factors related to the team having the rights to make decisions about the product
29	Team competence	Factors related to competence, technical skills
30	Team spirit	Factors related to team spirit, trust and a good atmosphere within the team
31	Testing	Factors related to software testing and techniques

Table 4.10 explains how the success factors identified in the first survey were mapped to existing success factors identified by literature. The number "Found" identifies how many success factors from the survey results could be mapped to that success factor from the literature.

Table 4.11 explains how the success factors from survey one, that were not mapped to an existing success factor from the literature, were placed in an existing category from the literature. The table also contains the success factors, and the new category, team spirit, that were not previously identified by literature.

4.2.4 Measurements found

Beyond the success factors found, the measurements found in the first survey were also compiled into a list. Each measurement only appeared once in the survey, there

Table 4.10: Success factors from literature (Including the ones modified)

<i>Dimension</i>	<i>Factors</i>	<i>Found</i>
<i>Organizational</i>	1. Strong executive support	3
	2. Committed sponsor or manager	0
	3. Cooperative organizational culture instead of hierarchal	3
	4. Oral culture placing high value on face-to-face communication	7
	5. Organizations where Agile methodology is universally accepted	8
	6. Collocation of the whole team	2
	7. Facility with proper Agile-style work environment	6
	8. Reward system appropriate for Agile	0
<i>People</i>	9. Team members with high competence and diverse expertise	19
	10. Team members with great motivation	8
	11. Managers knowledgeable in Agile process	1
	12. Managers who have light-touch or adaptive management style	1
	13. Coherent, self-organizing teamwork	17
	14. Good customer relationship	5
<i>Process</i>	15. Following Agile-oriented requirement management process	4
	16. Following Agile-oriented project management process	0
	17. Following Agile-oriented configuration management process	0
	18. Strong communication focus with daily face-to-face meetings	4
	19. Honoring regular working schedule – no overtime	2
	20. Strong customer commitment and presence	3
	21. Customer having full authority	0
<i>Technical</i>	22. Well-defined coding standards up front	1
	23. Pursuing simple design	1
	24. Rigorous refactoring activities	1
	25. Right amount of documentation	1
	26. Regular delivery of software	5
	27. Delivering most important features first	5
	28. High testability of codebase	5
	29. Appropriate technical training to team	0
	<i>Project</i>	30. Project nature being non-life-critical
31. Project type being of variable scope with emergent requirement		5
32. Projects with dynamic, accelerated schedule		0
33. Projects with small team		1
34. Projects with no multiple independent teams		0
35. Projects with up-front cost evaluation done		0
36. Projects with up-front risk analysis done		0

4. Results

Table 4.11: Factors found not present in literature

<i>Dimension</i>	<i>Factors</i>	<i>Found</i>
<i>Organizational</i>	1. Transparent organization	6
	2. Share vision with the company	6
	3. Low volatility of team compositions	4
	4. Employees are willing to improve and get the chance to do so	8
<i>People</i>	5. Sharing knowledge within the organization	4
	6. Good communication and collaboration between different teams	7
	7. Good communication and collaboration within the team	17
	8. A trusting team environment	4
<i>Technical</i>	9. Well defined user stories	8
	10. Low external blocking factors	6
	11. Low external dependencies	3
	12. Proper development environment and tools	5
	13. Good architecture	3
	14. High quality code	2
<i>Project</i>	15. Projects with realistic planning	6
	16. Diverse projects	1
	17. Well focused sprints	5
<i>Team spirit</i>	18. Friendly and positive environment in team	11
	19. Mental well being among team members	6
	20. Prestigeless among teammembers	2
	21. Having fun at the workplace	4

were no repeated suggestions. The categories they are placed in relate to the success factor that was suggested alongside it and it is thus placed in the same category as that success factor, this can be seen in Table 4.12 and Table 4.13.

Table 4.12: Measurements derived from surveys

<i>Dimension</i>	<i>Measurements</i>
<i>Organizational</i>	<ol style="list-style-type: none"> 1. Amount of time per sprint blocked and average time length of blocking issues 2. By the amount of bad communication 3. Number of stories delivered in a sprint 4. Daily stand ups attendance 5. Improvements on velocity, number of improvements made 6. How many stay for a long time 7. By the ability of a team member to tackle tasks that are different in nature or complexity than the ones they were used to tackle earlier in their career. 8. The longevity of the team, how many days until the team is disbanded 9. How much each developer perceive that they have the management's support. 10. Time spent in predictably pointless meetings 11. N.o projects, on top of normal Agile workload 12. Spotify squad healthcheck 13. Man-hours spent by developers in meetings where their opinion/expertise is not solicited. 14. By the degree the work conducted matches a team member's ambitions. 15. NPS, NKI, sales targets, cost reduction. 16. By the degree a team member feels the work conducted and the working environment align with their personal views on what is right, fair or moral. 17. Average time the team members have been a part of the team
<i>People</i>	<ol style="list-style-type: none"> 18. % of important questions that may be decided by us 19. How often a developer has to prove (s)he is not faking the results. 20. By the amount of tasks finished with input by more than one person. Input does not necessarily have to be developing, but brainstorming, reviewing etc. 21. The number of jiras you solve each sprint. 22. Happiness 23. By the degree of satisfaction of each member with the help they receive from their other team members. 24. Number of completed Jira tasks 25. Conversion rate, bounce rate, phone calls to customer support, customer feedback online. 26. How many stories in progress 27. Amount of people asking for help. 28. Storypoints/velocity 29. Spotify squad healthcheck 30. Average throughput end-to-end, with analysis of bottlenecks. 31. NKI (Nöjd kundindex), feedback on pages, how often does team take part of user feedback 32. list competence and needed competence 33. By the number of different types of personalities and backgrounds of the team members. 34. How well and frequent does the stakeholder and developers in the team collaborate. And yes - the user is always a stakeholder. 35. The level of team-members' satisfaction of the Scrum Master's work. 36. Number of interactions with other teams?

4.3 Survey Two

This section will present the results from survey two. It will start by presenting the background information collected via the survey followed by presenting the ranked list compiled from the ratings. The total number of respondents for this survey was 27. All of the responses gathered could be used as none were deemed invalid.

4. Results

Table 4.13: Measurements derived from surveys cont.

<i>Dimension</i>	<i>Measurements</i>
<i>Process</i>	37. Quality of deliverable, n.o bugs 38. How many hours of work are put, how often do the team meet outside of work
<i>Technical</i>	39. Number of blocked tasks / amount of time spent blocked 40. Seconds that the servers is down. 41. Days unable to continue work 42. Comparing estimates to real world effort required. 43. Test coverage 44. How often do we make releases over time 45. Number of iterations on story due to changes or misunderstandings, amount of time spent discussing instead of doing after the task is added to a sprint 46. How much we load the servers with CPU/Memory 47. Tracking time spent "on hold" or fixing non-story related issues. 48. Measure through the amount of time developers needs to discuss with the product owner. 49. How often automated builds show red 50. Tracking time spent doing maintenance work, and comparing that to time spent working on development stories, to see how much time is claimed maintaining the application. 51. hours of correcting problems from other team 52. By the throughput and customer satisfaction 53. How many delays are done in a sprint due to external parties.
<i>Project</i>	54. Enjoyment on workplace 55. By checking how many are working on same story/task 56. Number of refined sprints ready to put in sprint backlog 57. How well do we succeed in completing our sprint goals over a period of time/sprints 58. Number of stories added to or removed from sprint 59. Measured life expectancy of a feature (how long will it take before the stakeholder changes her mind and makes us do-over from scratch). (stakeholder could be the team itself in this context). 60. How often and how many tasks/stories are not completed each sprint.
<i>Team spirit</i>	61. Team health 62. Happy workers 63. Number of conflicts and discussions around workflow in the group. 64. By people staying 65. By the degree a team member feels "at home" while working in the team.

4.3.1 Background Information of Respondents

The data presented below was gathered in order to show the background and experience of the respondents. From table 4.14 it can be seen that the respondents are quite experienced with a majority of the respondents having developed software for more than five years and being knowledgeable in Agile practices for more than five years. We do however see that people are not as experienced with developing software using Agile practices were more than half have developed software for three or more years. From table 4.15 we can see that a clear majority of the respondents are developers. The roles stated under other were Kanban master and Agile coach. In table 4.16 it is shown that there is a wide spread of team sizes for this survey but the majority are at least more than 5 but less than 10. The complexity of the products the respondents work on can be seen in table 4.17.

Table 4.14: Experience of respondents from Survey two

<i>Years</i>	<i><1 years</i>	<i>1-2 years</i>	<i>3-4 years</i>	<i>5+ years</i>	Total
<i>I have developed software for:</i>	2	1	7	17	27
<i>I have known about Agile practices for:</i>	0	2	10	15	27
<i>I have developed software using Agile practices for:</i>	2	6	12	7	27

Table 4.15: Role of respondents from Survey two

<i>Role</i>	<i>Responses</i>
<i>Developer</i>	19
<i>Product Owner</i>	2
<i>Project Manager</i>	0
<i>Scrum Master</i>	2
<i>UX Designer</i>	2
<i>Other</i>	2
Total	27

4.3.2 Ranked List

The ranking of the factors identified can be seen in Table 4.18 and Table 4.19. Here the average of all responses for each factor can be seen. These tables represent the combination of responses from both survey two as well as the interviews performed with developers and scrum masters. The total number of responses, for interviews and surveys combined, was 31. Table 4.20 shows the ranking of the categories between each other. Their score is acquired by taking the average of the success factors score in that category. The ranked list derived from only survey two responses can be seen in Appendix D and the ranked list derived from the interviews can be seen in Appendix E.

Table 4.16: Number of people in team from Survey two

<i>People in team</i>	<i>Responses</i>
4	3
5	4
6	1
7	6
8	5
9	3
10+	5
Total	27

Table 4.17: Complexity of product from Survey two

<i>Complexity</i>	<i>Responses</i>
<i>Average Complexity</i>	8
<i>More than average Complexity</i>	6
<i>High Complexity</i>	12
<i>Very high Complexity</i>	1
Total	27

4.4 Interviews

This section will present the results from the interviews carried out. In total four interviews were carried out. Two of these were with scrum master and the other two were with developers.

4.4.1 Interviewee A

Interviewee A has the role of a Scrum master/Agile agent. There are 6 people in his team and he has been working there for 3 years. They don't have any specific way of evaluating his team today but they have some goals that his team should fulfill. They call this continuous improvement and they are supposed to improve in 5 different areas. Two of these are how well everything flows within the team as well as the teams health, this is followed up on retrospectives. Once in a while they hold workshops with other teams were they cooperate on how they can improve their performance.

Success for interviewee A in the context of a team is that everyone is on the same page and knows what should be done and when. He also feels that the team is successful if their customer is happy with the work they are doing. Success in the context of a project on the other hand is having clearly marked requirements and some way of checking how things are going, this doesn't however have to be a traditional requirements specification. It is also good to have a customer that knows what they want as well as having a product owner that gives them a clear way

Table 4.18: Ranked success factors, italic is newly found. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Factor</i>	<i>Average</i>
1(PPL)	<i>Good communication and collaboration within the team</i>	4.87
2(PPL)	<i>A trusting team environment</i>	4.80
3(TS)	<i>Friendly and positive environment in team</i>	4.68
4(O)	<i>Employees are willing to improve and get the chance to do so</i>	4.58
5(TS)	<i>Mental well being among team members</i>	4.53
6(T)	<i>Low external blocking factors</i>	4.4
7(PPL)	<i>Good communication and collaboration between different teams</i>	4.37
8(PPL)	Team members with great motivation	4.35
9(T)	<i>Well defined user stories</i>	4.28
10(TS)	<i>Prestigeless among team members</i>	4.27
11(PPL)	Team members with high competence and diverse expertise	4.23
12(T)	Delivering most important features first	4.22
13(T)	<i>Proper development environment and tools</i>	4.22
14(O)	<i>Low volatility of team compositions</i>	4.21
15(T)	High testability of codebase	4.17
16(PPL)	Coherent self organizing teamwork	4.16
17(T)	Pursuing simple design	4.15
18(O)	Organizations where Agile methodology is universally accepted	4.14
19(T)	<i>High quality code</i>	4.11
20(TS)	<i>Fun at work</i>	4.11
21(O)	Strong executive support	4.09
22(T)	Regular delivery of software	4.06
23(PRT)	<i>Well focused sprints</i>	4.03
24(PPL)	Managers knowledgeable in Agile process	4.01

Table 4.19: Ranked success factors cont., italic is newly found. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Factor</i>	<i>Average</i>
25(T)	<i>Good architecture</i>	3.97
26(PPL)	Managers who have light-touch or adaptive management style	3.93
27(T)	Appropriate technical training to team	3.93
28(O)	Cooperative organizational culture instead of hierarchical	3.88
29(T)	Right amount of documentation	3.85
30(T)	<i>Low external dependencies</i>	3.83
31(O)	Committed sponsor or manager	3.82
32(PRT)	<i>Projects with realistic planning</i>	3.77
33(O)	<i>Transparent organization</i>	3.75
34(PPL)	Good customer relationship	3.75
35(PPL)	<i>Sharing knowledge within the organization</i>	3.73
36(PRS)	Following Agile-oriented requirement management process	3.70
37(PRS)	Strong communication focus with daily face-to-face meetings	3.68
38(O)	<i>Share vision with the company</i>	3.61
39(PRS)	Strong customer commitment and presence	3.58
40(T)	Well-defined coding standards up front	3.53
41(T)	Rigorous refactoring activities	3.53
42(PRS)	Following Agile-oriented project management process	3.49
43(O)	Collocation of the whole team	3.44
44(PRT)	Projects with small team	3.33
45(PRS)	Following Agile-oriented configuration management process	3.25
46(PRS)	Honoring regular working schedule - no overtime	3.15
47(O)	Facility with proper Agile-style work environment	3.08
48(PRT)	Project nature being non-life-critical	3.07
49(PRT)	Project type being of variable scope with emergent requirements	2.91
50(PRT)	Projects with up-front risk analysis done	2.86
51(PRS)	Customer having full authority	2.84
52(O)	Oral culture placing high value on face-to-face communication	2.81
53(PRT)	<i>Diverse projects</i>	2.74
54(PRT)	Projects with dynamic, accelerated schedule	2.70
55(PRT)	Projects with no multiple independent teams	2.56
56(PRT)	Projects with up-front cost evaluation done	2.46
57(O)	Reward system appropriate for Agile	2.28

Table 4.20: Ranking of categories

<i>Rank</i>	<i>Factor</i>	<i>Average</i>
1	Team Spirit	4.40
2	People	4.22
3	Technical	4.02
4	Organizational	3.64
5	Process	3.38
6	Project	3.04

forward.

He thinks that the factor that mostly affects a teams success is the well being of the team and how well they work together. He thinks that it is important to let the team work out how they work best and not try to fit them into some predefined structure of how a team should be and work. He thinks that a way to measure this would be to do regular health checks on both an individual and team level.

4.4.2 Interviewee B

Interviewee B has the role of Developer at the service company. The team consists of 10 people, although he said that it can be reduced to 7 or 8 depending on if you count the tester and designer that are not as involved in their daily work. The interviewee has been working at the company for about a year, and is under the impression that Agile is working so-so at the company today. The interviewee was under the general impression that Agile may not always be a good way to work, especially since he mentioned that it doesn't always work that well in the company. This was a result of rapidly changing requirements, which suits Agile, with hard deadlines that doesn't suit Agile. The team had retrospectives one to two times per month where they wrote down feedback on notes that contained opinions on what had worked well, and what could be improved.

The interviewee mentions that success in the context of a team is mainly to deliver value of some sort, not necessarily programming related. His example was if the customer comes up with a demand, and they convince the customer that doing so is a bad choice, it should also be counted towards success. On a project level his impression is that success is that the project was conducted under budget and delivers as much as possible. He also mentioned that there were some communication issues with their remote manager and that the contact with the customer was pretty difficult to handle due to the customer representative having to run everything through their superiors.

He mentions that success factors for the teams success is how knowledgable you are about the work you do, as an example he mentions that it is clearly noticable since

they have a lot of new recruits coming in, and the the average team age is about one year. His point being that it is clear that people that have been working at the service company for a longer period is able to make more work happen with the time given. He also mentions that it is important how well defined the demands are from the beginning, and how far in the process these are changed.

As proposed measurements, his main point is that they are bad at writing acceptance criteria beforehand. He proposed that you should be able to measure how many of these have changed over time, as it would be very useful if you could access those measurements.

4.4.3 Interviewee C

Interviewee C has the role of Developer and his team consists of 7 people. He has been working there for 1 year and his team has no formal way of evaluation today except the retrospectives that are a part of the Agile way of working. During these retrospectives they go over what has been good and bad and discuss how to improve this. Success for him in the context of his team is that their user stories get done and get delivered to the customer within the time limit. For success in the context of a project he feels that success is when it is actually delivered. He also feels that it needs to be delivered with good quality. He also states that "If you don't deliver, it is like you have not done anything".

The factors that mostly affect success according to interviewee C is people being away, making good estimates and how much time that is spent with bureaucracy. He states that he can't think of any good way to measure this. He states that a way to measure success is how many story points you get done in a sprint but this is very subjective as well since these are just estimates and it is hard to know if they are good or not.

4.4.4 Interviewee D

Interviewee D is a developer and scrum master at the company. His team size varied from 7 to 9 people. He was scrum master for one team but there were two teams that shared the same product domain, which meant that they had the same product owner. The interviewees Agile experience was around 15 months in total, all of which he had gain in his position at the service company where he had been appointed scrum master after the previous scrum master left the team. In their retrospectives, they followed the same method as the previous scrum master had used. The process was that they went through each and every member who would then talk about how they felt that the last sprint has proceeded, what was good and what could be improved.

His definition of success within an Agile team is that the team reaches the goals that it set for that sprint and that everyone manages to complete their sprints in time. As for a definition of success for a project, the interviewee was not entirely sure but gave a suggestion of that everything is delivered according to some road

map in time.

The interviewee believes that success factors for an Agile team are strongly related to well defined goals as well as a healthy team environment where team members trust each other, have a good group coherence and where everyone has the proper competence needed for the job. The interviewee proposed no measurements as he felt that it wasn't performed within the company, although he rated external blocking factors as an important thing to measure. When asked to elaborate on the measurement, the interviewee described how external blocking factors played a major role in their daily work. He emphasizes on that communication is key and that communication was the most challenging problem at the company, especially with product owners sitting in a remote location.

4.4.5 Rating of Measurements

Here the ratings for the measurements elicited from the interviewees will be provided. The interviewees were asked to rate them with 1 to 3 where 1 was "No, this is not a good measurement", 2 is "Maybe, this measurement could be good" and 3 is "Yes, this is a good measurement". The closer to 1 the worse the measurement and the closer to 3 the better. The measurements were divided into the same categories as the success factors and the participants were asked to rate them in relation to these. The results of this process can be seen in Table 4.21 and Table 4.22.

4. Results

Table 4.21: Ranked measurements, TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Measurement</i>	<i>Average</i>
1(O)	Time spent in predictably pointless meetings	3
2(PPL)	Average throughput end-to-end, with analysis of bottlenecks.	3
3(PRS)	Quality of deliverable, n.o bugs	3
4(PPT)	How well do we succeed in completing our sprint goals over a period of time/sprints	3
5(TS)	Team health	3
6(T)	Days unable to continue work	3
7(O)	Improvements on velocity, number of improvements made	2.75
8(O)	Spotify squad healthcheck	2.75
9(O)	Man-hours spent by developers in meetings where their opinion/expertise is not solicited.	2.75
10(PPL)	By the degree of satisfaction of each member with the help they receive from their other team members.	2.75
11(T)	Number of blocked tasks / amount of time spent blocked	2.75
12(T)	How often do we make releases over time	2.75
13(T)	By the throughput and customer satisfaction	2.75
14(T)	How many delays are done in a sprint due to external parties.	2.75
15(O)	Amount of time per sprint blocked and average time length of blocking issues	2.5
16(PPL)	Conversion rate, bounce rate, phone calls to customer support, customer feedback online.	2.5
17(PPL)	Spotify squad healthcheck	2.5
18(PPL)	How well and frequent does the stakeholder and developers in the team collaborate. And yes - the user is always a stakeholder.	2.5
19(PPL)	The level of team-members' satisfaction of the Scrum Master's work.	2.5
20(PRT)	Enjoyment on workplace	2.5
21(PRT)	How often and how many tasks/stories are not completed each sprint.	2.5
22(TS)	Happy workers	2.5
23(T)	Comparing estimates to real world effort required.	2.5
24(T)	Number of iterations on story due to changes or misunderstandings, amount of time spent discussing instead of doing after the task is added to a sprint	2.5
25(T)	Tracking time spent "on hold" or fixing non-story related issues.	2.5
26(T)	Tracking time spent doing maintenance work, and comparing that to time spent working on development stories, to see how much time is claimed maintaining the application.	2.5
27(O)	How much each developer perceive that they have the management's support.	2.25
28(O)	NPS, NKI, sales targets, cost reduction.	2.25
29(PPL)	% of important questions that may be decided by us	2.25
30(PPL)	By the amount of tasks finished with input by more than one person. Input does not necessarily have to be developing, but brainstorming, reviewing etc.	2.25
31(PPL)	Happiness	2.25
32(PPL)	list competence and needed competence	2.25
33(PRT)	Number of stories added to or removed from sprint	2.25
34(TS)	By people staying	2.25
35(TS)	By the degree a team member feels "at home" while working in the team.	2.25
36(O)	By the amount of bad communication	2
37(O)	How many stay for a long time	2
38(O)	By the ability of a team member to tackle tasks that are different in nature or complexity than the ones they were used to tackle earlier in their career.	2
39(O)	The longevity of the team, how many days until the team is disbanded	2
40(O)	Average time the team members have been a part of the team	2
41(PPL)	How often a developer has to prove (s)he is not faking the results.	2
42(PPL)	Storypoints/velocity	2
43(PRS)	How many hours of work are put, how often do the team meet outside of work	2
44(PRT)	By checking how many are working on same story/task	2
45(PRT)	Measured life expectancy of a feature (how long will it take before the stakeholder changes her mind and makes us do-over from scratch). (stakeholder could be the team itself in this context).	2

Table 4.22: Ranked measurements cont., TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Measurement</i>	<i>Average</i>
46(TS)	Number of conflicts and discussions around workflow in the group.	2
47(T)	Test coverage	2
48(T)	How much we load the servers with CPU/Memory	2
49(T)	How often automated builds show red	2
50(O)	Number of stories delivered in a sprint	1.75
51(O)	Daily stand ups attendance	1.75
52(O)	N.o projects, on top of normal Agile workload	1.75
53(O)	By the degree the work conducted matches a team member's ambitions.	1.75
54(O)	By the degree a team member feels the work conducted and the working environment align with their personal views on what is right, fair or moral.	1.75
55(PPL)	How many stories in progress	1.75
56(PPL)	Amount of people asking for help.	1.75
57(PPL)	NKI (Nöjd kundindex), feedback on pages, how often does team take part of user feedback	1.75
58(PPL)	By the number of different types of personalities and backgrounds of the team members.	1.75
59(PRT)	Number of refined sprints ready to put in sprint backlog	1.75
60(T)	Seconds that the servers is down.	1.75
61(T)	hours of correcting problems from other team	1.75
62(PPL)	Number of interactions with other teams?	1.5
63(T)	Measure through the amount of time developers needs to discuss with the product owner.	1.5
64(PPL)	The number of jiras you solve each sprint.	1.25
65(PPL)	Number of completed Jira tasks	1.25

5

Discussion

In this chapter the results as well as the methodology used will be discussed in more detail. It will discuss the difference between the findings as well as answer the research questions posed in the introduction.

5.1 Surveys: Differences between findings

When comparing the first survey to the second survey we can see some differences between the results regarding the respondents background information. The first survey had 53 respondents and the second survey had 27 respondents. This means that the first survey got roughly 50% more answers than the second which in turn means that the first survey should, if we want approximately the same results, have double the amount of responses per section. If we compare the two according to this we can see that they differ slightly from each other. We can see that the first survey had a higher average of experienced people answering it than the second survey. It is also visible that the percentage of respondents who are developers answering the second survey is slightly higher than for the first survey. This could indicate that developers were more interested in the results from this study than the other groups surveyed. The distribution of the complexity for the products is somewhat similar where there is a higher percentage of respondents claiming to have an average complexity product in the second survey compared to the first where the first survey has a higher percentage claiming more than average complexity of their products. The distribution of these results can be seen in Section 4.2.1 and Section 4.3.1.

5.1.1 Team size

When examining the number of people that are part of the team the results vary quite a lot between the first and second survey. In the first survey there were 52% claiming to be 9 or more in the team whereas only 30% claimed this during the second survey. The large size of the teams are also interesting given that according to Wheelan, groups consisting of 8 or more people are less productive [33]. These differences were brought up during an interview in order to get an answer. It was discussed that it could be due to the difference in calculating the team size between the respondents. Some might be confused that the team size is everyone working on the same product and not just the team.

5.1.2 Overlap in Survey Respondents

From the points in the previous sections we can draw conclusions regarding the respondents of the surveys and if they are the same or not. Both surveys were sent to the same sample in order to get as similar people answering as possible. Based on the different distributions of the answers between the surveys we can not say that we have the exact same people responding to the second survey as the first survey. We do however see that even though the respondents of survey two are a little less experienced it is not by a lot. Therefore, we think it is probably the same people answering this survey. The high experience of respondents should, in turn, provide confidence that the answers provided to the survey are given by experienced people that know their area, increasing their reliability.

5.2 Agility of company

Since this study aims to examine the success factors of Agile teams it is important that the organization and teams used for data collection are Agile. Therefore during the first round of surveys the respondents were asked to state what Agile practices they use. The results showed, as visible in Table 4.7 and Table 4.8, that 28 out of the 50 Agile practices (56%) asked about, were used either sometimes or more often by more than 53% of the respondents. The practice that was used most had 98% stating they use it always. This practice was "Daily Meeting". None of the practices listed were never used, with more than 50% of respondents stating that they use 76% of the practices at least occasionally. This should make us fairly certain the respondents are knowledgeable in Agile ways of working. The least used practices, that more than 50% state they don't use or don't know about, are listed below

- Design Studio
- Facilitation
- Information Radiators
- Invest
- Lead Time
- Milestone Retrospective
- Mob Programming
- Relative Estimation
- Sign Up For Tasks
- Simple Design
- Sustainable Pace
- Team Room

If we look at the research presented by Alahyari et al. [7] none of these are part of the most used practices in their study. Alahyari et al. also mentions that their top factors fall in line with previous research. None of the practices that are not used by the respondents as much, fall under the category of practices that are used a lot in different companies. This should make us fairly certain that the respondents use the practices that are most commonly used today, to some extent, which should make us somewhat certain that the respondents are knowledgeable in Agile.

5.3 Most important success factors

The ranking of the most important success factors shown in Table 4.18 and Table 4.19 can help answer the first research question:

RQ1 What factors are important for an Agile team's success?

If we look at the top ten highest rated success factors from the tables mentioned we can compile this list showing the most important factors for an Agile team's success:

1. Good communication and collaboration within the team
2. A trusting team environment
3. Friendly and positive environment in team
4. Employees are willing to improve and get the chance to do so
5. Mental well being among team members
6. Low external blocking factors
7. Good communication and collaboration between different teams
8. Team members with great motivation
9. Well defined user stories
10. Prestigeless among team members

This list together with Table 4.18 and Table 4.19 answer the first research question.

5.4 Comparison of the ranked list of success factors to the literature

Misra et al. [3] identified the following factors for an Agile software project's success, where success is defined by reduced delivery schedules, increased return on investment, increased ability to meet with the current customer requirements, increased flexibility to meet with the changing customer requirements and improved business processes.

- Customer satisfaction
- Customer collaboration
- Customer commitment
- Decision time
- Corporate culture
- Control
- Personal characteristics
- Societal culture
- Training and learning

Comparing these success factors to the results in Table 4.18 and Table 4.19, we identify some similar success factors. The success factors regarding the customer from the results above, received a significantly lower rating in our results, where "Strong customer commitment and presence" was ranked 39 out of 57, "Good customer relationship" was ranked 34 out of 57 and "Customer having full authority"

was ranked 51 out of 57. These factors may not be entirely interchangeable, however they may give an indication of what are success factors for a team versus a project. The success factor "Training and Learning" from the list above may be related to the success factor "Employees are willing to improve and get the chance to do so" identified in Table 4.18 at place number 4. This is the only success factor in the top five from our results that is an organizational success factor.

The comparison of the findings of both studies indicates that there is a difference between success factors for an Agile team, and success factors for the success of an Agile project. However, it is important to note that the factors identified in [3] may not be identified in our study simply because the members of the Agile teams may not be that involved in customer relations, hence not valuing it that high. However, as discovered in 4.4.2, one interviewee indicated that there had been some issues with customer communication and collaboration and that it was easier with less involvement.

In the ranked list of success factors visible in Table 4.18 and Table 4.19 we can see that all of the top five factors are new success factors found from the first survey not present in literature. Out of the top ten factors, nine were new factors found from the first survey and not present in literature. As mentioned previously in Section 2.5 the high level factors found to have an impact on success by Chow and Cao [1] were:

- Team environment
- Project management process
- Agile software techniques
- Customer involvement
- Delivery strategy
- Team capability

How these relate to the success factors can be seen in Table 5.1

However, since the context of this study was different than the context used in the previous studies, we can't say for sure that we have identified new success factors compared to the ones from literature. What we can do, is to use these results to confirm and prioritize the old factors identified in the literature since they are based on the same context, that is project success.

The following sections will compare the ranked success factors to the different high level success factors found by Chow and Cao. Finally, the last section will identify why they differ on a more broad point.

5.4.1 Team Environment

When comparing the high level success factors found by Chow and Cao to the results acquired in this study we can see some similarities. The importance of team environment is visible due to the fact that the factor "A trusting team environment"

Table 5.1: Relation between high level factors to low level factors

<i>High Level Factor</i>	<i>Low Level Factor</i>	<i>Rank out of 57</i>
<i>Team environment</i>	Collocation of the whole team	43
	Coherent, self-organizing teamwork	16
	Projects with small team	44
	Projects with no multiple independent teams	55
<i>Project management process</i>	Following Agile-oriented requirement management process	36
	Following Agile-oriented project management process	42
	Following Agile-oriented configuration management process	45
	Strong communication focus with daily face-to-face meetings	37
	Honoring regular working schedule	46
<i>Agile software techniques</i>	Well-defined coding standards up front	40
	Pursuing simple design	17
	Rigorous refactoring activities	41
	Right amount of documentation	29
	Correct integration testing	15
<i>Customer involvement</i>	Good customer relationship	34
	Strong customer commitment and presence	39
	Customer having full authority	51
<i>Delivery Strategy</i>	Regular delivery of software	22
	Delivering most important feature first	12
<i>Team capability</i>	Team members with high competence and expertise	11
	Team members with great motivation	8
	Managers knowledgeable in Agile	24
	Managers who have adaptive management style	26
	Appropriate technical training to team	27

is ranked second by the respondents. "Friendly and positive environment in the team" also relates to the team environment and is ranked as third. The category "Team Spirit" is also loosely related to "Team environment" and 3 out of the 4 factors found in the "Team Spirit" category are present in the top ten factors ranked by the respondents. The factors that are presented under the category "Team Spirit" are shown below:

- Friendly and positive environment in team
- Mental well being among team members
- Prestigeless among teammembers
- Having fun at the workplace

Other than these newly found factors relating to team environment the factors found by Chow and Cao relating to this fall pretty far down on the ranking list with one of them being in 16th place while the others all fall below rank 43 as can be seen in Table 5.1. This leads to use to believe that the team environment factor as such needs to be expanded to fit more factors relating to it. The factors that got low rankings relate to the size of the team, the location of team members as well as how many teams are working on the project. An explanation to this could be the fact that video conferencing today is very easy to perform. It is also very easy to have video conferences with many people even no matter where they are.

5.4.2 Project management process

The project management process is however not deemed as important by the respondents ranking these at 36, 37, 42 and 45 out of 57, these rankings can be seen in Table 4.19. Thus this factor can not be deemed to have the same importance now as at the time of the study by Chow and Cao. This could however be due to that this study looks at a team's success and not a project's success.

5.4.3 Agile software techniques

When examining the ranked list of success factors it is also evident that factors relating to Agile software techniques are spread out over the list with some deemed important and some not so important. Two factors that got high rankings relating to this are "High testability of codebase" and Pursuing simple design" which were ranked at 15 and 17 respectively. Some factors that are not ranked as high relating to Agile software techniques ,and are ranked at 29 and 41 out of 57, are "Right amount of documentation" and "Rigorous refactoring activities". Based on this we can not say that Agile software techniques are not important neither important as a whole however, we can say that certain Agile software techniques are deemed more important than others for the team's success.

5.4.4 Customer involvement

The high-level factor customer involvement is deemed not as important as when Chow and Cao performed their study. Factors relating to this are ranked at rank

34, 39 and 51 respectively, this can be seen in Table 5.1. This is an interesting find since it is the customer that decides how the product evolves. However, this could be explained that the team, the respondents, does not need to communicate with the customer in order to be successful since this is done by either the scrum master, through the product owner or by someone in management. Thus, the team itself doesn't communicate with the customer directly and it is therefore not necessary for their success. This is also visible in the results were product owners and scrum masters rated the success factor "Good customer relationship" higher than the rest of the respondents. So these results could be present due to the high number of developers responding to the survey. The interviewees also mentioned that their had bee some problems when communication with the customer. This could also indicate why these types of factors were ranked high.

5.4.5 Delivery strategy

When viewing the factors relating to delivery strategy these fall on the ranks 12 and 22 out of 57. Therefore this factor can still be deemed important for the teams success.

5.4.6 Team capability

Some factors relating to team capability are ranked at 8 and 11 out of 57 respectively with none of them ranking below 27 out of 57. This makes us inclined to believe that the team capability is still important for the success of an Agile team.

5.4.7 Why are the important factors different from literature?

When looking at the most important success factors that were identified in Table 4.18, we can see that four of the top five are related to good communication and collaboration within the team, a trusting team environment, a friendly and positive environment within the team, and mental well being among the team members.

These are factors that are related to millenials in the workplace [34], where Myers and Sadaghiani explain that millenials expect more open communication with their managers and within the organization in general. They also argue that millenials thrive in teams as it is considered more fun and that there is less risk involved for the individual since they trust their team to back them up. Hershatter and Epstein also explains that millenials are more likely to prioritize work-life balance, and even let it affect their career choices. They present that 87% believe that work-life matters and use their time off to explore passions [35]

We believe that relating our findings to these theories concerning millenials may explain why the top five in this study differs from research that was conducted 10 years ago, since millenials are generally defined as people born between the early 1980's and the late 1990s. To investigate this further, we should have requested the

age of the survey respondents. This would mean, that ten years ago, an employee with a masters degree would have been born earliest in 1983 and the majority of the employees would have belonged to generation X, which is the generation before the millenials starting from the mid 1960s.

Another reason for the difference in factors may be that Agile has become more popular over the last decade and that the communication, collaboration and trust factors. These are all important parts of the Agile way of working and something that Agile promotes.

One possible explanation in the difference compared to existing literature, is the fact that previous research has been aimed towards Agile project success factors. This could indicate that there is a difference between what is important for a project to be successful and what is important for a team's success, even though the teams are a large part of the project.

5.5 Interviews

During the interviews, we discovered a common denominator in how all the interviewees responded regarding their retrospectives, where they reflected on the previous sprint. Although in different forms, they all seemed to perform their retrospectives by giving everyone a chance to voice their opinion on the last sprint. In one case, the feedback was collected through notes that was read up during the retrospective, which in turn granted a form of anonymity for the feedback. While this was a quite interesting approach, it gave the impression that the team members did not always feel that criticism was welcome. This impression was backed up during another interview where the interviewee said that a lot of decisions never actually happen if they are not agreed upon by the senior team members who had been there for a long time. As an example, in two cases we were told that some of the Agile transformation the company was trying to achieve was hindered by senior individuals who did not like that way of working. The general opinion from the interviews was that there were several blocking factors in the Agile transformation where the management did not agree on the Agile principles. Unfortunately, we could never confirm or deny this since we never got the chance to interview the managers.

Relating back to the retrospectives, they never seemed to evaluate the working environment, such as mental health or the satisfaction of the team members. This was an interesting finding for us, since the rating of the success factors later revealed that four of the top five success factors, rated by the team members, was related to communication and collaboration within the team, the mental health of team members, a trusting team environment and a friendly and positive team environment. If it is the case that these factors are never brought up during the retrospectives or within the company in general, this would argue that doing so would be of great benefit to the company.

5.6 Measurements

During the interviews the interviewees were asked to give an indication on how good they thought the measurements collected during the first survey were. The responses from the survey were collected "as is" and not all of them are truly measurements, but they are regarded as such and included in the analysis and result. The interview answers resulted in a ranked list seen in Table 4.21 and Table 4.22. There were six factors that all the interviewees agreed were good, meaning that they were all rated with a three, as seen in table 4.21. These measurements were:

- Time spent in predictably pointless meetings
- Average throughput end-to-end, with analysis of bottlenecks
- Quality of deliverable, number of bugs
- How well do we succeed in completing our sprint goals over a period of time/sprints
- Team health
- Days unable to continue work

Some of these measurements are highly linked to success factors that received a high rating, as stated previously in Section 5.4. One example is "Team Health", which relates to a number of success factors regarding the team, as identified when processing the first survey. Stated in the interviews were that many of the measurements from the first survey would be very good to measure, such as "Days unable to continue work", "Average throughput end-to-end, with analysis of bottlenecks" and "Quality of deliverable, number of bugs". It was stated that performing these measurements would help in improving their day to day work. For example, by measuring the amount of days a person is unable to work, it could make the organization more inclined to remove these blockers so that people are not just sitting around doing nothing and waiting for someone else.

The first measurement, time spent in predictably pointless meetings, may be related to the traits of the millennials [34]. This is because they have other advanced communication skills and methods, that they may feel are superior to meetings that may not always be very efficient.

If we compare the top six measurements found visible in Table 4.21 to the existing literature some differences become evident. The measurements identified here does not measure conformance to agility as suggested by Matthies et al. [24]. These measurements mostly relate to how the everyday work could be measured. Therefore it is more closely related to the measurements proposed by Kupiainen et al. [25] which suggest a number of measurements for the performance of Agile. An example of a similar measurement is critical defects sent by customer which is closely related to quality of deliverable, number of bugs. Overall the measurements received can be seen as more related to the performance of the team rather than to the conformance of ways of working.

The results found in Table 4.21 and Table 4.22 answer the third research question:

RQ2 How can you measure the success factors of an Agile team?

5.7 Threats to validity

Below the different threats to validity will be discussed, beginning with threats to construct validity and ending with threats to external validity.

5.7.1 Construct Validity

The construct validity refers to the relation of the theory behind the experiment and the observations [36]. There is the threat that, during the second round of surveys, the respondents did not understand what was meant with the different factors. Since they had no one present to answer questions for them, they could only answer based on their interpretation of the factor. To minimize this threat, the survey was examined by the supervisors in order to improve the understanding for any external readers. There was also a section beneath each category in the survey where the respondents could explain how they rated the factors, hence making it easier for us when processing the survey results if the respondents had left any comments about how they interpreted the factors.

When conducting surveys instead of interviews, there is a limitation depending on the enthusiasm and interest of the subjects that are answering the surveys. The surveys may generate better data if the subjects are willing to participate and takes the time and effort to answer the questions in an elaborate way, but may also generate answers of lower quality if the subjects are not interested in providing proper responses. We believe that performing two surveys may have been too exhaustive for the sample, since there was quite a mix in how they were treated with regards to responses. Although, we feel that the participants of the surveys may not have had enough encouragement from within the organization or enough incentive to perform the surveys as it would take time of their work. In the best case scenario, the respondents would've gotten permission and encouragement from the company to perform the surveys during their working hours in order to provide as good data as possible that could be useful for the company later down the road. The feeling that the respondents didn't get time required to complete the task during working hours, were backed by numerous respondents saying, along the lines "Sorry, I can't do this during working hours because there are other things I must do". Also, we believe that if the respondents felt that the outcome of the surveys had a possibility to help in their day to day work, they would have been more inclined to complete it during their free time.

One threat to the construct validity is that we specifically ask about success factors and measurements regarding the team, and doing so may possibly introduce a bias in the answers of the respondents, since they may not longer consider other factors that are not related to the team.

5.7.2 Internal Validity

The mapping of success factors identified from the first survey to the literature is a threat to validity. The mapping could have been done independently and then measuring intercoder reliability. However, we decided to perform the mapping in collaboration and had a discussion whenever we disagreed about how the mapping should be done. We felt that there would be no difference in performing the mapping individually, than comparing and discussing the proposed mappings and discussing them when we were not in agreement. The mappings were also made several times with some days in between just to ensure that the mappings were consistent and did not change on a day to day basis. The reviews were also recorded in order for the researcher to be able to listen to them again to make sure they understood the interviewees correctly.

As explained in section 3.1.2, you should not really use average on a likert scale since the numbers are ordinal rather than nominal. However, after further discussion, we decided to do so anyway because it was the only feasible way for us to somehow rank the importance of the factors based on the survey responses. The argument for not averaging the likert scale numbers, is that the distance from, say 3 and 4, may not be equal to the distance between 4 and 5. By calculating the averages for the likert scale numbers, we acknowledge that these averages are approximations. However, our other options would have been to rank the factors by min, max or mode according to responses. This, however, would not have produced very clear results and would have made it difficult to compile a ranked list of factors. There was also the possibility of using the \$100 test to rank the factors, but it was ruled out since there were too many factors to use this technique. As a result, we had a collection of independently rated success factors, and therefore decided to use an average to display which factors were considered more important according to our sample.

Another threat to our internal validity is whether the organization is actually agile, like they say they are. If the organization in reality is not very agile, this could affect the results of our surveys since the respondents may not actually be using the agile practices. Conducting the study at only one company should also be seen as a threat to validity as we are only asking the employees at this company what success factors are for them, and how to measure these. This threat relates a bit to the external validity threat about generalization.

5.7.3 External Validity

The threats to external validity are concerned with whether or not we can generalize the results outside of the company studied [36]. The results of this study are limited to the companies studied. But as stated by Alahyari et al. [7] most qualitative studies are rarely generalisable outside of the scope. In order to help the reader understand the setting, the background of the respondents as well as the agility of the organization was collected and presented in Section 4.2 and Section 4.3.

5.8 Future work

Further research into what factors an organization feels are most important for the success of their Agile teams are needed. Since this study was not able to interview managers at the company this could be a good strategy for further research into the area. Different companies, settings and countries could be interesting to examine as well. These studies can in turn be compared to the results gathered in this study in order to see the difference in views of managers and team members from different settings, companies and countries.

There is also need for further validation of the measurements proposed in this thesis. There might be even more measurements that have not been discovered in this thesis but still relate to the success factors uncovered. The ranking of the measurements by the interviewees is not enough to say if the measurements proposed are good or not therefore further validation is required. It would be good to perform some sort of ranking here as well in order to find out what is interesting to measure as such. Surveys could be used where the most important success factors are presented and measurements for these are asked for. These can then be ranked into a list of the most important measures for the respondents in relation to these factors.

6

Conclusion

This study was performed in collaboration with Knowit and their customer, a large telecom service company, with the goal to find success factors and measurements in Agile teams. To achieve this and to answer the research questions below, five interviews and two surveys were conducted.

RQ1 What factors are important for an Agile team's success?

RQ3 How can you measure the success factors of an Agile team?

The top five success factors that had the most impact on the success on an Agile team is shown below, while the full list of success factors is found in Table 4.18 and Table 4.19:

1. Good communication and collaboration within the team
2. A trusting team environment
3. Friendly and positive environment in team
4. Employees are willing to improve and get the chance to do so
5. Mental well being among team members

The six measurements that were deemed most valid to measure the success factors of an Agile team according to the interviews are shown below, the full list of ranked measurements can be seen in Table 4.21 and Table 4.22:

- Time spent in predictably pointless meetings
- Average throughput end-to-end, with analysis of bottlenecks
- Quality of deliverable, n.o bugs
- How well do we succeed in completing our sprint goals over a period of time/sprints
- Team health
- Days unable to continue work

These findings may help Agile teams and organizations to identify areas of improvement. By examining teams with these success factors in mind, one could improve the success of the team by aiding them in improving these factors. The proposed measurements could be used to identify if the team fulfills these factors. If that's not the case, they have a solid ground for improving the teams overall success. The findings indicate that a majority of the most important factors relate to the team

and the people within. Therefore, it is important that the team and the organization work together in order to improve the success of each team.

Bibliography

- [1] T. Chow and D. Cao, “A survey study of critical success factors in agile software projects,” *The Journal of Systems & Software*, vol. 81(6), pp. 961–971, 2008.
- [2] D. J. Power, A. S. Sohal, and S. Rahman, “Critical success factors in agile supply chain management - an empirical study,” *International Journal of Physical Distribution & Logistics Management*, vol. 31(4), pp. 247–265, 2001.
- [3] S. C. Misra, V. Kumar, and U. Kumar, “Identifying some important success factors in adopting agile software development practices,” *The Journal of Systems & Software*, vol. 82(11), pp. 1869–1890, 2009.
- [4] T. G. M. Khurum and M. Wilson, “The software value map - an exhaustive collection of value aspects for the development of software intensive products,” *Journal of Software: Evolution and Process*, vol. 25, pp. 711–741, 2013.
- [5] K. P. M. Khurum and T. Gorschek, “Extending value stream mapping through waste definition beyond customer perspective: Extending value stream mapping,” *Journal of Software: Evolution and Process*, vol. 26, pp. 1074–1105, 2014.
- [6] A. Aurum and C. Wohlin, “A value-based approach in requirements engineering: Explaining some of the fundamental concepts,” pp. 109–115, 2007.
- [7] H. Alahyari, R. B. Svensson, and T. Gorschek, “A study of value in agile software development organizations,” *Journal of Systems and Software*, vol. 125, pp. 271–288, 2017.
- [8] P. Castka, C. Bamber, J. Sharp, and P. Belohoubek, “Factors affecting successful implementation of high performance teams,” *Team Performance Management: An International Journal*, vol. 7(7/8), pp. 123–134, 2001.
- [9] L. Gren, R. Torkar, and R. Feldt, “Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies,” *Journal of Systems and Software*, vol. 124, pp. 104–119, 2017.
- [10] T. Dybå and T. Dingsøy, “Empirical studies of agile software development: A systematic review,” *Information and Software Technology*, vol. 50, pp. 833–859, 2008.
- [11] S. Nerur, R. Mahapatra, and G. Mangalaraj, “Challenges of migrating to agile methodologies,” *Communications of the ACM*, vol. 48(5), pp. 72–78, 2005.
- [12] J. Erickson, K. Lyytinen, and K. Siau, “Agile modeling, agile software development, and extreme programming: The state of research,” *Journal of Database Management (JDM)*, vol. 16, pp. 88–100, 2005.
- [13] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and

- D. Thomas, “Manifesto for agile software development.” [Online]. Available: <http://agilemanifesto.org/> Visited on: 2017-04-27.
- [14] “12 principles behind the agile manifesto.” [Online]. Available: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/> Visited on: 2017-04-27.
- [15] “Agile glossary and terminology.” [Online]. Available: <https://www.agilealliance.org/agile101/agile-glossary/> Visited on: 2017-04-27.
- [16] H. Merisalo-Rantanen, T. Tuunanen, and M. Rossi, “Is extreme programming just old wine in new bottles: A comparison of two cases,” *Journal of Database Management (JDM)*, vol. 16, pp. 41–61, 2005.
- [17] P. Mcbreen, *Questioning Extreme Programming*. Boston, MA, USA: Pearson Education, 2003.
- [18] M. Stephens and D. Rosenberg, *Extreme Programming Refactored: The Case Against XP*. Apress, 2003.
- [19] G. Keefer, *Extreme Programming Considered Harmful for Reliable Software Development 2.0*. AVOCA GmbH, Online Report, 2003.
- [20] M. L. D. Cohen and P. Costa, *An introduction to agile methods*. in Anonymous Elsevier Science & Technology, 2004, pp. 1–66.
- [21] B. Schatz and I. Abdelshafi, “Primavera gets agile: A successful transition to agile development,” *IEEE Software*, vol. 22(3), pp. 36–42, 2005.
- [22] D. Karlström and P. Runeson, “Combining agile methods with stage-gate project management,” *IEEE Software*, vol. 22(3), pp. 43–49, 2005.
- [23] S. Gren and J. Hochberger, “Validation studies of the group development questionnaire,” *Small Group Research*, vol. 27, pp. 143–170, 1996.
- [24] C. Matthies, T. Kowark, M. Uflacker, and H. Plattner, “Agile metrics for a university software engineering course,” in *2016 IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–5.
- [25] E. Kupiainen, M. Mantyla, and J. Itkonen, “Using metrics in agile and lean software development - a systematic literature review of industrial studies,” *Information and Software Technology*, vol. 62, pp. 143–163, 2015.
- [26] C. Kaner, S. Member, and W. P. Bond, “Software engineering metrics: What do they measure and how do we know?” in *In METRICS 2004. IEEE CS*. Press, 2004.
- [27] N. Fenton and J. Bieman, Eds., *Software Metrics: A Rigorous and Practical Approach, Third Edition. (3;3rd; ed.)*, 2014.
- [28] R. Torkar, *Survey Research: Original slides by Dr. Wasif Afzal*, Chalmers University of Technology Std., 2015.
- [29] N. Kurapati, V. S. C. Manyam, and K. Petersen, *Agile software development practice adoption survey*. in Anonymous Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 16–30.
- [30] G. M. Sullivan and A. A. Jr, “Analyzing and interpreting data from likert-type scales,” *Journal of Graduate Medical Education*, vol. 5, pp. 541–542, 2013.
- [31] J. Neale, “Iterative categorization (ic): a systematic technique for analysing qualitative data,” *Addiction*, vol. 111, pp. 1096–1106, 2016.
- [32] “Team.” [Online]. Available: <https://www.agilealliance.org/glossary/team/> Visited on: 2017-04-27.

- [33] S. Wheelan, “Group size, group development, and group productivity,” *Small Group Research*, vol. 40, p. 247–262, 2009.
- [34] K. K. Myers and K. Sadaghiani, “Millennials in the workplace: A communication perspective on millennials’ organizational relationships and performance,” *Journal of Business and Psychology*, vol. 25, pp. 225–238, 2010.
- [35] A. Hershatter and M. Epstein, “Millennials and the world of work: An organization and management perspective,” *Journal of Business and Psychology*, vol. 25, pp. 211–223, 2010.
- [36] R. Feldt and A. Magazinius, “Validity threats in empirical software engineering research - an initial survey,” in *Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering*, 2010, pp. 374–379.

A

Interview Script

Introduction

- Introduce ourselves
- Present the purpose of the interview & study
- Ask for permission to voice record the interview

Questions

- What is your role?
 - How many are in your team?
 - Which teams do you work with?
 - What products are you working on?
- What experience do you have with agile software principles?
- What is your management role?
- Do you evaluate the teams in any way today?
 - How do you evaluate the teams today?
- What does the term ‘success’ in the context of an agile team mean to you?
- What does the term ‘success’ in the context of a project mean to you?
- What factors do you think affect a team’s success?
 - How could you measure these factors?
- How would you rate these success factors? (Presents the factors collected from the first survey and literature seen in table 4.10 and table 4.11
 - Why do you rate them like this?
- Would you say these measurements could be used to measure these factors? (Presents measurements collected from first survey seen in table 4.12
 - Why/why not?
- Any additional comments?

B

Survey One

The survey was made with Google forms and the following are screenshots of what it looked like:

Background survey about agile teams

This section will gather insight about your current experience with agile principles and software development, as well as your role and team composition. The answers regarding identifying information (team name and product) will be anonymized before publishing.

***Required**

Background information *

	<1 year	1-2 years	3-4 years	5+ years
I have developed software for:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I have known about agile practices for:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I have developed software using agile practices for:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

My role in the team *

Developer

Project Manager

Product Owner

Other: _____

What is your team name? *

Your answer _____

What product are you working on? *

Your answer _____

Figure B.1: Top of first page of survey

How many people are in your team


3 4 5 6 7 8 9 10+

Number of people

Complexity of your product *

Average Complexity More than average complexity High Complexity Very High Complexity

Complexity

NEXT  Page 1 of 3

Never submit passwords through Google Forms.

Figure B.2: Bottom of first page of survey

Background survey about agile teams

*Required

Agile principles and methods

This section will investigate which agile principles your team uses and to what extent. If you want to read about a practice check out this website <https://www.agilealliance.org/agile101/agile-glossary/>.

We use *


	We don't use this	We use it rarely	We use it sometimes	We use it always	I am not familiar with the term
Acceptance Testing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Automated Build	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Backlog	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Backlog Refinement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
BDD(Behavior Driven Design)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Burndown Chart	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Collective Ownership	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Continuous Deployment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Continuous Integration	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Daily Meeting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Definition Of Done	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Definition Of Ready	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Design studio	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure B.3: Top of second page of survey

Estimation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Exploratory Testing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Facilitation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Frequent Releases	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Incremental Development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Information Radiators	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Integration	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Invest	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Iteration	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Iterative Development	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kanban Board	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lead Time	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Milestone Retrospective	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mob Programming	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mock Objects	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pair Programming	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Personas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Planning Poker	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Points (Estimates In)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Refactoring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Relative Estimation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure B.4: Middle of second page of survey

Scrum Of Scrums	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sign Up For Tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simple Design	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Story Mapping	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Story Splitting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sustainable Pace	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Task Board	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TDD	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Team Room	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Timebox	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Unit Testing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usability Testing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
User Stories	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Velocity	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Version Control	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

 Page 2 of 3

Never submit passwords through Google Forms.

Figure B.5: Bottom of second page of survey

Background survey about agile teams

*Required

What factors do you think affect the success of your agile team?

Here you will suggest which factors you think affect the success of your agile team. Try and come up with at least 5, we don't expect you to fill out all 15 fields but fill out as many as you can. Success factors could be for example technical, organizational, environmental etc.

Factor 1 *

Your answer _____

How could you measure this?

Your answer _____

Factor 2 *

Your answer _____

How could you measure this?

Your answer _____

Factor 3 *

Your answer _____

How could you measure this?

Your answer _____

Figure B.6: Factor 1-3 of last page of survey

Factor 4 *
Your answer _____

How could you measure this?
Your answer _____

Factor 5 *
Your answer _____

How could you measure this?
Your answer _____

Factor 6
Your answer _____

How could you measure this?
Your answer _____

Factor 7
Your answer _____

How could you measure this?
Your answer _____

Figure B.7: Factor 4-7 of last page of survey

Factor 8
Your answer _____

How could you measure this?
Your answer _____

Factor 9
Your answer _____

How could you measure this?
Your answer _____

Factor 10
Your answer _____

How could you measure this?
Your answer _____

Factor 11
Your answer _____

How could you measure this?
Your answer _____

Figure B.8: Factor 8-11 of last page of survey

Factor 12
Your answer _____

How could you measure this?
Your answer _____

Factor 13
Your answer _____

How could you measure this?
Your answer _____

Factor 14
Your answer _____

How could you measure this?
Your answer _____

Factor 15
Your answer _____

How could you measure this?
Your answer _____

Any additional suggested success factors or measurements?
Your answer _____


 Page 3 of 3

Figure B.9: Factor 12-15 of last page of survey

C

Survey Two

The survey was made with Google forms and the following are screenshots of what it looked like:

Success Factors

This is a continuation of the previous survey, don't be discouraged if you didn't do the first survey your response will be extremely valuable anyway.

The intention of this survey is to rate the importance of the provided success factors, which consists of the results from the previous survey along with additional success factors from literature.

The last section contains the measurements provided by respondents of the first survey, the goal is to find potential measurements that could be useful to measure a certain category.

***Required**

Background information *

	<1 year	1-2 years	3-4 years	5+ years
I have developed software for:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I have known about agile practices for:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I have developed software using agile practices for:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

My role in the team *

- Developer
- Project Manager
- Product Owner
- Scrum Master
- UX Designer
- Other: _____

Figure C.1: Top of first page of survey

What is your team name? *

Your answer _____

What product are you working on? *

Your answer _____

How many people are in your team? *

	3	4	5	6	7	8	9	10+
Number of people	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Complexity of your product *

	Average Complexity	More than average complexity	High Complexity	Very High Complexity
Complexity	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

NEXT

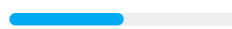
 Page 1 of 2

Figure C.2: Bottom of first page of survey

Success Factors

*Required

Success factors

Here we want you to estimate how important the factor is for the success of your team. It will also be very valuable if you fill out the reasoning behind your selection in the box below each category.

These categories are based on the results from the previous survey where we gathered as many different success factors as possible. The purpose here is to find out which of the identified factors are most important. Some factors are also based on literature combined with previous feedback.

Team spirit Success Factors *

	1. Not at all important	2.	3. Neutral	4.	5. Very important
Friendly and positive environment in team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mental well being among team members	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Prestigeless among teammates	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fun at work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any additional reasoning behind rating?

Your answer

Figure C.3: Team spirit success factors

Organizational Success Factors *

	1. Not at all important	2.	3. Neutral	4.	5. Very important
Strong executive support	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Committed sponsor or manager	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cooperative organizational culture instead of hierarchal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Oral culture placing high value on face-to-face communication	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organizations where agile methodology is universally accepted	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Collocation of the whole team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Facility with proper agile-style work environment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reward system appropriate for agile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Transparent organization	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Share vision with the company	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Low volatility of team compositions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Employees are willing to improve and get the chance to do so	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any additional reasoning behind rating?

Your answer _____

Figure C.4: Organizational success factors

People Success Factors *

	1. Not at all important	2.	3. Neutral	4.	5. Very important
Team members with high competence and diverse expertise	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Team members with great motivation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Managers knowledgeable in agile process	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Managers who have light-touch or adaptive management style	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Coherent, self-organizing teamwork	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good customer relationship	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sharing knowledge within the organization	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good communication and collaboration between different teams	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good communication and collaboration within the team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A trusting team environment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Team takes responsibility	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any additional reasoning behind rating?

Your answer

Figure C.5: People success factors

Process success factors *

	1. Not at all important	2.	3. Neutral	4.	5. Very important
Following agile-oriented requirement management process	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Following agile-oriented project management process	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Following agile-oriented configuration management process	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Strong communication focus with daily face-to-face meetings	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Honoring regular working schedule – no overtime	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Strong customer commitment and presence	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Customer having full authority	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any additional reasoning behind rating?

Your answer _____

Figure C.6: Process success factors

Technical success factors *

	1. Not at all important	2.	3. Neutral	4.	5. Very important
Well-defined coding standards up front	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pursuing simple design	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rigorous refactoring activities	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Right amount of documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regular delivery of software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delivering most important features first	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High testability of codebase	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Appropriate technical training to team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Well defined user stories	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Low external blocking factors	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Low external dependencies	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Proper development environment and tools	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Good architecture	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
High quality code	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any additional reasoning behind rating?

Your answer _____

Figure C.7: Technical success factors

Project success factors *

	1. Not at all important	2.	3. Neutral	4.	5. Very important
Project nature being non-life-critical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Project type being of variable scope with emergent requirement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projects with dynamic, accelerated schedule	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projects with small team	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projects with no multiple independent teams	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projects with up-front cost evaluation done	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projects with up-front risk analysis done	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projects with realistic planning	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Diverse projects	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Well focused sprints	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any additional reasoning behind rating?

Your answer

 Page 2 of 2

Figure C.8: Project success factors

D

Ranking from second survey

In this appendix the ranking of success factors compiled from the results of survey two are shown.

Table D.1: Ranked success factors. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Factor</i>	<i>Average</i>
1(TS)	Friendly and positive environment in team	4.851851852
2(PPL)	Good communication and collaboration within the team	4.740740741
3(TS)	A trusting team environment	4.592592593
4(PPL)	Team takes responsibility	4.592592593
5(TS)	Mental well being among team members	4.555555556
6(PPL)	Team members with great motivation	4.444444444
7(O)	Employees are willing to improve and get the chance to do so	4.407407407
8(T)	High testability of codebase	4.333333333
9(TS)	Pursuing simple design	4.296296296
10(T)	Low external blocking factors	4.296296296
11(TS)	Fun at work	4.222222222
12(T)	High quality code	4.222222222
13(T)	Delivering most important features first	4.185185185
14(T)	Proper development environment and tools	4.185185185
15(T)	Good architecture	4.185185185
16(T)	Regular delivery of software	4.111111111
17(PPL)	Coherent self organizing teamwork	4.074074074
18(TS)	Prestigeless among team members	4.037037037
19(O)	Organizations where agile methodology is universally accepted	4.037037037
20(PPL)	Good communication and collaboration between different teams	4.037037037
21(PRT)	Projects with realistic planning	4.037037037
22(O)	Cooperative organizational culture instead of hierarchical	4
23(O)	Transparent organization	4
24(PPL)	Good customer relationship	4

Table D.2: Ranked success factors cont. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Factor</i>	<i>Average</i>
25(O)	Share vision with the company	3.962962963
26(PPL)	Team members with high competence and diverse expertise	3.962962963
27(PPL)	Sharing knowledge within the organization	3.962962963
28(O)	Low volatility of team compositions	3.925925926
29(O)	Strong executive support	3.925925926
30(O)	Committed sponsor or manager	3.888888889
31(O)	Collocation of the whole team	3.888888889
32(PPL)	Managers who have light-touch or adaptive management style	3.851851852
33(PRS)	Strong communication focus with daily face-to-face meetings	3.851851852
34(T)	Appropriate technical training to team	3.851851852
35(T)	Well defined user stories	3.814814815
36(PRT)	Well focused sprints	3.814814815
37(PPL)	Managers knowledgable in agile process	3.777777778
38(T)	Right amount of documentation	3.703703704
39(T)	Low external dependencies	3.666666667
40(PRS)	Strong customer commitment and presence	3.666666667
41(T)	Well-defined coding standards up front	3.555555556
42(T)	Rigorous refactoring activities	3.555555556
43(O)	Facility with proper agile-style work environment	3.407407407
44(PRT)	Projects with small team	3.407407407
45(O)	Oral culture placing high value on face-to-face communication	3.37037037
46(PRS)	Honoring regular working schedule - no overtime	3.296296296
47(PRS)	Following agile-oriented project management process	3.222222222
48(PRT)	Diverse projects	3.222222222
49(PRS)	Following agile-oriented requirement management process	3.148148148
50(PRT)	Project nature being non-life-critical	3.148148148
51(PRT)	Projects with no multiple independent teams	3.111111111
52(PRT)	Project type being of variable scope with emergent requirements	3.074074074
53(PRS)	Following agile-oriented configuration management process	3
54(PRT)	Projects with up-front risk analysis done	2.962962963
55(PRS)	Customer having full authority	2.925925926
56(PRT)	Projects with dynamic, accelerated schedule	2.888888889
57(O)	Reward system appropriate for agile	2.814814815
58(PRT)	Projects with up-front cost evaluation done	2.666666667

E

Ranking from interviews

In this appendix the ranking of success factors compiled from the results of the interviews are shown. When performing these the factor "Team takes responsibility" was missed from the interview paper therefore it is not included in this list.

Table E.1: Ranked success factors. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Factor</i>	<i>Average</i>
1(PPL)	Good communication and collaboration within the team	5
2(TS)	A trusting team environment	5
3(O)	Employees are willing to improve and get the chance to do so	4.75
4(T)	Well defined user stories	4.75
5(O)	Low volatility of team compositions	4.5
6(PPL)	Team members with high competence and diverse expertise	4.5
7(T)	Low external blocking factors	4.5
8(TS)	Friendly and positive environment in team	4.5
9(TS)	Mental well being among team members	4.5
10(TS)	Prestigeless among team members	4.5
11(O)	Strong executive support	4.25
12(O)	Organizations where agile methodology is universally accepted	4.25
13(PPL)	Team members with great motivation	4.25
14(PPL)	Managers knowledgeable in agile process	4.25
15(PPL)	Coherent self organizing teamwork	4.25
16(PRS)	Following agile-oriented requirement management process	4.25
17(T)	Delivering most important features first	4.25
18(T)	Proper development environment and tools	4.25
19(PRT)	Well focused sprints	4.25
20(PPL)	Managers who have light-touch or adaptive management style	4
21(PPL)	Good communication and collaboration between different teams	4
22(TS)	Pursuing simple design	4
23(T)	Right amount of documentation	4
24(T)	Regular delivery of software	4

Table E.2: Ranked success factors cont. TS is Team Spirit, PPL is People, O is Organization, T is Technical, PRT is Project and PRS is Process.

<i>Rank / Category</i>	<i>Factor</i>	<i>Average</i>
25(T)	High testability of codebase	4
26(T)	Appropriate technical training to team	4
27(T)	Low external dependencies	4
28(T)	High quality code	4
29(TS)	Fun at work	4
30(O)	Committed sponsor or manager	3.75
31(O)	Cooperative organizational culture instead of hierarchical	3.75
32(PRS)	Following agile-oriented project management process	3.75
33(T)	Good architecture	3.75
34(O)	Transparent organization	3.5
35(PPL)	Good customer relationship	3.5
36(PPL)	Sharing knowledge within the organization	3.5
37(PRS)	Following agile-oriented configuration management process	3.5
38(PRS)	Strong communication focus with daily face-to-face meetings	3.5
39(PRS)	Strong customer commitment and presence	3.5
40(T)	Well-defined coding standards up front	3.5
41(T)	Rigorous refactoring activities	3.5
42(PRT)	Projects with realistic planning	3.5
43(O)	Share vision with the company	3.25
44(PRT)	Projects with small team	3.25
45(O)	Collocation of the whole team	3
46(PRS)	Honoring regular working schedule - no overtime	3
47(PRT)	Project nature being non-life-critical	3
48(O)	Facility with proper agile-style work environment	2.75
49(PRS)	Customer having full authority	2.75
50(PRT)	Project type being of variable scope with emergent requirements	2.75
51(PRT)	Projects with up-front risk analysis done	2.75
52(PRT)	Projects with dynamic, accelerated schedule	2.5
53(O)	Oral culture placing high value on face-to-face communication	2.25
54(PRT)	Projects with up-front cost evaluation done	2.25
55(PRT)	Diverse projects	2.25
56(PRT)	Projects with no multiple independent teams	2
57(O)	Reward system appropriate for agile	1.75