Agile Management Outside of Software Development

A Case Study of How Agile Management Should Be Used within a Small Team at a Fast Growing Financial Institute

Authors: Kajsa Alenmyr & Antonia Nilsson
Lund University 2016
Agile Management outside Software Development - A Case Study of How Agile Management Should Be Used within a Small Team at a Fast Growing Financial Institute

Copyright © 2016 Kajsa Alenmyr and Antonia Nilsson

Faculty of Engineering, Lund University
Box 118
SE-22100 Lund
Sweden

Printed in Sweden
Preface

This Master Thesis was conducted during the spring of 2016 in collaboration with Klarna AB and is the final academic project of the authors, graduating from Industrial Engineering and Management at Faculty of Engineering, Lund University. This project gave the authors insight in how to apply academic theory to a real case study as well as knowledge about the enablers and practices related to an agile work methodology. As the Master Thesis has combined the academic rigor of research with a real business problem it is the perfect bridge for the authors from being students to starting out their careers outside the university.
Acknowledgement

We would like to express our sincerest gratitude for the possibility to carry out this Master Thesis Project with the case company, Klarna AB. A special thanks to Wilhelm Back, Credit Risk Analyst at Credit Strategy & Analytics and our supervisor at Klarna AB, who has provided us with great insights during our project and always been available. We would also like to thank all team members at Credit Strategy & Analytics for welcoming us and being open and helpful during our entire project.

We would also like to thank everyone who answered our survey at other companies and who wrote us emails to give us information. It is obvious that the agile community is extremely helpful, which is something we are very thankful for.

Special thanks are of course also given to our supervisor, Bertil I Nilsson, for his guidance, support, and for sharing great and important input throughout the entire project. Lastly, we would also like to thank our opponents, Anna Bökberg and Karl Hedlund, for their valuable input on our Master Thesis.

Lund, May 2016.

Kajsa Alenmyr

Antonia Nilsson
Abstract

“Agile Management outside Software Development - A Case Study of How Agile Management Should Be Used within a Small Team at a Fast Growing Financial Institute”

Title

Agile Management Outside of Software Development

Authors

Kajsa Alenmyr and Antonia Nilsson
M.Sc. of Industrial Engineering and Management
Faculty of Engineering, Lund University

Supervisors

Bertil I Nilsson
Department of Production Management
Faculty of Engineering, Lund University

Wilhelm Back
Credit Strategy & Analytics
Klarna AB

Background

Agile Management is a management method based on the principles behind Agile Software Development. Compared to Traditional Management, Agile Management is more iterative and is therefore suited for fast paced and unsecure environments. Even though agile is a fairly old methodology; there are few case studies outside the software industry. There are indications, though, that Agile Management can work for other industries, especially those similar to software. To be able to use agile, some enablers and practices exist according to theory that are important for agile to be successful. Klarna AB is a fast growing financial institute where some teams work with agile today. Credit Strategy & Analytics is one of these teams, which consists of seven members who wanted to explore whether this methodology could work for them.
The purpose of the Master Thesis Project was to examine how agile methodology can be used at Credit Strategy & Analytics at Klarna AB by defining Success Factors and conducting a case study. More specifically, the aim was to provide recommendations on how the team can exploit agile methods in order to enhance their results.

In order to serve the purpose of the Master Thesis Project, four research questions were formed. When they were answered the goal was attained.

RQ 1: What agile practices are proposed in theory and previous case studies?

RQ 2: What Success Factors does a team need to possess in order to benefit from working with agile compared to other management methodologies?

RQ 3: What are the gaps between the Success Factors and practices that a team ideally should have to benefit from working with agile and the capabilities and practices the case team currently has?

RQ 4: What can be done in order for the case team to fill the gaps mentioned in RQ3?

The case study was limited to analyzing, evaluating ongoing processes, and proposing recommendations to Credit Strategy & Analytics at Klarna AB in Stockholm. This means that no implementation of the recommendations was prepared, conducted, or evaluated at the case team. The survey that was carried out only covered the basic enablers for working with agile and due to the time limitation of the Master Thesis Project, it was limited to 62 agile professionals.
The time limitation of the Master Thesis Project was 20 weeks and any areas that were not covered are suggested as future research.

The Master Thesis consisted of a theory review, a qualitative case study at Credit Strategy & Analytics at Klarna AB, and a quantitative survey with representatives outside the case company. Primary data was collected from the case study as well as the survey. Secondary data was found through a theory review as well as from previous case studies within the field. The case study consisted of unstructured interviews, observations, a survey, and in-depth interviews. The research has been conducted using an abductive approach and was carried out in an iterative manner to ensure good results in the end.

The Master Thesis Project came up with nine Success Factors for a team using agile. A Success Factor is an enabler the authors consider to be of large importance to be able to benefit from using agile. These are: **Flexibility, Acceptance, Management Support, Understanding, Leadership, Small teams, Dedicated Stakeholders, Long-term perspective, and Collocation.**

The Master Thesis Project also came up with recommendations for the case team based on these nine Success Factors together with theory and previous case studies regarding practices and tools. The recommendations for the case team’s agile strategy include adding an education about agile, both for new and current employees, creating competence cards for the team to increase competence visibility, become better at being on time for the short stand-up meetings, and start using Kanban instead of Scrum.
The authors have also created an alternative recommendation for continued use of Scrum. The recommendations for the use of Kanban include stopping time estimating tasks, having slightly longer stand-ups where short planning sessions are included, and have a Kanban board with a continuous flow of work. The recommendations for the use of Scrum include adding a Scrum master role in the team that changes between the team members every sprint, to plan for 70 instead of 80 hours in a sprint to have time for ad hoc assignments and meetings, and also to create tasks in a way so that they are possible to finish within a sprint of two weeks.

Keywords

Agile, Agile Management, Enabler, Kanban, Scrum, Sprint, Success Factor, Traditional Management, Financial Institute
Sammanfattning

Titel

"Agil styrning utanför programvaruutveckling - en fallstudie av hur agil styrning bör användas i ett litet team på ett snabbt växande finansiellt institut"

Författare

Kajsa Alenmyr och Antonia Nilsson
M.Sc. i Industriell ekonomi
Lunds tekniska högskola

Handledare

Bertil I Nilsson
Produktionsekonomi
Lunds Tekniska Högskola, Lunds Universitet

Wilhelm Back
Credit Strategy & Analytics
Klarna AB

Bakgrund

Syfte

Syftet med rapporten var att undersöka hur agil styrning kan användas utanför programvaruutveckling på Credit Strategy & Analytics på Klarna AB. Mer specifikt var målet att ge rekommendationer kring hur teamet bör använda agila metoder för att förbättra sina resultat.

Forskningsfrågor

För att möta syftet av examensarbetet formades fyra forskningsfrågor. När dessa frågor var besvarade var målet för uppsatsen nått.

FF1: Vilka agila tillvägagångssätt föreslås inom teorin och tidigare fallstudier?

FF2: Vilka “Success Factors” och tillvägagångssätt behöver ett team ha för att dra fördelar från att arbeta agilt jämfört med andra styrningsmetodiker?

FF3: Vad är gapen mellan de “Success Factors” och tillvägagångssätt som ett team bör besitta för att dra fördelar av att arbeta agilt, och de som fallstudieteamet besitter just nu?

FF4: Vad kan göras för att fallstudieteamet ska fylla gapen som nämns i FF3?

Avgränsningar

Fallstudien var begränsad till att analysera, utvärdera pågående processer och föreslå rekommendationer till Credit Strategy & Analytics på Klarna AB i Stockholm. Detta innebär att ingen implementation av rekommendationerna förbereddes, genomfördes eller utvärderades på fallstudieteamet. Enkätundersökningen som gjordes omfattade endast grundläggande möjliggörare för att arbeta med agilt och på
grund av tidsbegränsningen av examensarbetet begränsades den till 62 personer som arbetar med agilt.

Tidsbegränsningen för examensarbete var 20 veckor och alla områden som inte täcktes av projektet föreslås som framtida forskning.

Metod


Slutsats


Examensarbete resulterade också i rekommendationer för fallstudieteamet baserat på de nio ”Success Factors” som tidigare bestämts, och teori samt tidigare fallstudier. Rekommendationerna om fallstudieteamets strategi för agil styrning inkluderar att ha en utbildning om agil styrning,
både för nyanställda och för nuvarande anställda, att skapa kompetenskort för teamet för att öka transparensen, bli bättre på att vara i tid till avstämningsmöten och börja använda Kanban istället för Scrum.

Författarna har också skapat en alternativ rekommendation för fortsatt användning av Scrum. Rekommendationerna för användning av Kanban inkluderar att sluta tidsuppskatta uppgifter, ha något längre avstämningsmöten där ett kort planeringsmöte ingår, och att ha en Kanbanboard med ett kontinuerligt flöde av uppgifter. Rekommendationerna för användning av Scrum är att inkludera en Scrum master-roll i teamet, vilken skulle rotera bland teammedlemmarna varje sprint, att planera för 70 i stället för 80 timmar i en sprint för att ha tid för ad hoc-uppdrag och möten, och även för att skapa uppgifter på ett sätt så att de är möjliga att avsluta inom en sprint på två veckor.

Nyckelord

Agilt, Agil styrning, Möjliggörare, Kanban, Scrum, Sprint, Success Factor, Traditionell styrning, Finansiellt institut
# Agile Management Outside of Software Development

## Table of Contents

1 INTRODUCTION .................................................................................................................. 1  
1.1 BACKGROUND OF THE STUDY ...................................................................................... 1  
  
  1.1.1 Theoretical Background ................................................................................................. 1  
  1.1.2 Background of Klarna AB .............................................................................................. 2  
1.2 PROBLEM .......................................................................................................................... 3  
1.3 PURPOSE ............................................................................................................................. 3  
1.4 RESEARCH QUESTIONS ...................................................................................................... 3  
1.5 DELIMITATIONS ............................................................................................................... 4  
1.6 REPORT OUTLINE ............................................................................................................ 4  

2 METHODOLOGY .................................................................................................................. 7  
2.1 RESEARCH STRATEGY ....................................................................................................... 7  
2.2 RESEARCH DESIGN .......................................................................................................... 7  
2.3 DATA COLLECTION ........................................................................................................... 8  
  
  2.3.1 Observations ................................................................................................................ 9  
  2.3.2 Semi-Structured Interviews ......................................................................................... 9  
  2.3.3 Survey .......................................................................................................................... 11  
2.4 DATA ANALYSIS .............................................................................................................. 12  
2.5 WORK PROCESS ................................................................................................................. 13  
  
  2.5.1 The Qualitative Research Cycle .................................................................................. 13  
  2.5.2 Literature Review ........................................................................................................ 14  
  2.5.3 Benchmarking Against Work from Other Agile Professionals .................................... 14  
  2.5.4 Case Study at CS&A ..................................................................................................... 14  
  2.5.5 Analyzing and Identifying Data to Create Recommendations ..................................... 15  
2.6 CREDIBILITY .................................................................................................................... 16  
  
  2.6.1 Validity ......................................................................................................................... 16  
  2.6.2 Reliability .................................................................................................................... 17  
  2.6.3 Transferability ............................................................................................................. 18  

3 THEORY ................................................................................................................................ 21  
3.1 TM ...................................................................................................................................... 21  
  
  3.1.1 What is TM? ................................................................................................................ 21  
3.2 AM ..................................................................................................................................... 23  
  
  3.2.1 What is AM? ................................................................................................................ 23  
  3.2.2 History ......................................................................................................................... 26  
  3.2.3 Agile Methods .............................................................................................................. 27  
  3.2.4 Agile Support Systems .................................................................................................. 35  
  3.2.5 Agile outside Software Development ........................................................................... 35  
3.3 COMPARISON BETWEEN AM AND TM ......................................................................... 36  
3.4 ENABLERS FOR CHANGING MANAGEMENT METHODS ........................................... 38  
  
  3.4.1 Deciding Whether Agile is Right for a Team ................................................................ 39
4 PREVIOUS CASE STUDIES ........................................................................................................... 41
4.1 INCREASED PRODUCTIVITY WHEN GOING ALL-IN ................................................................. 41
4.2 AGILE IN LIBRARY IT INNOVATIONS ......................................................................................... 43
4.3 FROM SCRUM TO KANBAN AT STORMPATH ............................................................................. 46
4.4 AGILE LEGAL TEAM AT THE LONELY PLANET ........................................................................ 47

5 FINDINGS .................................................................................................................................. 51
5.1 FINDINGS FROM SURVEY ........................................................................................................... 51
5.2 FINDINGS FROM CASE STUDY AT KLRNA ................................................................................ 52
  5.2.1 Observations .......................................................................................................................... 52
  5.2.2 Interviews .............................................................................................................................. 54
5.3 FINDINGS OF ENABLERS ........................................................................................................... 60
  5.3.1 Enablers from the Survey ...................................................................................................... 60
  5.3.2 Enablers from Previous Case Studies .................................................................................... 61
  5.3.3 Enablers from Theory ........................................................................................................... 63
  5.3.4 The Nine SFs ......................................................................................................................... 64

6 ANALYSIS ................................................................................................................................... 67

7 CONCLUSION .............................................................................................................................. 75
  7.1 ANSWERS TO RESEARCH QUESTIONS .................................................................................. 75
  7.2 RECOMMENDATION ................................................................................................................ 76
    7.2.1 General Recommendation ................................................................................................. 76
    7.2.2 Recommendation for Kanban ............................................................................................ 77
    7.2.3 Recommendation for Scrum ............................................................................................... 78

8 DISCUSSION ............................................................................................................................... 79
  8.1 CONTRIBUTION TO THE ACADEMIA .................................................................................... 79
  8.2 GENERAL CONTRIBUTION ...................................................................................................... 79
  8.3 FURTHER WORK RECOMMENDATIONS ................................................................................ 80

REFERENCES ............................................................................................................................... 82

APPENDIX ..................................................................................................................................... 87
  APPENDIX 1 - INTERVIEW GUIDE .............................................................................................. 87
  APPENDIX 2 - SURVEY WITH OTHER COMPANIES AND CS&A .................................................... 92
  APPENDIX 3 - THE 41 ENABLERS SUGGESTED BY CONFORTO ET AL. ........................................ 98
  APPENDIX 4 - RESULTS FROM SURVEY WITH OTHER COMPANIES ......................................... 99
  APPENDIX 5 - LONG TEXT ANSWERS FROM SURVEY ............................................................... 114
  APPENDIX 6 - COMPETENCE CARD ............................................................................................ 115
List of Figures

Figure 1. The Methodology of the Master Thesis Project.

Figure 2. A visualization of the data collected in the Master Thesis Project.

Figure 3. The work-flow in TM, figure adapted from Lotz (2013).

Figure 4. The work-flow in AM, figure adapted from Lotz (2013).

Figure 5. Average productivity over time using waterfall methods or agile methods for a specific large technical company, figure adapted from Putnam (2014).

Figure 6. The nine success factors that the authors came up with during the Master Thesis Project.

Figure 7. The recommendations for CS&A’s agile strategy.

List of Tables

Table 1. Pros and cons with seven different agile methodologies, table adapted from OPS International LLC (2015).

Table 2. A comparison between agile and heavy methods based on different factors, table adapted from Awad (2005).

Table 3. Agile and heavyweight discriminators, table adapted from Awad (2005).

Table 4. The six practices within agile teams, table adapted from Conforto et al. (2014).

Table 5. A description of the ten most important enablers for AM and what they look like in agile teams, table adapted from Conforto et al. (2014).

Table 6. The tools used in the study of Chang (2010).

Table 7. The six first properties of Crystal and how they were met in the case study of Chang (2010).

Table 8. The agile methodologies applied by Lonely Planet Legal Affairs, table adapted from Sullivan (2013).

Table 9. The business challenges that the legal team at Lonely Planet were facing and how the results changed after implementing an agile model, table adapted from Sullivan (2013).

Table 10. Correlations between the factors brought up in the survey questions and whether it was a successful and best suited method.
# Glossary and Acronyms

## Glossary

### Master Thesis Report Definitions

The following expressions are defined by the authors of the Master Thesis Report and explain how the words are used in the report. Thus, they might at some point differ from the general agreed-upon definition.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad hoc assignment</td>
<td>An assignment that is not included in the product backlog.</td>
</tr>
<tr>
<td>Agile management</td>
<td>The alternative methodology to traditional management</td>
</tr>
<tr>
<td>Enabler</td>
<td>A condition found in theory that enables agile.</td>
</tr>
<tr>
<td>Jira</td>
<td>Jira Software</td>
</tr>
<tr>
<td>Klarna</td>
<td>Klarna AB</td>
</tr>
<tr>
<td>Management</td>
<td>Both project management and process management is covered in this word.</td>
</tr>
<tr>
<td>Master thesis project</td>
<td>The project carried out by the authors that is described in this report.</td>
</tr>
<tr>
<td>Master thesis report</td>
<td>This report.</td>
</tr>
<tr>
<td>Process management</td>
<td>Ongoing work without a limited time-frame.</td>
</tr>
<tr>
<td>Project management</td>
<td>Work with a clear outcome and a limited time-frame.</td>
</tr>
<tr>
<td>Success factor</td>
<td>One of the nine enablers that the authors claim to be the most important.</td>
</tr>
<tr>
<td>Traditional management</td>
<td>The alternative methodology to agile management</td>
</tr>
</tbody>
</table>
Agile Definitions

The following expressions are used in the agile society and in the Master Thesis Report. They might not be known by a reader who is not familiar with the subject and are therefore explained in advance.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptance criteria</td>
<td>The criteria that defines the features of a task in order for a product owner or customer to accept it (Glossary n.d.).</td>
</tr>
<tr>
<td>Acceptance testing</td>
<td>The test that determines whether a system meets the needed requirements. It is usually expressed as an example or a usage scenario (Guide to Agile Practices n.d.).</td>
</tr>
<tr>
<td>Agile coach role</td>
<td>A person that helps the team adopt and improve agile methods and practices (Kelly 2009).</td>
</tr>
<tr>
<td>Backlog</td>
<td>The list of features and tasks that are on the team's agenda. This is where the tasks are prioritized and the tasks with the highest priority in the backlog are the ones handled first (Guide to Agile Practices n.d.).</td>
</tr>
<tr>
<td>Backlog grooming</td>
<td>The team meets regularly to &quot;groom&quot; the product backlog, meaning removing user stories that are no longer relevant, creating new tasks from discovered needs or correcting estimates of tasks (Guide to Agile Practices n.d.).</td>
</tr>
<tr>
<td>Burndown chart</td>
<td>A chart which tells the team the quantity of work remaining on one axis, and the time elapsed on the other (Guide to Agile Practices n.d.).</td>
</tr>
<tr>
<td>Coding standards</td>
<td>Programmers use the same standards when they create the code. This makes it easier to refactor, extend, and maintain the code (Coding Standard n.d.).</td>
</tr>
<tr>
<td>Collective code ownership</td>
<td>Every team member is allowed to change any code file that the team is working with (Guide to Agile Practices n.d.).</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Used to minimize the duration and effort required by</td>
</tr>
</tbody>
</table>
each integration stage, as well as being able to deliver a product version at any moment. This is achieved through the use of specific tools (Guide to Agile Practices n.d.).

**Epic**
A story that represents a large story or set of features, which is further described by user stories (Glossary n.d.).

**Estimation**
The quantified evaluated measure of the time needed to carry out a given task (Guide to Agile Practices n.d.).

**Interactive facilitated workshops**
A facilitator will be a part of a workshop with the aim to create good conditions for effective group processes (Guide to Agile Practices n.d.).

**Kanban board**
The visual board where all tasks are handled and moved when they change status. Can be both offline on a whiteboard, or online in a software system (Guide to Agile Practices n.d.).

**Metaphor**
When a team within extreme programming creates a vision of how the program works through a metaphor. For example “this program works like a hive of bees, going out for pollen and bringing it back to the hive” (Metaphor n.d.).

**MoSCoW**
An acronym used to prioritize work, or aspects of a specific task. Stands for must have, should have, can have, and will not have (Glossary n.d.).

**Pair programming**
Two programmers work together and share the same workspace, where one is the “driver” and one the “navigator”. They are expected to switch roles every five minutes (Guide to Agile Practices n.d.).

**Planning game**
The main planning process of the extreme programming method. The game is a meeting that takes place once every iteration, typically once a week (Planning Game n.d.).
<table>
<thead>
<tr>
<th><strong>Agile Management Outside of Software Development</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prioritizing</strong></td>
</tr>
<tr>
<td><strong>Product owner role</strong></td>
</tr>
<tr>
<td><strong>Refactoring</strong></td>
</tr>
<tr>
<td><strong>Retrospective</strong></td>
</tr>
<tr>
<td><strong>Scrum master</strong></td>
</tr>
<tr>
<td><strong>Simple design</strong></td>
</tr>
<tr>
<td><strong>Small, frequent releases</strong></td>
</tr>
<tr>
<td><strong>Sprint/Iteration</strong></td>
</tr>
<tr>
<td><strong>Stand-up meeting</strong></td>
</tr>
<tr>
<td>Term</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>Sustainable pace</td>
</tr>
<tr>
<td>Task</td>
</tr>
<tr>
<td>Task board</td>
</tr>
<tr>
<td>Test-driven development</td>
</tr>
<tr>
<td>Ticket</td>
</tr>
<tr>
<td>User story</td>
</tr>
<tr>
<td>WIP-limits</td>
</tr>
</tbody>
</table>
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>Agile Management</td>
</tr>
<tr>
<td>ASD</td>
<td>Agile Software Development</td>
</tr>
<tr>
<td>CR</td>
<td>Credit Risk</td>
</tr>
<tr>
<td>CS&amp;A</td>
<td>Credit Strategy &amp; Analytics</td>
</tr>
<tr>
<td>DSDM</td>
<td>Dynamic Systems Development Method</td>
</tr>
<tr>
<td>FDD</td>
<td>Feature-Driven Development</td>
</tr>
<tr>
<td>IU</td>
<td>Implementation Unit</td>
</tr>
<tr>
<td>LSD</td>
<td>Lean Software Development</td>
</tr>
<tr>
<td>QSM</td>
<td>Quantitative Software Management Inc.</td>
</tr>
<tr>
<td>SF</td>
<td>Success Factor</td>
</tr>
<tr>
<td>TM</td>
<td>Traditional Management</td>
</tr>
<tr>
<td>WIP</td>
<td>Work In Progress</td>
</tr>
<tr>
<td>XP</td>
<td>Extreme Programming</td>
</tr>
</tbody>
</table>
1 Introduction

This chapter initiates the study by presenting the background of the Master Thesis Project, the underlying problem description, and research questions. Additionally, the chapter presents the purpose of the Master Thesis Project together with delimitations, deliverables, and an outline of the report.

1.1 Background of the Study

1.1.1 Theoretical Background

Agile Management (AM) is a management methodology based on the principles behind Agile Software Development (ASD). ASD was sprung from the Agile Manifesto in 2001 where 17 developers created and signed it (What is Agile? n.d.). The Agile Manifesto is based on four basic principles. Firstly, it stresses the importance of valuing individuals and interactions over processes and tools. Secondly, a working software is more important than a comprehensive documentation. Thirdly, collaboration with the customers outweigh the importance of contract negotiation. Lastly, responding to change is prioritized rather than following a predefined plan (Fowler & Highsmith 2001).

AM is an alternative to Traditional Management (TM). In TM, almost everything is planned before the execution of the work. Once the work is done, there will not be any revisions of the outcomes. A common method within TM is the waterfall method, which symbolizes the inability of moving up the waterfall (Hass 2007).

Even though agile is a fairly old technique within software development, there are few case studies made outside the software industry. There are indications, though, that AM is easiest implemented in areas similar to software development, such as product development (Conforto et al. 2014). Agile companies are those that combine stability and speed, which means that the terms are not contradicting as some argue (Bazigos, De Smet & Gagnon 2015). Bazigos, De Smet & Gagnon (2015) showed in a study that agile companies have a better long-term health.
In order to succeed when implementing any new management method, some enablers need to be fulfilled. A few of them are general and similar for many methods, while others differ. Conforto et al. (2014) have discovered some enablers, internal and external factors, making it beneficial to use AM over TM. They are factors such as the structure of the company, the size and location of the team, and the involvement of stakeholders (Conforto et al. 2014).

1.1.2 Background of Klarna AB

Klarna AB (Klarna) was founded in Stockholm 2005 with the business idea to simplify buying (About Us n.d.). Klarna is a financial institute offering invoices and partial payments to customers when they shop online. As of 2015, Klarna has a total of 45 million end-users (Klarna Statistics n.d.).

The case team belongs to the department Credit Risk (CR), which mainly works with the departments Commercial and Engineering. Some of the departments at Klarna work with AM, primarily the teams working with software development, but also some teams at CR (Mr. W Back 2016, pers.comm., 8 February).

Within CR, there is a team called Credit Strategy & Analytics (CS&A). This team decides on limits for purchases in different countries (Mr. W Back 2016, pers.comm., 8 February). It consists of seven people where the vast majority has been in the team shorter than six months. CS&A has worked with agile for a short period of time, more specifically since November 2015, and is still trying to find the right approach to use it (Mr. J Olesen 2016, pers.comm., 17 February).

A team member found that CS&A had no clear deadlines and lack of focus and therefore suggested implementing AM. The aim of the implementation for CS&A was primarily to increase structure and visibility, as well as to facilitate prioritization between, and ownership of, different tasks (Mr. J Olesen 2016, pers.comm., 17 February). The team works with sprints of two weeks, planning sessions of one hour, stand-ups of approximately ten minutes every other day, and retrospectives of one hour at the end of a sprint. To make the work visible for all members of the team they use the software tools Jira Software (Jira) and Confluence by Atlassian (Mr. W Back 2016, pers.comm., 8 February).
1.2 Problem

The problem is twofold: there is a small amount of knowledge of AM outside software development, and especially case studies within this field (Conforto et al. 2014), and CS&A has a need of further developing their work methodology. This Master Thesis Report will therefore contribute to academia as it increases the amount of case studies on agile outside software development. It also means that there needs to be a combination of literature studies and own research to find the Success Factors (SFs) and the optimal solution for CS&A.

As CS&A has recently started to use AM they first need to know if it is the right methodology for them to use and secondly, how to use it optimally. The aim is to find SFs to decide if CS&A has what is needed for AM to be as beneficial as possible, and to find remedies where there are gaps between the case team’s present capabilities and the SFs.

1.3 Purpose

The purpose of the Master Thesis Project was to examine how agile methodology can be used at Credit Strategy & Analytics at Klarna AB by defining Success Factors and conducting a case study. More specifically, the aim was to provide recommendations on how the team can exploit agile methods in order to enhance their results.

1.4 Research Questions

In order to serve the purpose of the Master Thesis Project, four research questions were formed. When they were answered the goal was attained.

RQ 1: What agile practices are proposed in theory and previous case studies?
RQ 2: What Success Factors does a team need to possess in order to benefit from working with agile compared to other management methodologies?
RQ 3: What are the gaps between the Success Factors and practices that a team ideally should have to benefit from working with agile and the capabilities and practices case team currently has?
RQ 4: What can be done in order for the case team to fill the gaps mentioned in RQ3?
The first question was answered through a literature review of theory and case studies within the subject. To be able to answer the second question a survey was created and sent out to professionals who have experience of working with AM. The survey investigated their enablers for working with agile. The survey was created through combining theory from literature and inferences of that literature. The third question was treated through conducting observations and interviews with CS&A. The interviews were based partly on following up the survey, which the team answered in beforehand, and partly on understanding the current situation in the team. The fourth question was answered through an abductive approach combining the research of the first two questions and conducting an analysis of these.

1.5 Delimitations

The case study was limited to proposing recommendations to CS&A at Klarna in Stockholm. This means that no implementation of agile methodology was prepared, conducted, or evaluated at the case company. The survey carried out only covered the basic enablers for working with agile and due to the time limitation of the Master Thesis Project it was limited to 62 professionals working with agile.

The time limitation for the Master Thesis Project was 20 weeks and any areas that are not covered will be suggested as future research.

1.6 Report Outline

Chapter 1 Introduction

This chapter initiates the study by presenting the background of the Master Thesis Project, the underlying problem description, and research questions. Additionally, the chapter presents the purpose of the Master Thesis Project together with delimitations, deliverables, and an outline of the report.

Chapter 2 Methodology

The following chapter will describe the methodology of the Master Thesis Project and Report. It will handle the rationale of the research, data collection and analysis. Lastly, it will discuss the credibility through parameters such as validity, reliability and transferability.
Chapter 3 Theory
This chapter introduces AM and TM by presenting the history, methods, practices, and when to use the different methods. There is also a comparison between the two management methods.

Chapter 4 Previous Case Studies
This chapter brings up some previous case studies made in the field of AM, both within, and outside of software development. The challenge, methodology implemented, results and conclusion from the different studies are presented.

Chapter 5 Findings
In this chapter the findings from the case study at CS&A and the survey to other professionals working with agile will be shown. There will also be an aggregation of all enablers found in the theory, external case studies, and the survey with a conclusion of the final nine SFs.

Chapter 6 Analysis
This chapter includes a discussion about the current situation at CS&A, what is good as it is and where gaps exist between the team’s method and best practice. All facts that do not have a reference come from the findings of the Master Thesis Project.

Chapter 7 Conclusion
In this chapter the analysis will be synthesized into a conclusion of the research questions and recommendations are formed to CS&A about how they can improve their work in order to become best practice.

Chapter 8 Discussion
In this chapter there is a discussion about how the Master Thesis Project has contributed to the academia, the chosen methodology, and propositions for further work within the subject.
2 Methodology

The following chapter will describe the methodology of the Master Thesis Project and Report. It will handle the rationale of the research, data collection and analysis. Lastly, it will discuss the credibility through parameters such as validity, reliability and transferability.

2.1 Research Strategy

When creating this Master Thesis Project, the authors used a qualitative research strategy together with an abductive reasoning approach.

An abductive research approach was chosen instead of inductive and deductive reasoning. It differs from the deductive and inductive approach as they focus on finding relations between known structures. In abductive research, the researchers go from established knowledge to new knowledge, by observing something that cannot be explained by existing theories and best practices. The deductive approach is about reviewing the literature, and then creating hypotheses and propositions that are tested in a research setting. The inductive approach, on the other hand, uses observations to form propositions, without consulting the literature (Kovacs & Spens 2005).

The abductive approach is more focused on specific situations that might deviate from the rule, than on generalizations (Kovacs & Spens 2005). The method was appropriate for the Master Thesis Project, as it revolved around a case study. It was therefore important for the researchers to recognize which of the findings that could be generalized and form propositions to the theoretical rule, and which ones that were specific to the situation (Kovacs & Spens 2005). Furthermore, in abductive research the data is collected at the same time as the theory is built, which supports working in iterations (Kovacs & Spens 2005). This was suitable as the research is about agile.

2.2 Research Design

A case study constituted the main part of the Master Thesis Project. According to Yin (2014), case studies are a good approach when the phenomena researched might be hard to distinguish from its context. The design of a case study is flexible, and questions
and directions can be revised during the course of the study. The data collected in the study is mainly qualitative (Höst, Regnell & Runeson 2006).

To complement the case study, a literature study and survey was conducted. As mentioned, data collection was conducted at the same time as the analysis. This enabled the authors to use the gathered data and reiterate when further depth in the analysis was needed. The data collected was compared to the theory in order to find similarities and differences, which could be used in the further iterations of the Master Thesis Project.

2.3 Data Collection

There are two main research strategies, qualitative and quantitative (Höst, Regnell & Runeson 2006). The difference between qualitative and quantitative research strategy is not always entirely clear. Qualitative research is a good strategy when gathering information about for example underlying attitudes (Lekvall & Wahlbin 2001). The strategy enables more depth of detail in the participants’ experiences (Hennink, Hutter & Bailey 2011). This suited the study well as it was meant to understand the underlying enablers to the fitness of working with agile. Furthermore, a qualitative strategy is beneficial when researching a relatively young subject (Starrin & Svensson 1994), which the Master Thesis Project was doing. Quantitative research, however, is when the results from the study is presented in numbers and can be analyzed using statistical calculations (Lekvall & Wahlbin 2001).

The case study consisted of semi-structured interviews and observations, and to complement that information a survey and literature review was conducted. The semi-structured interviews were chosen, as they tend to give more information about a topic, than focus groups do (Jamshed 2014). Also, there is a risk that a few more dominant respondents will dominate the discussion, not letting opinions of everyone through. The advantage of using focus groups is that it may create a group dynamic, which can make the participants talk about attitudes and opinions that the individual interviews would not have brought up (Lekvall & Wahlbin 2001). However, the authors decided that the individual interviews were preferable.
2.3.1 Observations

When conducting observations the researcher observes an event, and notes the sequence of the things happening (Robson 2002). It is a good method for confirmation and complementary data gathering (Jamshed 2014). This is also how the method was used, together with the interviews. In most observations conducted, the authors were complete observers, meaning that they participated in the event as little as possible. However, they were still present in the room where the event took place and took notes visibly. By not partaking in the events, the authors ensured that they would influence the event minimally (Höst, Regnell & Runeson 2006). The events that the authors observed were sprint planning meetings, stand-up meetings and retrospectives.

2.3.2 Semi-Structured Interviews

Semi-structured interviews are in-depth interviews with an interview guide that has predefined questions and open-ended answers. In general, these interviews are between 30 minutes and up to over an hour (Jamshed 2014), which was also the case in the Master Thesis Project. The interview guide is a beneficial tool as it helps explore the subject more systematically, and thereby increases the efficiency and effectiveness of the interview. Furthermore, the interview guide is centered around a few core questions, which in turn have a few follow-up questions on the topic. It is suggested to have a test interview before having the actual interviews as the interviewers will have the possibility to further improve the interview guide (Jamshed 2014). Jamshed (2014) suggests recording the interviews, as it is more reliable than just having hand-written notes. The authors chose to both collect data through using a recorder and taking notes.

All interviews were conducted face-to-face as it is supposed to give the interview better dynamics than for example using phone interviews (Lekvall & Wahlbin 2001). Also, the interviews were longer than 30 minutes, which exceeds the suggested time limit for phone interviews (Lekvall & Wahlbin 2001).

Pre-Interviews with Key Information-Holders

In order to gather basic knowledge about Klarna, the agile initiative at CS&A, and agile work at Klarna in general, interviews were conducted with different key information-
holders. The interviews were around 30 minutes long. With some of the interviewees it was sufficient with one sit-down, while others required a follow-up.

*Interviews with CS&A*

As the team is small, interviews could be conducted with all seven team members. Therefore, a comprehensive study was made, which is possible when the population is small (Höst, Regnell & Runeson 2006). All the interviews, except for one, were conducted in the same day in order to minimize the probability for the team members to discuss the content and therefore inflict on each others’ answers. Due to the same reason, the interview guide was not sent to the interviewees in advance.

The interview that was not conducted in the same day was a test interview with one of the team members. As this interviewee was the authors’ supervisor at the case company, it was concluded that the information gathered from the interview might have been compromised anyway from previous correspondence with the authors. Therefore, the opportunity was used to test the interview guide. This interviewee was also sent the guide in advance to get feedback. The data gathered from the interview was therefore taken less into account than the data gathered from the other interviews.

Before the interviews, an online survey was conducted with the interviewees to get a good base for follow-up questions in the interviews. This survey is treated closer in section 2.3.4. The interviews were around one hour long, conducted at Klarna’s Stockholm office and contained 73 questions divided into 27 question areas, which in turn were divided into five parts. The interview guide was based on the survey, and contained follow-up questions on it. This means that the 27 question areas were mainly the questions from the survey. The parts were: an introduction to the interview format, background questions, the agile work, the current team situation, and lastly some wrap-up questions. After the test interview some questions that were more fact-based and not needed to be posed to all interviewees were removed to enable the interview to be at most one hour long. The interview guide can be further consulted in appendix 1.
2.3.3 Survey

As part of the Master Thesis Project, a quantitative survey was carried out. Surveys are a good approach when needing results from a large number of people (Höst, Regnell & Runeson 2006). The survey was conducted online via Google Forms.

The survey was sent out to CS&A and professionals with experience from working with agile. However, the two groups got separate links to access the survey so that the answers could be kept apart. The only difference between the surveys sent out to the two groups was that the members of CS&A were asked to give out their name. The survey had five sections and was mainly based on the enablers suggested by Conforto et al. (2014). The first was an information text without questions, the second contained background questions, the third contained the nature of agile work at the project or process, the fourth was about capabilities and needs for successful implementation of agile, and the last section contained concluding remarks of the overall success of agile work in the process or project. For further details of the survey see appendix 2.

Survey with CS&A

The seven members of CS&A were asked to complete the survey ahead of their interviews to give a basis of information about their opinions and work with agile in their team. The reason that the names were asked for was to be able to match the survey response with the interviewee. All members answered the survey, there were therefore no external non-response. There was no internal non-response as all team members answered all questions.

External Survey

The aim of conducting an external survey was to identify best practices and find enablers for working with agile. To find the respondents of the survey, convenience and snowball sampling were used. These were chosen as the study was of exploratory nature (Höst, Regnell & Runeson 2006), and due to the time constraints of the Master Thesis Project. The link for the survey was posted on several communities with relations to the agile community. Also, the authors scanned their own networks to find more respondents that fit the criteria: employees in, or consultants to companies that work with agile either in their processes or projects. When finding possible respondents, the au-
thors asked them to scan their network for suitable respondents, thereof the snowball sampling. This sampling method should be avoided in the furthest, but is sanctioned when the target population is small and widely spread (Lekvall & Wahlbin 2001), which was the case for this study.

The survey got 62 responses and was open to answer between February 29th and April 5th 2016. The external non-response was hard to measure due to the sampling methods. The internal non-response was 0.16%.

2.4 Data Analysis

Each interview was conducted in the same manner. Of the two authors, one took notes and the other one was the interview conductor asking questions. Both authors asked follow-up questions. Further, the interviews were recorded so that the authors could go back and consult the recordings if there were any lacking parts in the notes from the interviews. Following, the authors summed up the interviews and sent the transcripts to the interviewees for approval.

All the answers from the interviews were processed by gathering the respondents’ answers for the same question in a table. This allowed the authors to get an overview of what the entire team thought about a specific topic. The tables with the collected answers where then printed out before being coded. The authors coded half the interview questions each and then all questions were discussed together. In the coding process, answers that occurred several times, that related to something in the theory or that in any other way was seen as extra interesting by the authors, were highlighted. The highlighted answers were then categorized into different categories for every question in order to find similarities and differences between the answers. When this was done, an Excel file was created where every question with its categories were compiled. This Excel file constituted the base for the findings. As the interviews were anonymous, the result is displayed in an aggregated way in order to not give out too detailed information about single interviewees.

The survey conducted within CS&A was analyzed together with the answers from the interviews. The survey conducted with professionals outside CS&A was analyzed in Ex-
cel. The analysis was divided into two parts. In the first part, every question was analyzed on a basic level where the number and percentage of respondents for every answer were calculated. In the second part, the correlations between different aspects brought up in the survey, and how successful agile had been were calculated. Correlations were also calculated for the same aspects and whether it had been the best method to work with in the process or project.

The enablers that the authors found from the theory, previous case studies and findings of the survey were collected in an Excel file. Each author studied the enablers to find recurring enablers. After this, nine SFs were identified that were seen as the most important.

2.5 Work Process

2.5.1 The Qualitative Research Cycle

When working with a qualitative model, the work is iterative, as mentioned. However, it can be divided into three subparts: the design cycle, the ethnographic cycle, and the analytic cycle. Since the work is iterative these do not come in sequence, but are rather performed simultaneously. In the design cycle, the research question is formulated and a literature review is conducted. In the ethnographic cycle the execution of the method is performed. Lastly, in the analytic cycle, the authors analyze and conclude the gathered data (Hennink, Hutter & Bailey 2011).

The qualitative research cycle is how the authors worked in the Master Thesis Project. Firstly, a few research questions were decided upon. Secondly, the authors conducted an initial literature review in order to get to know the research subject better and decide how the study at the case company could be conducted to optimize data collection. The data collection was divided into three parts. Firstly, initial observations, unstructured interviews and a survey at the case team were conducted. Secondly, survey to other professionals working with agile was sent out. Thirdly, semi-structured interviews with the case company were conducted. Each of the data collection steps were used to construct a more sophisticated next step in the process. The data gathered also gave a good basis to achieve a deeper understanding about the subject from narrowing
the scope of the literature. During the entire process, the research questions were revisited and revised. It was important to be flexible and to be able to change the plan due to any new information. All three cycles were therefore often revisited throughout the Master Thesis Project.

2.5.2 Literature Review

The first step in the literature review was to get to know the subject better in order to be able to refine the research questions. This meant a broad scope of the literature scanned, an explorative review. The initial review gave an understanding on what research and case studies that had been done before and where there was a gap to be filled (Lekvall & Wahlbin 2001). It was found that there was a gap regarding the amount on case studies done on the subject of working with agile outside software development (Conforto et al. 2014). As there was a gap here, the scope of the scanning of previous case studies were widened to both inside and outside of software development.

Understanding the theory in the literature review enabled forming a framework for the continued research. This, because it gave the authors an understanding of what questions that were relevant and what methods and techniques that have been successful when working with agile.

2.5.3 Benchmarking Against Work from Other Agile Professionals

A survey was sent out to professionals working with agile outside CS&A. 62 respondents answered the survey. This was done in order to find best practice examples on how working with agile can be done, as well as identifying enablers that later could be compared with the enablers found in the literature.

2.5.4 Case Study at CS&A

The case study at Klarna was performed due to three different reasons. Firstly, to find gaps between the agile approach at CS&A and the theory and best practices. Secondly, to understand the attitude towards working in this manner. Finally, to understand the linkages between stakeholders and team members, but also in the supporting system for working with agile. This was done in two steps. The first step was to observe the current way of working through observations of meetings and semi-structured inter-
views with key personnel behind the agile initiative. The second step was conducting semi-structured interviews to get all team members’ view of the way of working, as well as how they interact with each other.

2.5.5 Analyzing and Identifying Data to Create Recommendations

The data collection was divided into four parts: theory review, observations, interviews and surveys. These four parts together helped answer the research questions, see figure 1.

Theory and case studies were reviewed in order to find enablers, advantages and disadvantages, as well as to identify ways to work with agile. These ways to work could be in form of methods, techniques, etc. The surveys to professionals working with agile outside CS&A gave an understanding of the enablers for making agile work optimally, and the survey with CS&A gave a base for deeper investigation of the team members in the interviews. The interviews and observations contributed to understanding the current situation for working with agile at CS&A, more specifically what was good at the moment and what could be improved. Furthermore, it gave a good picture of the understanding of AM in CS&A.

![Figure 1. The Methodology of the Master Thesis Project](image)
2.6 Credibility

Measuring validity and reliability is common when deciding on the credibility of a study (Lekvall & Wahlbin 2001). In qualitative research, more weight is put on transferability of the study than on generalizability (Golafshani 2003). The validity, reliability and transferability will therefore be discussed further in detail in this chapter.

2.6.1 Validity

Validity treats whether the study method measures what it is intended to measure (Lekvall & Wahlbin 2001). An example of this is if the respondent to a survey misinterpret the question and the answer therefore responds to something else. Measuring validity is difficult, as there is no way to compare the results given with the “true” results. Instead a subjective evaluation of the method is normally done (Lekvall & Wahlbin 2001).

The resulting recommendations are difficult to evaluate, as the time frame was too limited to test and evaluate the results of the recommendations at CS&A. However, the method of the study can be scrutinized. The study is satisfactorily valid as there was a wide range of different sources used leading up to the final results. These sources were both primary, such as interviews, observations and surveys, and secondary such as articles.

There is a risk that observing and visiting the same team for several months could have biased the authors. This could have influenced them to have the same perspective as the case team and not be objective enough. However, this risk is assessed to be low, as the authors did not have access to the office space, resulting in conducting the vast majority of the study off-site. Also, the research conducted on other companies helped the authors stay objective as they got first hand input from other organizations. Finally, as the semi-structured interviews were conducted quite early in the Master Thesis Project, there was less time to get a biased view before conducting the interviews. The observations were complementary to the interviews and served as confirmation for the information gathered in the interviews.
The interview guides and surveys were both carefully reviewed by the authors, the supervisor at Faculty of Engineering at Lund University, and objective parties outside the Master Thesis Project. After conducting the interviews, the notes were sent to the interviewees to examine and correct them to find any misunderstandings in the interviews. All semi-structured interviews were also recorded to be able to go back and listen if there would be any confusion after the interview. To further ensure the validity of the Master Thesis Project, regular meetings were held with both the supervisor at Klarna and at the university. Also, the report was sent to them regularly for reviewing.

2.6.2 Reliability

Reliability handles if the measurement method in the study is affected by outside impact. A good reliability means that the study could be done multiple times in different settings, while still giving the same or similar results (Lekvall & Wahlbin 2001).

The sampling of the interviewees and the respondents to the survey could have affected the results of the study. As mentioned in section 2.3.2, all members of CS&A were interviewed resulting in the reliability being strong.

The reliability of the survey is weaker. This is because the authors used convenience and snowball sampling. If the respondents had been different, for example more heavily skewed to the non-software development industries, the findings of the survey might have differed. The reliability of the survey was made as strong as possible through screening the Internet for different communities and websites interested in agile so that the survey could reach many people with different backgrounds. However, this might be a problem as only the truly interested answered the survey. As the interviews were analyzed in such a diligent way and that both the authors were part of the analyzing, the findings from them is assumed to be reliable. The interviews were conducted in Swedish and then translated into English, which increased the risk of misinterpretations from the authors.

Furthermore, the surveys were sent out before the interviews, but analyzed after, except for the surveys going out to CS&A. This means that the result of the survey did not inflict on the interviews and when the interviews were done, the results of the survey
were already in. This means that the results of the two different methods inflicted on each other minimally. Further, a big part of the observations were done before the interviews, which decreased the risk of biases from knowing the team’s opinions when performing the observations. The results might have differed if all theory was gathered before the study of the case team. However, the study would not have had the same value as the search for relevant information could be narrowed after the case study.

When conducting the literature review, both authors scanned for interesting literature and got input from their supervisor at the university. This means that the sources came from different directions, which gave a wide, decoupled, first impression of the subject. However, one of the key references that was used, Conforto et al. (2014), was a previous literature review on AM outside software. If this article can be found have low reliability, the literature review would have lower reliability than initially aimed for. However, this is not assumed to be the case as the head author is a PhD at MIT, which is a highly renowned university. Also, it was published in Project Management Journal, which is a peer-reviewed academic publication (Project Management Institute, Inc 2016).

As some of the information gathered on the Internet came from websites and blogs, there is a risk that changes have been done to these after the information was gathered, and that they overall have a lower reliability as they are not reviewed. However, gathering information from these sources was necessary as the subject is quite young and there is not a sufficient amount of data in academic literature.

2.6.3 Transferability

As the Master Thesis Project revolves around a qualitative case study, it is of bigger interest to be able to transfer the research than to be able to generalize the results (Golafshani 2003). Transferability is of special interest in case studies, as they often are rich in detail but hard to generalize. A study being transferable means that the same study can be executed once again in a similar setting somewhere else (Colorado State University 2016). In order to make the study transferable, the authors provided a detailed methodology on every step of the research, a detailed description of the context of the case study company, as well as the interview guides and survey. The authors also made sure to collect data from both inside and outside the company. This can be seen in figure 2.
The part of the Master Thesis Project that can be argued to be generalizable is the SFs as they are not based on the case study but rather a literature review and a survey.

Figure 2. A visualization of the data collected in the Master Thesis Project.
3 Theory

This chapter introduces AM and TM by presenting the history, methods, practices, and when to use the different methods. There is also a comparison between the two management methods.

3.1 TM

3.1.1 What is TM?

The TM-method is also called the heavy method when referring to software development (Javanmard & Alian 2015). Management in general can be roughly divided into project- and process management. In contrast to projects, ongoing work is generally described as a process (Project Management Institute 2008). Figure 3 describes how the flow of work looks like in TM.

![Workflow in TM](image)

*Figure 3. The workflow in TM, figure adapted from Lotz (2013).*

**Project Management**

A project is temporary work undertaken to create a result. The temporary feature of a project indicates a definite beginning and end. Project management is the application of skills and knowledge to a project in order to achieve the wanted results (Project Management Institute 2008). In traditional project management, almost everything is
planned before the project starts. The tasks are performed in sequence after each other. Once the period where a task is performed is over, there will not be any revisions of that work. Therefore, the most commonly used method within traditional project management is the waterfall method, which symbolizes the inability to move back up the waterfall. Due to the nature of the method, the circumstances should not change considerably during the execution of a project. All outcomes of the project are delivered at the same time, which is called the “Big Bang” (Hass 2007).

In TM, the biggest focus is often to finish the project on time and not exceed the predefined budget. This means that the team might neglect the underlying reason of the project as well as what would constitute success. Furthermore, there is a risk of losing focus on the total cost of ownership and instead merely focus on the cost of the specific project. However, a positive effect of using TM is that it becomes important to understand the requirements before starting with the project, and thereby avoid unnecessary iterations with non-value adding activities (Hass 2007).

Process Management

From the beginning, the management was product-oriented and only focused on the end product. An increased focus has been placed on the entire process leading up to the end product (Bawden & Zuber-Skerritt 2002). This means measuring and optimizing the event flow (Palmer n.d.), and not the outcomes, and is therefore more about handling relationships than something physical (Bawden & Zuber-Skerritt 2002). These activities follow each other with the aim of creating a viable product to the end customer or meeting some other business objective (Palmer n.d.). Further, it is about letting a group find what their goals should be and how to reach those goals. Overall process management is about leading a group to find consensus regarding desired outcomes and thereby create structure (Bawden & Zuber-Skerritt 2002).

ISO 9001 defines a process as “An activity using resources, and managed in order to enable the transformation of inputs into outputs, can be considered as a process” (ISO 2000).
3.2 AM

The authors of the Master Thesis Report treat agile process management and agile project management as the same phenomenon since agile work is performed in iterations. This means that it is not assumed that there are any big differences between an ongoing process and a project with a clearly defined beginning- and end-state in the daily management of the agile work.

3.2.1 What is AM?

AM partly comes from the principles of ASD. By definition, ASD is a term including different methods and practices that has sprung from the agile manifesto (What is Agile? n.d.). The agile manifesto has four ground pillars. Firstly, it stresses the importance of valuing individuals and interactions rather than the processes and tools. Secondly, a working software is more important than a comprehensive documentation. Thirdly, collaboration with the customers outweighs the importance of contract negotiation. Lastly, responding to change is prioritized rather than following a predefined plan. Except for the ground pillars, there are also 12 principles of the agile manifesto (Fowler & Highsmith 2001):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.

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

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

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

In AM, the teams are generally self-organizing and cross-functional (Chang 2010). Agility is defined as “... the ability to both create and respond to change in order to profit in a turbulent business environment” (Highsmith 2004). Thus, when the external environment is uncertain, success comes from creating own change, through innovation, to create an even more uncertain environment for the competitors. Because of this, the own company also needs to be able to react quickly to external changes from competitors and customers. This ever-changing environment needs other, more flexible, management methods than TM (Highsmith 2004). Figure 4 visualizes what the workflow looks like in AM, as opposed to figure 2 showing the same process with TM.

![Figure 4. The workflow in AM, figure adapted from Lotz (2013).](image)

Some people believe that working with agile is synonymous with lack of structure (Highsmith 2004). Highsmith (2004) means that too much structure implicates a rigidity that makes it hard for the company to be flexible. Therefore, it is beneficial to balance the optimal amount of flexibility and stability to manage the company in the best way in
a fast-changing environment. To elaborate further, it is not beneficial in this kind of environment to strive for standardized and repeatable processes, which means a precise and structured process, as the needs are ever-changing. The value rather lies in having a flexible and moderately structured process that can be changed when the needs change (Highsmith 2004).

Because of the dynamic environment, the leader of the team should not be a taskmaster. This means that he or she should not give detailed instructions on how to solve tasks, but rather create collaborative relationships in the team and set ground rules for the team to follow (Hass 2007). This is supported by the fifth principle of the agile manifesto (Fowler & Highsmith 2001).

Roles and responsibilities are important when working with AM. The role means what you do, and the responsibility is what you have authority to decide. The roles are not defined by the employee's title, as in TM, but rather the employee's area of expertise. This is because the agile environment tends to be fast-changing and the team members need to contribute where they can. The roles create boundaries, in which a role is defined. However, the boundaries are not static but can be pushed through with communication and collaboration. They can be seen as guidelines to work in the right direction. As the environment is fast-changing, one need to be prepared and adjust the boundaries thereafter. This means that some roles might be eliminated or reformed into new ones. It is important that the roles are outspoken and documented in order to eliminate misunderstandings (Chin 2004).

Nicholls, Lewis & Eschenbach (2015) mean that the tasks should be small and the communication intervals short in order to maximize the use of the team members' knowledge and insights. This in turn gives the team quick wins, which improves its morale (Nicholls, Lewis & Eschenbach 2015). Also, Nicholls, Lewis & Eschenbach (2015) have benefitted from not working on the same material at the same time except from when preparing for meetings. A big part of collaborating within agile, however, is facilitated by the agile practices themselves, such as the planning meetings, stand-ups, and retrospectives. They bring the team together and make them communicate with each
other, which make barriers between the team members vanish. Relationships between the team members are also built up by these activities (Mahale 2011).

As principle number six of the agile manifesto states, the most efficient and effective way to communicate is face-to-face (Fowler & Highsmith 2001). Therefore, the team should be collocated. However, ideally this would also mean that the customers and end-users would be placed in the same room as the team (Hass 2007).

3.2.2 History

The first step towards AM was made in 1968 in Conway’s Law, which constitutes that when an organization designs a system it will subconsciously be a copy of the organization’s structure for communication. In 1976, Panzl described tools that enabled automated software testing. Further, the first steps towards incremental development were taken in 1980 at IBM’s Federal System Division. In 1984, Barry Boehm examined projects using prototyping, which is an iterative strategy, giving the practice more attention. Also, alternatives to the partly criticized waterfall approach became crisper in their nature in 1984. The year after, the first named alternative, Rom Gilb’s Evolutionary Delivery Model, was created (Practices Timeline n.d.).

The following years, in the end of the 1980’s and beginning of the 1990’s, much happened in the field with new methodologies in both testing and developing. One of these events was when Jim Coplien stressed the productivity benefits of daily meetings, in an article in 1994. It has later been described as one of the main influences of Scrum. Later, in 1995, the sprint was formulated as an iteration within Scrum. In 1998, daily stand-up and continuous integration were defined as core practices within Extreme Programming (XP), a new term within agile at the time (Practices Timeline n.d.)

ASD was first introduced in February 2001. This was at a meeting with 17 software developers who created and signed the Agile Manifesto. Short after, the Agile Alliance was formed in order to continue the work of developing and preaching ASD (What is Agile? n.d.).
Even though agile is an old technique within software development; there are few case studies on it outside the industry. There are tendencies indicating that AM is easiest to implement in industries similar to software development, such as product development. Both these industries can gain from using similar tools, such as prototyping (Conforto et al. 2014).

3.2.3 Agile Methods

Scrum

Scrum is an agile project management framework with broad applicability regarding management and control of flexible and iterative projects. Lately, Scrum has gained increased popularity in the ASD-community due to its simplicity and proven productivity (McLaughlin n.d.).

When using Scrum methodologies, the product owner and the team work closely together in order to create the product backlog. It contains features, bug fixes, and other tasks that should be resolved in order to deliver a working software system. The product owner drives the prioritization, while the cross-functional teams estimate and execute the different tasks during a sprint. After a sprint, the product backlog is reorganized and reprioritized, if necessary. During a sprint, only the team itself can add tickets to the product backlog (McLaughlin n.d.). However, there is a philosophy that a team using Scrum should strive towards not making changes to the backlog during a sprint (A Brief Introduction to Kanban n.d.).

Roles within Scrum

There are three defined roles within the Scrum methodology. These are the developing team, the Scrum master, and the product owner (Sutherland & Schwaber 2013).

The product owner’s role is to maximize the value of the product and the development team’s work by expressing clear backlog items, ordering the items in the backlog, optimizing the value of the work the team performs, ensuring that the backlog is visible, ensuring that the team understands items in the backlog, etc. The product owner is one person, but can have the team do the work while still remaining accountable. The team
consists of professionals who do the work of delivering in each sprint (Sutherland & Schwaber 2013).

The Scrum master’s primary role is to be the interface between the product owner and the development team. The Scrum master is responsible for daily coordination meetings, as well as facilitating the development process and ensuring that the team uses the full range of appropriate agile values, practices, and rules (Bass 2014). He or she is also responsible for ensuring that Scrum is understood within the team. The Scrum master serves both the team and the product owner in different ways. He or she serves the product owner through finding techniques for effective backlog management, helping the Scrum team understand the need for clear and concise backlog items, ensuring that the product owner knows how to arrange the backlog to maximize value, facilitating Scrum events as requested or needed, etc. The Scrum master serves the development team by coaching them in self-organization and cross-functionality, helping them create high-value products, removing impediments to the team’s progress, etc. (Sutherland & Schwaber 2013).

**Rules within Scrum**

The Scrum master is responsible for the rules being followed. If the team wants to change the rules, this can be done at the sprint retrospectives. It is important that rule changes originate from the team, and not management, and should only be approved by the Scrum master if sufficient knowledge and understanding of agile is present within the team. The rules within Scrum can be divided into: sprint planning meeting, daily Scrum meeting, sprint, sprint review meeting, and sprint retrospective meeting (Schwaber 2004). Here follows the rules as suggested by one of the co-founders of Scrum, Schwaber (2004).

**Sprint Planning Meeting**

The sprint planning meeting is time-boxed to eight hours divided into two four-hour segments. The first segment is for selecting the product backlog and the second for preparing the sprint backlog. The attendees at the meeting are the Scrum master, the product owner, and the development team. Other parties can be involved to provide specific information at the meeting. The product owner must prepare the product backlog prior to the meeting. The goal of the first segment is to choose the functionality that the team
believes they can commit to. This functionality will be demonstrated at the sprint re-
view meeting. The team can make suggestions, but the product owner is responsible for
what product backlog makes up the sprint. The team is then responsible for deciding
how much of the work they will attempt to do during the sprint. The second segment
occurs right after the first. The team is solely responsible for figuring out how to turn
the selected product backlog into an increment of shippable product functionality. The
product owner must be present to answer questions from the team, but is not allowed
to do anything but observe and answer questions. The output of the second segment is a
list, called the sprint backlog, including assignments and estimates for the team to work
on during the sprint (Schwaber 2004).

*Daily Scrum Meeting*

The daily Scrum meeting is time-boxed to 15 minutes and should be held in the same
place every time. The daily Scrum meeting is best held first thing in the morning so team
members have a chance to reflect on what they did yesterday and what they are going
to do today. All team members are required to attend the meeting and in case of absence
the team member should attend by phone or by updating another team member about
the status of his or her work (Schwaber 2004). Schwaber (2004) points out that it is
important that the Scrum master starts the meeting at the appointed time and suggests
that anyone who is late should pay one dollar to the Scrum master immediately. Each
team member should answer the following three questions (Schwaber 2004):

- What have you done since the last daily Scrum meeting regarding this project?
- What will you do between now and the next daily Scrum meeting regarding this
  project?
- What impedes you from performing your work as effectively as possible?

It is important that the team members stick to these questions only, and it is up to the
Scrum master to make sure that the meeting is efficient (Schwaber 2004).

*Sprint*

The sprint is time-boxed to 30 consecutive days. The team can seek outside advice, sup-
port, or information during the sprint, but no one can provide directions or information
without the team’s enquire. The product backlog is frozen during the sprint and cannot
be changed until the end. If the team realizes that they cannot finish all tasks during a sprint, or that they have more time and could commit to more tasks, they can consult the product owner as to which tasks to remove or add to the sprint. The team has two administrative tasks during the sprint: to attend daily Scrum meetings and to keep the product backlog up to date (Schwaber 2004). It is important that a team delivers what they promise within the sprint. When planning a sprint, there should therefore be enough room for ad hoc work without falling behind on planned backlog tasks (Ackerson & Marschall n.d.).

*Sprint Review Meeting*

The sprint review meeting is time-boxed to four hours and the team should not spend more than one hour to prepare for the meeting. The purpose of the meeting is to present the functionality created during a sprint for the product owner and stakeholders. The sprint review starts with a team member presenting the sprint goal, the product backlog committed to, and the product backlog completed. The team can then discuss what went well and what was unsatisfactory during the sprint. After presentations of the functionality, every stakeholder is polled to get their impressions, any desired changes, and the priority of these changes (Schwaber 2004).

*Sprint Retrospective Meeting*

The sprint retrospective meeting is time-boxed to three hours. This meeting is attended by the team, the Scrum master, and the product owner. The product owner's presence, however, is not mandatory. The meeting starts with all team members answering two questions:

- What went well during the sprint?
- What could be improved in the next sprint?

The Scrum master writes down all the answers and the team prioritizes in which order to discuss the potential improvements (Schwaber 2004).

*Lean Software Development*

Lean Software Development (LSD) is an iterative agile methodology that was originally developed by Mary and Tom Poppendieck. LSD sprung from the Lean Enterprise move-
Agile Management Outside of Software Development

ment and practices from companies like Toyota and focuses on delivering value to the customer (McLaughlin n.d.). According to McLaughlin (n.d.), the main principles behind the methodology are:

- Eliminating waste
- Amplifying learning
- Deciding as late as possible
- Delivering as fast as possible
- Empowering the team
- Building integrity
- Seeing the whole

Lean methodology eliminates waste through focusing only on the most valuable features for a system and relies on rapid and reliable feedback between programmers and customers. The method focuses on authority of individuals and small teams, rather than a hierarchical flow. LSD also strives to create productivity by using team resources as efficiently as possible (McLaughlin n.d.).

**Kanban**

Kanban is based on three basic principles. The first principle is to visualize what the team is doing and being able to see all items in context of each other. The second principle is that the amount of Work In Progress (WIP) should be limited as it helps to balance the flow-based approach, so that teams do not commit to too much work at once. The third, and last, principle is to enhance flow by starting to work with the highest priority in the product backlog after a task is finished. Kanban aims at promoting continuous collaboration, improving workflow, and ongoing learning (McLaughlin n.d.). Compared to Scrum, within Kanban there are no time limited sprints but a continuous flow of work. There is also a philosophy that change can happen at any time (A Brief Introduction to Kanban n.d.).

**XP**

XP was originally created by Kent Beck and has become one of the most controversial and popular agile methodologies. The aim of XP is to deliver high-quality software quickly by promoting high customer involvement, rapid feedback loops, continuous
Agile Management Outside of Software Development

testing, planning, and close teamwork in frequent intervals, typically one to three weeks. XP was originally based on four simple values: simplicity, communication, feedback, and courage (McLaughlin n.d.). It also has twelve supporting practices, according to McLaughlin (n.d.).

- Planning game
- Small releases
- Customer acceptance tests
- Simple design
- Pair programming
- Test-driven development
- Refactoring
- Continuous integration
- Collective code ownership
- Coding standards
- Metaphor
- Sustainable pace

In XP, the customer works closely with the development team to create user stories, which are granular units of functionality. The team estimates, plans, and delivers the highest priority user stories (McLaughlin n.d.).

Crystal

The Crystal method was developed by Alistair Cockburn and contains a family of methods: Crystal clear, Crystal yellow, Crystal orange, and Crystal red. Which method to be used is decided based on the size of the project, measured in number of people, and risk mitigation. The method is about enhancing the work of the people, reducing paperwork to the absolute minimum, and starting out with a prototype smaller than needed to be able to grow it to the right size (Chang 2010). Chang (2010) describes seven properties within Crystal that a project should have in order to become successful.

- Frequent deliveries to stakeholders
- Continuous feedback and reflective improvement
Agile Management Outside of Software Development

- Constant communication with collocation
- Personal safety
- Team members being able to focus without interruption
- Easy access to users, stakeholders and experts
- A technical environment

The three first are mandatory, while the latter four are desirable but not necessary. The fourth property is both about the team members being comfortable to communicate openly and the safety of end-users. The last property is about having a technical environment advanced enough to be able to handle activities such as versioning and automated testing (Chang 2010).

*Dynamic Systems Development Method*

Dynamic Systems Development Method (DSDM) grew out of the need to find an industry standard framework for a method called Rapid Application Development. DSDM evolved from business needs, active user involvement, empowered teams, frequent delivery, integrated testing, and stakeholder collaboration. The method promotes the possibility that 80% of the system can be deployed in 20% of the time. Requirements are delivered in short iterations and are prioritized using the MoSCoW rules, specifying whether they are must haves, should haves, could haves, or will not have at this time but potentially later. The DSDM framework is independent and can be used together with other methodologies, such as XP (McLaughlin n.d.).

*Feature-Driven Development*

Feature-Driven Development (FDD) was originally developed by Jeff De Luca. The method is a model-driven, short-iteration process. It starts out with deciding the general scope and then continues with a series of two-week sprints (McLaughlin n.d.). FDD mainly uses the following eight practices, according to McLaughlin (n.d.):

- Domain object modeling
- Developing by feature
- Component/class ownership
- Feature teams
- Inspections
Agile Management Outside of Software Development

- Configuration management
- Regular builds
- Visibility of the progress and results

Proponents of the FDD methodology claim that it is more scalable than other agile approaches, and is therefore better suited for larger teams. Unlike other agile approaches, the method describes short and specific phases of work that should be performed separately for each feature in the software system (McLaughlin n.d.).

Comparison of Methods

The methodologies within agile are quite similar to each other. However, there are some differences between them. In table 1, the pros and cons with Scrum, LSD, Kanban, XP, Crystal, DSDM, and FDD are highlighted according to OPS International LLC (2015).

<table>
<thead>
<tr>
<th></th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| Scrum  | - Maximizes personal communication and informal knowledge sharing.  
       | - Breaks project into manageable pieces.  
       | - Progress is made even if requirements are not stable. | - Weak on measurement practices.  
       | - Weak on business system.  
       | - Technical, and concept development practices.
| LSD    | - Seeks to change companies from the top down.  
       | - Guidelines for business enterprise very defined. | - Allows little change in requirements.  
       | - Literature for applying in a software environment is limited.  
       | - Poorly defined technical and measurement practice. |
| Kanban | - Allows teams to visualize their work and eliminate bottlenecks.  
       | - Can lead to exponential improvements in operational efficiency and quality. | - Using JIT delivery will inevitably lead to delays at some point when JIT turns into "Not In Time".  
       |                                                                                      | - Does not capture or show dependent tasks very well; hard to see overall project status. |
| XP     | - Simple.  
       | - Iterative.  
       | - Values communication.  
       | - Based on best practices.  
       | - Puts emphasis on design. | - Lacks design documentation.  
       | - Highly prescriptive.  
       | - Lacks measurement processes.  
       | - Does not address deployment. |
| Crystal Family | - Strong on communication.  
               | - Well-defined guidelines for project teams. | - Largely theoretical.  
               |                                                                                      | - Does not define guidelines for the business |

Table 1. Pros and cons with seven different agile methodologies, table adapted from OPS International LLC (2015).
### 3.2.4 Agile Support Systems

**Jira**

Jira is a software development tool offered by the company Atlassian. Jira can plan, track, and release software. In the system, it is possible to create user stories and issues, plan sprints, and distribute tasks across a team. It is also possible to prioritize and discuss the team’s work with complete visibility. Jira combines important elements of agile development, such as flexible Scrum and Kanban boards, together with real-time reporting. With options for sorting, color coding, etc., it is possible for a team to customize the system for their own use (Jira Software n.d.).

**Confluence**

Confluence is a software tool developed by the company Atlassian. The system aims to create, organize, and discuss work within a team. Confluence can work within any field and is therefore not specific to software development. In Confluence, it is possible to create meeting notes, file lists, financial forecasts, product requirements, etc. It is also possible to give feedback on any Confluence page or file with comments. Atlassian recommends that a team uses Confluence together with Jira in order to create transparency in projects (Confluence n.d.).

### 3.2.5 Agile outside Software Development

The challenges that exist when using agile outside software development are similar to the ones that generally exist within software development, such as long stand-ups or large tasks that do not progress (Davies 2012). There are some factors, however, that are different when applying agile methodologies outside software development according to Davies (2012). One difference can be that it is more difficult to share the same
goal outside software. Another challenge can be that team members are not used to being told what to do or are uncomfortable with increased visibility of their work. In software, most teams consist of engineering graduates, while outside there might be more varied experience levels. Work outside software development might also be fluid or constrained by lead times and dependencies on other groups (Davies 2012). Davies (2012) brings up a few ways that can start the conversation with teams that do not use agile:

- Explain the value of agile over a coffee or a beer
- Introducing facilitation as a way to make meetings more fun
- Executive support
- Arranging a tour of software development teams using agile
- Finding one open and curious person in the team
- Organizing a team reliance - involving all hands as supporters
- Making an agile coach available to the team
- Crafting an interesting elevator pitch for agile
- Creating interest for agile with visualized tools

### 3.3 Comparison between AM and TM

Agile and traditional (heavy) methodologies both have their strengths and weaknesses. However, teams normally use a customized methodology that might be a mix of the two. Table 2 shows the main differences between agile methods and heavy methods (Awad 2005).

<table>
<thead>
<tr>
<th></th>
<th>Agile methods</th>
<th>Heavy methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Approach</strong></td>
<td>Adaptive</td>
<td>Predictive</td>
</tr>
<tr>
<td><strong>Success measurement</strong></td>
<td>Business value</td>
<td>Conformation to plan</td>
</tr>
<tr>
<td><strong>Projects size</strong></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td><strong>Perspective to change</strong></td>
<td>Change adaptability</td>
<td>Change sustainability</td>
</tr>
<tr>
<td><strong>Culture</strong></td>
<td>Leadership-collaboration</td>
<td>Command-control</td>
</tr>
<tr>
<td><strong>Documentation</strong></td>
<td>Low</td>
<td>Heavy</td>
</tr>
<tr>
<td><strong>Emphasis</strong></td>
<td>People-oriented</td>
<td>Process-oriented</td>
</tr>
</tbody>
</table>

*Table 2. A comparison between agile and heavy methods based on different factors, table adapted from Awad (2005).*
When deciding on what method to use, there are three major factors affecting the decision: project size, people factors, and risk factors. A limitation to agile is project size. When a team is larger, or the bigger budget a project needs, the bigger the project size is. This creates more requirements and need of coordination. Heavyweight methodologies support this by providing plans, documentation, and processes for better communication (Awad 2005). A large part of the agile manifesto’s values deal with human factors (Fowler & Highsmith 2001).

If an organization is not responsive to change and has many rules and procedures, an agile model will not be successful. When dealing with risk factors, agile is used in applications that can be built quickly and do not require extensive quality assurance. When a system is more critical, reliable, and safe, heavyweight methods are more suited. The more a project differs from the conditions in table 3, the more risk will be associated with using one of the two methodologies (Awad 2005).

<table>
<thead>
<tr>
<th></th>
<th>Agile discriminator</th>
<th>Heavyweight discriminator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary objective</strong></td>
<td>Rapid value</td>
<td>High assurance</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>Largely emergent, rapid changes, unknown</td>
<td>Knowable early, largely stable</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>Smaller teams and projects</td>
<td>Larger teams and projects</td>
</tr>
<tr>
<td><strong>Architecture</strong></td>
<td>Designed for current requirements</td>
<td>Designed for current and foreseeable requirements</td>
</tr>
<tr>
<td><strong>Planning and control</strong></td>
<td>Internalized plans, qualitative control</td>
<td>Documented plans, quantitative control</td>
</tr>
<tr>
<td><strong>Customers</strong></td>
<td>Dedicated, knowledgeable, collaborated, collocated on-site customers</td>
<td>As needed customer interactions, focused on contract provisions</td>
</tr>
<tr>
<td><strong>Developers</strong></td>
<td>Agile, knowledgeable, collocated, and collaborative</td>
<td>Plan-oriented; adequate skills access to external knowledge</td>
</tr>
<tr>
<td><strong>Refactoring</strong></td>
<td>Inexpensive</td>
<td>Expensive</td>
</tr>
<tr>
<td><strong>Risks</strong></td>
<td>Unknown risks, major impact</td>
<td>Well understood risks, minor impact</td>
</tr>
</tbody>
</table>

Table 3. Agile and heavyweight discriminators, table adapted from Awad (2005).
3.4 Enablers for Changing Management Methods

When it comes to implementing new management methods, Näslund (2013) means that the success of the methods often depend more on the approach the organization has towards the method, rather than specific factors of the method. Management support and culture are enablers that are seen as especially critical. The management plays a critical role as they set the goals, control resources, and decide on priorities. Some say that the management needs to directly participate and not just support the initiative. Furthermore, the management has a large impact of the culture (Näslund 2013). Näslund (2013) mentions three other important enablers that remain the same, independent of method:

- Strategic alignment
- Project management
- Training

Strategic alignment means that the change initiative should be aligned with the strategy of the organization and that there should be a long-term plan for the initiative. The initiative should be thought through and it is often good if it is based on experienced problems. A long-term view is important as it often takes more than three years, after the implementation of the new method, before the company experiences significant improvements. Something that can affect the long-term implementation is change in management, as the new management may not see the benefits of the new method (Näslund 2013).

The project management, for the change initiative, can be divided into: project selection/initiation, leadership/teams and approach, and monitoring/measuring. The chosen projects need to be aligned with the objectives of the organization and prioritized accordingly. The leadership must be competent and teams within an organization should be cross-functional. It is furthermore important to measure the success of a change initiative, both for the monitoring but also for the morale of the group. A part of this is connecting rewards for the employees to collaborating with the initiative (Näslund 2013).
Training is always important, and the question is rather in what way and how much training the initiative should include. Training is necessary for both management and team members. It is suggested that focus is placed on the most commonly used tools and supplement this on isolated trainings of tools that are used less frequently. However, some say that there is too large focus on tools and to little focus on the human factor. The training should be on an ongoing basis so that the initiative does not fade away as the time goes by (Näslund 2013).

3.4.1 Deciding Whether Agile is Right for a Team

AM suits teams that work in innovative and ever-changing environments outside software development (Highsmith 2004). Also, it works best in small teams up to eight people (Conforto et al. 2014). Theory mentions two important elements in order to decide if AM is appropriate for a business: practices and enablers. A practice is a management action that employs one or several techniques or tools in order to contribute to the execution of a process (Conforto et al. 2014). Conforto et al. (2014) discuss six practices within agile, see table 4.

<table>
<thead>
<tr>
<th>Aspect (practice)</th>
<th>Practice in agile teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project scope description</td>
<td>Minimal textual description</td>
</tr>
<tr>
<td>Tool used to communicate the project plan</td>
<td>Use of visual panels and boards, pictures, drawings, and so forth</td>
</tr>
<tr>
<td>Project planning approach</td>
<td>Developed by iteration</td>
</tr>
<tr>
<td>Project planning responsibility</td>
<td>Created collaboratively; shared responsibility</td>
</tr>
<tr>
<td>Project’s plan progress and updating responsibility</td>
<td>Shared responsibility</td>
</tr>
<tr>
<td>Project plan updating frequency</td>
<td>Weekly</td>
</tr>
</tbody>
</table>

An enabler is an internal or external factor that will influence the results of the implementation of AM, either directly or indirectly (Conforto et al. 2014). Conforto et al. (2014) have done a literature review on important enablers for agile and found 41 enablers in theory that were categorized into: organization, process, project team, and project type. The 41 enablers can be found in appendix 3. After identifying the 41 enablers, the most occurring ten were identified, see table 5 (Conforto et al. 2014). Conforto et al. (2014) found, from conducting a survey with 19 companies regarding the ten enablers, to what extent agile teams live up to the different factors.
Table 5. A description of the ten most important enablers for AM and what they look like in agile teams, table adapted from Conforto et al. (2014).

<table>
<thead>
<tr>
<th>Enabler</th>
<th>The enabler in agile teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational structure type</td>
<td>It is preferable if the organization is project oriented or strongly matrix oriented as opposed to balanced matrix oriented, and the worst case is if there is a weak matrix orientation.</td>
</tr>
<tr>
<td>Multidisciplinary project teams (various competences)</td>
<td>It is beneficial with cross-functional teams.</td>
</tr>
<tr>
<td>New product development process formalization level</td>
<td>The management should be partially standardized, which means that they have used a project management method before, even if it is not agile.</td>
</tr>
<tr>
<td>Customer/stakeholder involvement in the product development process</td>
<td>The customers and stakeholders should be involved, and influence, throughout the project.</td>
</tr>
<tr>
<td>Supplier or partner involvement in the product development process</td>
<td>The suppliers and partners should be involved, and influence, throughout the project.</td>
</tr>
<tr>
<td>Project team members’ experience (in years)</td>
<td>It is optimal when team members have at least two years of experience.</td>
</tr>
<tr>
<td>Project manager’s experience (in years)</td>
<td>It is optimal when the project manager has at least two years of experience.</td>
</tr>
<tr>
<td>Project team size (number of project team members)</td>
<td>The smaller the teams are, the better it is for AM. It is preferable to have teams of up to eight members.</td>
</tr>
<tr>
<td>Project team dedication (time dedicated exclusively to the project)</td>
<td>The amount of time allocated to the specific project should be as high as possible, preferably more than 75%.</td>
</tr>
<tr>
<td>Project team location</td>
<td>It is optimal with a team who sits close to each other, preferably in the same room.</td>
</tr>
</tbody>
</table>
4 Previous Case Studies

This chapter brings up some previous case studies made in the field of AM, both within, and outside of software development. The challenge, methodology implemented, results and conclusion from the different studies are presented.

4.1 Increased Productivity When Going All-in

To investigate how different agile methodology works compared to the traditional waterfall method, the software estimation company Quantitative Software Management Inc. (QSM) conducted a case study at a large technical business organization (Putnam 2014).

The Challenge

The company had tried to apply agile to its business since 2010 but the results were far from optimal. They lacked the infrastructure needed and organizational mind shift necessary to fully benefit from the agile principles. In 2011, the company made a second attempt, but with more resources. This time there were more organizational support and buy-in from senior management and key stakeholders. Part of the process was to conduct an assessment of development systems using tools and methods that would support agile (Putnam 2014).

Results

QSM investigated the productivity of the company before and after implementing agile. Productivity was measured in index points, a calculated proprietary QSM unit that ranges from 0.1 to 40. Projects using waterfall methods increase their average productivity with 1.5 to 2 index points per year. In 2010, after the first implementation, the productivity using agile was less successful than the traditional waterfall method. During the second attempt, the productivity increased dramatically with 7.5 index points while surpassing the average productivity using waterfall methods, see figure 5 (Putnam 2014).
Another finding was that the overlap between high-level design and construction phases was much greater, 97%, for agile methods compared to waterfall methods where the overlap was 30%. This was due to the iterative approach to planning and delivery that the agile model has. It was also apparent in the case study that the slope of the learning curve was steeper for agile than for waterfall. This was because the teams had to learn a whole new methodology of defining requirements, writing code, and concurrent testing. When agile was first implemented at the case company in 2010, the projects using waterfall methods delivered 58% faster with 74% less effort. In 2011, there was a shift making agile methods 34% faster with 27% less effort (Putnam 2014).

In order to investigate when to use agile, and when to use waterfall, the case study normalized the project sizes into a common standard called Implementation Units (IU). They found that if a project is larger than 12,000 IU, it is more beneficial to use agile than waterfall in terms of time to market, cost, and productivity. If a project is smaller than 12,000 IU, a waterfall method is more beneficial. It should be mentioned that the results are only based on one organization and that there are other studies that present agile as a beneficial method for smaller projects (Putnam 2014).

**Conclusion from Case Study**

Putnam (2014) suggests that when deciding between agile and waterfall, the decision should be based on the method best suited for the project’s intended environment. This
case study serves to show that agile can be extremely beneficial if implemented with the right factors considered. If adopted without appropriate cultural modifications and organizational support, agile has the potential to negatively impact an organization’s productivity (Putnam 2014).

Developing a new method is a strategic investment rather than a quick-fix. Before implementing a new model there should be time invested on all levels of the organization to fully understand the task and assess how long it will take before the wanted results show. As in this case study, informed and deliberate decisions will lead to better outcomes (Putnam 2014).

4.2 Agile in Library IT Innovations

To investigate how an IT-team at a library could benefit from using agile, the UMBC Library IT Services unit implemented an agile model for the development of new and innovative software. The team was also responsible for the management of computing resources and all technology of the library. Before the implementation, the team consisted of three full time employees and five student assistants (Chang 2010).

The Challenge

IT is changing in a fast pace, much faster than the library could keep up with due to lack of staff and financial resources. The ambition of implementing agile was to be able to achieve better project management and software development in the innovative IT-projects and to enable fast-tracking implementation of innovative projects. Furthermore, the hope was to energize the staff, as they would become more active in the development process (Chang 2010).

Methodology Implemented

The agile method used was the Crystal clear method. University student assistants, who had a various set of skills and an open attitude towards the new methodology, were recruited to be part of the agile team (Chang 2010). Chang (2010) found that an advantage with hiring student assistants is that they have a tendency to be more innovative in their thinking as well as being more familiar with new IT-trends.
There were sometimes multiple teams working on the same projects, mainly based on the students’ availability. The teams were customized to each project, according to the needed size and time. Further, the staff programmer could seldom be included in any projects because of other constraints. In order to enable coordination and communication, there was a schedule to ensure that students were working at the same time. This also meant that the other staff could work with administration that was not included in the project when the students were not working. The iterations had the length of one to two weeks. Milestones and deadlines were used as well as a task board, static cling dry erase sheets, and mind maps. The area around the task board became the meeting area for stand-up meetings (Chang 2010). Table 6 describes how Chang (2010) used the tools in the case study.

Table 6. The tools used in the study of Chang (2010).

<table>
<thead>
<tr>
<th>Tools</th>
<th>The Wanted Results</th>
<th>How they did it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task board</td>
<td>Displaying the project’s status and increasing the accessibility to the entire team.</td>
<td>Core requirements and project features were written down on sticky notes. On even smaller notes the team members’ names were written and posted on the sticky notes.</td>
</tr>
<tr>
<td>Static cling dry erase sheet</td>
<td>Written down notes and ideas.</td>
<td>The entire wall of the office was covered by the board. Due to a whiteboard being limited in size, this enabled to fit more notes. Each workstation had its own sheet.</td>
</tr>
<tr>
<td>Mind map/Content map</td>
<td>Visualizing of the project, which enables to plan the upcoming activities better. Also, helping in the brainstorming of ideas, because the team members do not have to tend to a specific order of importance.</td>
<td>Supplement to the task board on project level. This was a non-linear description, a tree structure, of ideas, tasks, etc. Each node represented a project that could be expanded for further detail.</td>
</tr>
</tbody>
</table>

Results

One of the outcomes was the possibility to reduce the number of meetings. Furthermore, it was found that the three first properties of Crystal were crucial, while the latter four were not necessary, but still desired as they created a better work environment (Chang 2010). Table 7 shows Chang’s (2010) experiences on what the different properties enabled. The seventh property was not included as there was not sufficient data about it in the case study.
<table>
<thead>
<tr>
<th>Property</th>
<th>How they met it</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent deliveries to stakeholders</td>
<td>There were some direct communication between the programmers and stakeholders.</td>
<td>The programmers could do quick changes and fine-tune aspects that the stakeholders wished for.</td>
</tr>
<tr>
<td>Continuous feedback and reflective improvement</td>
<td>The team took time internally to discuss each bigger iteration.</td>
<td>Improved the members’ confidence as well as their leadership capabilities</td>
</tr>
<tr>
<td>Constant communication with collocation</td>
<td>Moving to a larger office with more than two workstations.</td>
<td>Gave faster and direct communication as well as created a learning environment. Furthermore, it increased creativity.</td>
</tr>
<tr>
<td>Personal safety</td>
<td>Moving to a larger office with more than two workstations.</td>
<td>Created a safe ambiance for the team and increased the community feeling.</td>
</tr>
<tr>
<td>Team-members being able to focus without interruption</td>
<td>They moved to a new office that was more secluded.</td>
<td>It gave increased possibility to focus.</td>
</tr>
<tr>
<td>Easy access to users, stakeholders and experts</td>
<td>The stakeholders were often unavailable.</td>
<td>This meant possible delays in the development.</td>
</tr>
</tbody>
</table>

It took about three projects before the IT-team had adapted to the agile methodology, but had also been able to adapt the methodology to suit them. The applications created in the projects were even used by other organizations than the library, showing that the agile implementation was successful. Part of the reason for success came from the participants being flexible and excited about working in this new fashion (Chang 2010).

Chang (2010) found that one of the most important challenges was to find appropriate staff interested in change and creativity. It was found that around 60% of the programmers were stimulated by the way of working, while the others found a lack of structure and left. Also, the stakeholders were sometimes difficult to get a hold of and preferred conducting formal meetings with the entire development team instead of informal short communication with the various members, which could result in delays for the projects. The stakeholders sometimes had bigger needs for modifications, during the iterations, than the resources of the team permitted. This could be solved by rolling out beta-versions that later could be modified (Chang 2010).
**Conclusion from Case Study**

Agility suits creative teams and innovative environments, especially teams aiming for disruptive innovation. Crystal methodologies facilitated handling of the variety of IT-projects that the organization stood before. The success could largely be accounted for due to the flexibility and creativeness of the team members as well as the nature of the projects, as they were interesting and have high learning value (Chang 2010).

**4.3 From Scrum to Kanban at Stormpath**

Stormpath is a provider of user management and authentication services for development teams (About Stormpath n.d.).

**The Challenge**

While the team at Stormpath enjoyed agile, they used a methodology that was not working. This methodology was Scrum, which was too rigid, created too much overhead, and started to cause burnout to the entire team. Being a start-up company, they were operating in a fast-changing environment with a limited amount of resources. Scrum made everyone feel involved and helped with coordination, but this was not a weak point for the team. The sprint planning took half a day and forced engineers, which were focused on one component, to sit through discussions about other unrelated components (Salazar 2014).

Like most teams, they had some team members that would overestimate tasks and some that would underestimate them. The team felt that the time put on trying to estimate tasks at a sprint planning was wasted since they were probably not right anyway (Salazar 2014). Salazar (2014) elaborates further that the time estimation is redundant; if something is a priority, it does not matter how long it takes to handle, the task still needs to be finished. The unsatisfactory estimations made the workload for the team uneven since they would often underestimate a task, which led to unrealistic expectations. This would also contribute to uneven hours at the office where the team was stressed out at the end of a sprint, as they needed to finish something they had underestimated (Salazar 2014).
The fact that they were operating in a fast paced environment was also complicated since any changes that needed to be done to the sprint implicated extra meetings, resulting in even more overhead. Shortening a sprint could increase the productivity, but the overhead would limit the amount of work that could get done in a short sprint. Lengthening a sprint could release pressure, but would expose the team to a greater management risk (Salazar 2014).

**Results**

The team started using a Kanban methodology instead of Scrum. The transition from Scrum to Kanban was quite smooth and included around two days of planning and discussing. With the new methodology, there was almost no overhead except from stand-ups. Instead of Scrum retrospectives, they used Kaizen meetings every two weeks where they focused on the future and what to improve instead of criticizing what they had done (Salazar 2014).

Kanban made the team more efficient, happier, and increased their focus on quality software. Another big impact was that agile became a core part of their company culture, and is now used by other non-technical departments at the company like Marketing and HR. A major unplanned benefit of Kanban was the increased quality in their work. In Scrum, it is easy to sometimes cut corners in order to finish a task, but this did not happen with Kanban (Salazar 2014).

**Conclusion from Case Study**

The power of Kanban comes from focusing on getting fewer items out faster through WIP-limits. As with anything else, this worked well for the team at Stormpath but might not work for every team in every situation. For this team, however, a Kanban methodology instead of Scrum created a happier, more productive, and less tense team (Salazar 2014).

**4.4 Agile Legal Team at The Lonely Planet**

Lonely Planet is a travel publisher, printing over 120 million books in eleven different languages. Their products include guidebooks, e-books, a website, a magazine, and a range of digital travel products and applications. Their legal team, Lonely Planet legal affairs, consists of three lawyers with a wide variety of work tasks (Sullivan 2013).
The Challenge

As a result of the digitizing of companies, together with the amount of free information available to customers, Lonely Planet's product range has become wider during the past years. As this was happening, the legal team's workload was also increasing. Before implementing agile, the team used a partner model meaning that most of the work was assigned to the team by a senior lawyer. The senior representative was also responsible for the work being finished and the risk management of the projects. The major challenges that the team was facing were balancing workload and hours, prioritizing tasks, knowledge sharing within the team, balancing daily work with other projects, engaging stakeholders effectively, cross-skilling of the team, and the management of stakeholders (Sullivan 2013).

When bringing up agile, there were a few concerns of how it would work for the team. One concern was about giving up control since they did no longer plan their own work. Another concern was the visibility of each person's work. When using agile, it is clear what someone is working with and how long it took them, hence the increased visibility. Two other concerns were whether agile would be a suitable method for legal practices, and how customers would react to the new methodology (Sullivan 2013).

Methodology Implemented

The legal team started noticing other teams at Lonely Planet using an agile model. After consulting the Agile Manifesto, they started to try out various agile methodologies. They found it hard to know where to begin, but after trying the different methods they found what worked well for their team. The key methodologies applied by the legal team at Lonely Planet are presented in table 8 (Sullivan 2013).
Table 8. The agile methodologies applied by Lonely Planet Legal Affairs, table adapted from Sullivan (2013).

<table>
<thead>
<tr>
<th>Tools</th>
<th>The Wanted Results</th>
<th>How they did it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic training</td>
<td>Getting to know the principles behind agile.</td>
<td>They talked to other teams at Lonely Planet how they worked with the agile model.</td>
</tr>
<tr>
<td>Visible and tangible tools</td>
<td>Visualizing for the team what is going on.</td>
<td>They used a task board: a whiteboard with cards of the new tasks that came up.</td>
</tr>
<tr>
<td>Stand-ups</td>
<td>Updating the team on the daily work and the plan for the near future. Getting to know each other and the ability to give feedback on the realisticness of a person’s workload.</td>
<td>They stood around the whiteboard and told the team what happened yesterday, what will happen during the day and tomorrow.</td>
</tr>
<tr>
<td>Estimation</td>
<td>Improving skills at estimation of task size and cost.</td>
<td>They gave every task a certain amount of points based on time, cost, and difficulty of the task.</td>
</tr>
<tr>
<td>Prioritization</td>
<td>Being able to do the work that actually matters.</td>
<td>They prioritized the different tasks when they were put on the Kanban board.</td>
</tr>
<tr>
<td>Sprints</td>
<td>Improving clear deadlines and structuring the agile work.</td>
<td>They had one-week sprints from Tuesday to Tuesday.</td>
</tr>
<tr>
<td>Retro</td>
<td>Learning what worked and what did not work.</td>
<td>They had 10-15 minutes retrospective meetings every week, where they discussed what worked well and not.</td>
</tr>
<tr>
<td>Balancing work loads</td>
<td>Balancing the amount of time put into business-as-usual and projects.</td>
<td>A certain amount of points went to doing projects every sprint.</td>
</tr>
<tr>
<td>Measurement</td>
<td>Being able to track workflow.</td>
<td>By scoring every task, they tracked how many points that had been allocated to a department, person, etc.</td>
</tr>
<tr>
<td>Super retros and planning days</td>
<td>Making sure the team is on the journey together and feels empowered by the work they are doing.</td>
<td>Every quarter, they had a super retrospective where they allocated more time than during a regular retrospective. Every half year they had a planning day.</td>
</tr>
</tbody>
</table>

Results

The results of the study were improvements of the challenges that the team faced in the beginning. These are represented in table 9 together with the results of the team’s old partner model and their new agile model (Sullivan 2013).
Table 9. The business challenges that the legal team at Lonely Planet were facing and how the results changed after implementing an agile model, table adapted from Sullivan (2013).

<table>
<thead>
<tr>
<th>Business Challenge</th>
<th>Results with Partner Model</th>
<th>Results with Agile Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balancing workload and hours</td>
<td>No control over incoming work.</td>
<td>Limited WIP through the task board.</td>
</tr>
<tr>
<td>Prioritization of tasks</td>
<td>General Counsel determined priority and delegated tasks to the team.</td>
<td>Team prioritized tasks and delegated collectively.</td>
</tr>
<tr>
<td>Knowledge sharing</td>
<td>Team worked in silos, often little knowledge awareness of what colleagues were doing.</td>
<td>Everyone knew what everyone was doing.</td>
</tr>
<tr>
<td>Project work vs. daily demand</td>
<td>Project work was de-prioritized in favor of daily demand.</td>
<td>All work was prioritized in terms of its importance of the business as a whole.</td>
</tr>
<tr>
<td>Engaging stakeholder effectively</td>
<td>Legal function was seen as &quot;mysterious&quot;, little understanding of workflow or management processes.</td>
<td>Informed stakeholders, and let them know the process, moving them in the process.</td>
</tr>
<tr>
<td>Cross-skilling of the team</td>
<td>Individuals worked in accordance with their job description</td>
<td>Individuals identified areas of interest (opportunities to 'up-skill') and fed this into the process of work allocation.</td>
</tr>
<tr>
<td>Management of stakeholders</td>
<td>General Counsel managed stakeholder issues</td>
<td>Stakeholders consulted with one another to assess priorities</td>
</tr>
</tbody>
</table>

Conclusion from Case Study

The case study resulted in six enablers that the team believes that other teams need in order to apply agile methodology to their business (Sullivan 2013). According to Sullivan (2013) the six enablers for an agile model are:

- Understanding of the basic principles
- Constant engagement of the team
- Willingness to experiment
- Courage to change
- Confidence between the team members
- Enthusiasm
5 Findings

In this chapter the findings from the case study at CS&A and the survey to other professionals working with agile will be shown. There will also be an aggregation of all enablers found in the theory, external case studies, and the survey with a conclusion of the final nine SFs.

5.1 Findings from Survey

The number of respondents to the survey were 62 and most (87.1%) of the respondents had worked with agile inside software development. All scale questions in the survey were on a scale from one to seven. The majority (58%) thought that agile had been successful to an extent of six or seven. Even more (75.8%) believed that agile was the best method for them to use in the project or process to an extent of six or seven. Half (50%) of the respondents believed that the rest of the team in the project or process would have answered the same way as them to an extent of six or seven. The correlations between the survey questions and how successful the method worked for the project or process and whether it was the best method are presented in table 10. The survey can be seen in appendix 2 and all results from the survey can be found in appendix 4.

Table 10. Correlations between the factors brought up in the survey questions and whether it was a successful and best suited method. Green means high correlation, yellow medium correlation, and red little to no correlation.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Correlation to best method</th>
<th>Correlation to successful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leadership</td>
<td>0.35</td>
<td>0.45</td>
</tr>
<tr>
<td>Understanding</td>
<td>0.34</td>
<td>0.43</td>
</tr>
<tr>
<td>Acceptance</td>
<td>0.41</td>
<td>0.57</td>
</tr>
<tr>
<td>Collaboration</td>
<td>0.23</td>
<td>0.47</td>
</tr>
<tr>
<td>Clear goal</td>
<td>0.17</td>
<td>0.41</td>
</tr>
<tr>
<td>Complexity</td>
<td>0.01</td>
<td>0.12</td>
</tr>
<tr>
<td>Stakeholder involvement</td>
<td>-0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>Task dependency</td>
<td>-0.24</td>
<td>-0.12</td>
</tr>
<tr>
<td>Standardized tasks</td>
<td>0.13</td>
<td>0.37</td>
</tr>
<tr>
<td>Formal roles</td>
<td>-0.06</td>
<td>0.07</td>
</tr>
<tr>
<td>Entrepreneurial culture</td>
<td>0.36</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Other aspects that were brought up in the survey, but not in the same way as the one listed in table 10, were size and location. A team of five to ten people seemed to be the most successful and location did not seem to have a noticeable impact on success or best method. The last question in the survey was a long text answer where some interesting thoughts were brought up. These are available in appendix 5.

5.2 Findings from Case Study at Klarna

5.2.1 Observations

*Overview of CS&A’s Current Agile Methodology*

The sprints at CS&A are currently two weeks long. Every sprint include one sprint planning, five stand-up meetings, and one sprint retrospective. The sprint planning takes place on the first Monday in the sprint and the sprint retrospective on the last Friday in the sprint. The stand-up meetings take place Monday, Wednesday, and Friday in the first week of the sprint, and Monday and Wednesday in the second week.

*Observations of Sprint Planning*

The sprint planning was around one hour long and the entire team was present at the meeting. The team manager presented an Excel file with all tasks that were on the team’s agenda at the moment. This Excel file was prepared by the two policy owners the Friday before the new sprint started. The team manager described all the tasks in the file and the team discussed what their biggest projects were and what was most important. They marked everything that had to be done in green, everything that should
be done but that was not prioritized in yellow, and everything else was just nice to have. The team manager went through the tasks and assigned them to different team members with their input. When a team member was assigned a task, they were asked to estimate it and the estimation was put into the Excel file. Every team member counted the estimated time for their assigned tasks and the goal was to have ten days of assigned tasks per team member.

*Observations of Stand-up Meetings*

The stand-up meetings were around ten minutes long and the entire team was present at the meetings. One of the team members held the meeting and the member showed Jira and the ongoing sprint on a large screen in the room. They went through every person in Jira and told each other what they focused on at the moment and if they had any problems for any tickets. The person who held the stand-up meetings was the same every time.

*Observations of Sprint Retrospectives*

The sprint retrospective was around one hour long and the entire team was present. The meeting started with the team going through and following up the action points from the last retrospective. After this, everyone wrote down their own thoughts about this sprint on post-it notes by themselves. The thoughts were divided into what was good, what could be better, and questions. After around five to ten minutes, the team members went up to the board and talked about their thoughts. The things that were good that came up during the retrospective were that a new employee had gotten up to speed, that they had a session where they learned about Jira, that a team member had gotten help from another, and that they had collaborated with another team. Things that could be better that came up were team members being late to the stand-up meetings, that time estimation is hard and that they are bad at updating these in Jira. They decided to discuss the importance of time estimation of tasks further in the next sprint planning. Questions that came up were:

- Why does not the team respect the stand-ups?
- How does their agile work connect to a long term planning?
- How can the team collaborate more?
After going through every team member's thoughts, they voted for the improvements that they thought were the most important. The three with the most votes were then discussed, which in this case included a long-term backlog and how to collaborate more with other teams. Every action point that was created from this discussion was put into Confluence and assigned to someone in the team.

5.2.2 Interviews

Some team members mentioned that the agile method is good, but a few also mentioned that it needs to get better. Furthermore, some said that they do not know any other methods to compare to. The general opinion was that agile could work better within the team if they got a better understanding of agile and how to use it.

The CS&A-Team

Out of the seven interviewees, five were CR-analysts, two were policy owners whose responsibilities are divided over geographic regions, and one of the policy owners was also the team leader. The policy owners are the ones forming and preparing the tasks ahead of the planning meetings. There were also unofficial roles within the team, which are based on countries, specific competences such as technical knowledge, and agile duties.

Further, six out of the seven interviewees had worked at Klarna less than six months and the team leader had had his current position for one and a half months. Two of the team members had previous experience of working with agile before the agile initiative in the CS&A-team. The team especially described their culture as supportive, but also as inexperienced, entrepreneurial, and fun.

The Agile Work at CS&A

Four out of the seven team members were part of the team when the agile initiative was taken. The first thought for them at that time and for those introduced later was in general positive. Some of them felt like it could be difficult to perform at the same level in the beginning and some thought it could provide an increased structure in their working process.
The results that have come from working with agile were according to some members of the group increased clarity of ownership of tasks, increased structure, increased transparency, simplified prioritization, and less dependencies. When asked, most of the team did not prefer using any other method, however they mentioned that they did not have many other methods to compare to. Suggestions that came up for improvement of the method were to focus more on the team, to use a common backlog, and to have a more long-term perspective.

The team generally believed that it is quite easy to prioritize between tasks. The things that facilitate prioritization are that they can ask the team manager, and when they have clear deadlines. On the contrary, what makes it difficult to prioritize between tasks are when there are many important tasks, when they get ad hoc assignments, and when there is not a clear ownership of a task. When discussing whether this should change, most interviewees thought that it was difficult to do and that it is probably the nature of the job. It was also mentioned that it would be easier if management worked with the same agile method.

The entire team agreed that it is difficult to estimate time for the tasks and that the time estimations often differ from the time consumption. In general, they underestimate the time a task will take, but sometimes they overestimate it. The time estimation gets easier through breaking down the tasks in more detail and having more experience in the job. What makes it difficult is if the task has high complexity, dependency on systems or other people, and lacking sufficient information about the task. Some of the team members mentioned that a more dependable infrastructure would make it easier to estimate the time of a task.

When the interviewees were asked about the sprints, there was a discussion regarding if the sprints have the optimal length or not. Some felt like they have an optimal length and some were hesitant. Furthermore, a problem was identified regarding that even though the sprint starts on Mondays, the work with the tasks of the sprints rarely starts until Tuesdays, or even later. The workload is planned to be two full 40-hour weeks. Also, it was mentioned that the documentation of the time consumption on the planned tasks needs to be improved, that the tasks often are not finished within a sprint, and
that the sprint is not a clear deadline. A suggestion to improve the sprints that came up is to schedule when to do the tasks, and then block that time so that nothing else interferes. Other suggestions were to have a Scrum master, have retrospectives every week and to not use sprints at all, but instead use Kanban.

Regarding the stand-ups, there was a discussion about covering problems and questions at the stand-ups instead of only reporting what the team members have done since the previous meeting. These problems could also be of emotional character. Many in the group did not see that much value in the stand-ups as they are today and that they are not as good as they could be. However, there was no one that stated that they would like to stop with them. Being on time for the meetings is an improvement possibility that some stated.

The discussion about the retrospectives displayed some views regarding that the action points formed at the meetings need to be followed up better. Also, it was mentioned that it would be good for the team to discuss and support each others’ struggles. Otherwise, the views on the retrospectives were positive.

The planning sessions are good, and have improved over time, according to a big part of the team. There was, however, a discussion about the need of long-term goals and connecting them to the planning sessions. Further, it was discussed that the tasks are prepared on a short deadline before the planning meetings, which could create negative stress.

Overall, the general opinion was that the team needs to improve their agile work to be able to benefit from it as much as possible. Improvements would be learning how to break down and structure the tickets, the software tools, and the basic principles of agile. Also, it would be good for them to grow tighter as a team.

*Understanding and Acceptance*

Most of the team members felt like they did not get enough information in the beginning when using agile. The information they did get was basic about Jira, such as how to cre-
ate a ticket. In general the team felt like it was difficult and confusing to get into the new way of working. Some liked it and some did not.

About half of the team said that they understand the point of working with agile. The rest of the team said either that they do not understand it or that they understand it partly. When they were asked about whether they think the other team members understand the point, they either answered that they did not know, think that there is no understanding, or some understanding. The majority of the team thought that there was a need of more education in agile work within the team.

The entire team answered that they accept to work with agile. When they were asked about why the team accepts working with agile, they said that it creates more structure, that they do not know any other method, that it is mandatory, or that it is a new and interesting method. When the team was asked the opposite question, about what the reasons for not accepting agile work are, the majority of the team explained it with incomprehension of the method. Further, some said it is due to agile work being time consuming, and that they would structure their work anyway.

The general opinion about support systems within the team was that they have helpful tools, but that the team needs to learn more about them in order to utilize them better. The team also seemed to know more about Jira than Confluence. Since they started working with Jira and Confluence, they have had one session a few months into working with the system, where they got introduced to it. No specific new support systems were wanted by the team. The majority of the team mentioned that they need many different types of data and that the information accessibility generally is good. Some said that the problem is rather the opposite as it is difficult to get the right data because they have access to very large amounts of data. Also, some mentioned that the high staff turnover makes it difficult to know whom to ask for the information that they are looking for. Further, it was mentioned that good infrastructure and ease of asking other teams, due to personal relationships, facilitate information accessibility. Some thought that the information accessibility was good as it is, while others thought it would be good with an even better information infrastructure and more documentation. The documentation
could be about previous projects, to easily be able to consult it when conducting similar projects in the future.

The Environment of the Team

The external environment, everything outside of the team, was generally seen as fast-changing. Reasons for this according to the team were that Klarna is a big start-up, teams change, many decisions are made on management level, legal changes in countries, and uncertainties in customer behavior. When it comes to the internal environment, the environment within the team, it was also seen as fast-changing. Reasons for this were that priorities can change quickly and that they have been recruiting many new team members.

The process is complex according to most team members, but many also mention that it will be easier when they get more experience and that it could be difficult to change the complexity short term. Some mentioned that the complexity could be explained by the newness of the team but also because there are a lot of new assignments. The environment the team works in was also mentioned as a reason to complexity by some as it is complex in its nature and that there is a lot of information that needs to be understood and processed.

The team generally believed that they are flexible as a team. The main reasons for this are that they work with agile and that the company overall is flexible. The reasons that make them less flexible are deadlines and dependencies on other information sources, but a few also said that they are flexible in all ways.

Task dependency is not often occurring and is not a problem according to the general view of the team. Sometimes, they are dependent on other teams performing their tasks, but rarely on other CS&A-team members’ tasks. Furthermore, the team did not think that the task dependency could be changed much.

The team mentioned that there are generally not many tasks that are standardized or could be standardized in the future. Most team members brought up some reports and
analyses as potential tasks to standardize or even automatize. They generally believed that this would fit into the agile way of working.

**Leadership**

Overall, the team was content with the leadership as they thought that it was good and clear. The leader is approachable and present for questions. However, there were comments regarding the environment they operate in being fairly unstructured and there exists no long-term planning, which ultimately leads to that they have no clear direction as a team. Some of the interviewees wanted more structure, but commented that it might be hard in this kind of unstructured environment. Furthermore, some said that they have a lot of own responsibility and freedom, which is something that the ones mentioning it see as positive. The general view of the team was that they knew the task goals and had an overall understanding of the CR and company goals.

**Formal Roles and Collaboration**

The team agreed on that their roles are not formal, but they did not agree whether they should be more formal or not. Some team members thought it should stay the same, some thought that the roles should be more formally defined and some that they should become more formal in the future. The majority of the team said that they collaborate little in the team and when they do, it is mainly by asking each other questions when they need help. Also, some interviewees saw the agile components, such as the planning session, as a time when they collaborate.

The amount of collaboration changed through working with agile, according to most team members. The majority would also like to collaborate more than they are doing today if it would create added value. A suggestion that came up in the interviews, for increased collaboration, was to conduct certain projects in pairs, so that they could learn from each other and get two different views on the task. Something else that came up was the importance of knowing more about the competencies of the other team members to facilitate collaboration. Further, the policy owners could present tasks on a higher level to the team, on which they would have to collaborate to find an approach on how to solve and break down into smaller tasks. As it is today, the policy owners give out detailed instructions on how to solve the broken down tasks.
Stakeholders

What could be understood from the interviews is that the team in general does not understand who their stakeholders are, however, they are more aware of their internal stakeholders. Furthermore, the stakeholders of the team are not aware of that the team is working with agile. The team mentioned that they like when stakeholders are involved in their process. Some mentioned that they would like to have a better understanding of who the other teams are and what they are doing. CS&A was described as “the spider in the web” within Klarna. Something that came up in many interviews was that they had a coffee with one of their stakeholders, Risk Engineering, in order to get to know them better. This was appreciated and enabled easier contact afterwards.

Culture

The team expressed that it is possible to take initiatives as long as they create value. Examples of initiatives that previously have been taken in the group are everything from deciding on accepting a certain type of customers to being responsible for holding a course within the company. The general opinion was that taking own initiatives work well with the roles of the team.

The team saw the process as a learning organization. They explained a learning organization as being given new challenges every day and needing to learn by doing. However, a lack of innovation within the process was expressed and that there could be more space for that type of learning.

Management Support

The team said that the management is not very involved in whether they work with agile or not. Most team members would not want this to change, but it is also brought up that it would be easier if all teams within Klarna worked with agile.

5.3 Findings of Enablers

5.3.1 Enablers from the Survey

The correlations between the different analyzed aspects and whether it was the best method are lower for most aspects compared to the correlations between the aspects and whether it was successful. The correlations calculated from the survey imply that
acceptance, management support, and flexibility have the highest correlation to whether agile is successful and whether it is the best method. The ten aspects with the highest correlations are listed below in descending order, both correlated to success and if it was the best method.

Aspects with highest correlation to best method:

1. Flexibility
2. Acceptance
3. Management support
4. Entrepreneurial culture
5. Leadership
6. Understanding
7. Support systems
8. Prioritization
9. Task dependency
10. Collaboration

Aspects with highest correlation to success:

1. Acceptance
2. Management support
3. Flexibility
4. Collaboration
5. Leadership
6. Time estimation
7. Prioritization
8. Understanding
9. Clear goal
10. Standardized

5.3.2 Enablers from Previous Case Studies

The different case studies shed light on different enablers that were needed in their own particular case. The different enablers that were identified are shown in the following list as well as in which case they were found:

- Long-term perspective (Putnam 2014; Chang 2010)
- Understanding of agile (Sullivan 2013)
- Finding the right employees (Chang 2010)
- Open-mindedness, courage and flexibility of team (Sullivan 2013)
- Collocation (Chang 2010)
- Team support (Putnam 2014; Sullivan 2013)
- Management support (Putnam 2014)
- Supporting infrastructure (Putnam 2014)
- Stakeholder involvement (Chang 2010)
● Confidence between the team members (Sullivan 2013)
● Large projects (Putnam 2014)

In order for the agile initiative to be successful, there needs to be a long-term perspective. It should be seen as a strategic investment and not something that can be implemented using a standardized approach in a short period of time. Research should be done in the particular company to find its own specific needs (Putnam 2014). Furthermore, as the agile way of working is different from the traditional way of working it will take time to learn it and find the practices that works best with the team (Putnam 2014; Chang 2010). For the team to get a head start in the learning curve understanding is crucial, especially of the basic principles (Sullivan 2013).

It is important to find the “right” people for working with agile. They need to be flexible, creative, open-minded and not have a problem with a lower level of structure (Chang 2010). The entire team also needs to be open-minded, flexible, and have courage to keep changing to adapt the perfect method (Sullivan 2013). Being collocated, the team can more easily have fast and direct communication. It also creates a learning environment (Chang 2010). Enthusiasm (Sullivan 2013) or at least a support for adapting AM by the team (Putnam 2014) is necessary. Except for that the teams need to support the initiative, there needs to be a solid support by the management. This means for example investments in the infrastructure needed for working with agile (Putnam 2014). Something that might be harder to affect is that the stakeholders should be involved in the work. If they are hard to get a hold of it can cause delays (Chang 2010).

The team members should have confidence in each other when working with agile. This can be done through signing an agreement in the team (Sullivan 2013). However, Sullivan (2013) does not mention if there needs to be confidence between the team members in different areas than the agile work. Lastly, Putnam (2014) found that it was more beneficial to work in large projects. However, there are other case studies showing the opposite (Putnam 2014).
5.3.3 Enablers from Theory

Conforto et al. (2014) bring up 41 enablers, but focus more on the ten enablers that are mentioned the most often in literature, which might not necessarily mean that they are the most critical enablers for using agile. Since the enablers mentioned by Conforto et al. (2014) are already taken into account in the survey conducted in the Master Thesis Project, the focus here is on enablers mentioned by other authors than Conforto et al. (2014). These are listed below. Enablers that are very clearly linked to software are neglected.

- Formal roles (Chin 2004)
- Leadership (Hass 2007; Näslund 2013; Awad 2005)
- Small tasks (Nicholls, Lewis & Eschenbach 2015)
- Short communication intervals (Nicholls, Lewis & Eschenbach 2015; Fowler & Highsmith 2001)
- Collocated teams (Hass 2007)
- Innovative and ever-changing environments (Highsmith 2004; Awad 2005)
- Small teams (Nicholls, Lewis & Eschenbach 2015; Awad 2005)
- Responsive to change (Awad 2005; Fowler & Highsmith 2001; Highsmith 2004)
- Few rules and procedures (Awad 2005; Fowler & Highsmith 2001)
- Collocated (Awad 2005) and dedicated customers (Awad 2005; Fowler & Highsmith 2001)
- Management support (Näslund 2013; Davies 2012)
- Culture (Näslund 2013; Awad 2005)
- Long-term perspective (Näslund 2013; Fowler & Highsmith 2001)
- Training (Näslund 2013; Davies 2012)
- Cross-functional teams (Näslund 2013; Chang 2010)
- Measuring success (Näslund 2013)
- Not working on the same material at the same time (Nicholls, Lewis & Eschenbach 2015)
5.3.4 The Nine SFs

After analysis of the theory, previous case studies, and findings from the survey, the nine most important enablers were identified according to the authors. The SFs that were also found to be important from the survey are listed below:

- Flexibility (Sullivan 2013; Awad 2005; Fowler & Highsmith 2001; Conforto et al. 2014; Highsmith 2004)
- Acceptance (Putnam 2014; Sullivan 2013; Conforto et al. 2014)
- Management support (Putnam 2014; Näslund 2013)
- Understanding (Sullivan 2013; Näslund 2013; Conforto et al. 2014)
- Leadership (Hass 2007; Näslund 2013; Conforto et al. 2014; Awad 2005)
- Small teams (Nicholls, Lewis & Eschenbach 2015; Awad 2005; Conforto et al. 2014)

Other SFs, that were not seen as crucial in the survey, but still are seen as some of the most important enablers in this Master Thesis Project are the following:

- Dedicated stakeholders (Chang 2010; Awad 2005; Conforto et al. 2014; Fowler & Highsmith 2001)
- Long-term perspective (Putnam 2014; Näslund 2013; Chang 2010; Fowler & Highsmith 2001)
- Collocation (Chang 2010; Hass 2007; Fowler & Highsmith 2001; Conforto et al. 2014)

All nine SFs can be seen in figure 6.
Figure 6. The nine SFs that the authors came up with during the Master Thesis Project.
6 Analysis

This chapter includes a discussion about the current situation at CS&A, what is good as it is and where gaps exist between the team’s method and best practice. All facts that do not have a reference come from the findings of the Master Thesis Project.

The Master Thesis Project has brought up many aspects of AM and some of TM. When it comes to choosing between these two methodologies there is much that implicates that the preferable methodology for CS&A is AM. One of the main purposes of using agile is that it helps to work in an environment that is fast-changing (Highsmith 2010), which is the case for the entire company, but also the specific team. When the authors discuss enablers and whether agile is best for the team, AM is compared to TM, since this is seen as the natural option to agile in the Master Thesis Project.

There is a trade-off regarding increased documentation, which generates a bigger amount of saved information, which can save time in the future, but is also time consuming at the moment. According to the second ground pillar of the agile manifesto and the tenth practice it is important to not overdo the documentation (Fowler & Highsmith 2001). Especially as the processes are hard to standardize and that this is not the aim of agile (Highsmith 2004). However, Highsmith (2004) mentions that there should be a balance between structure and flexibility. It could therefore be an advantage to have a slightly increased amount of documentation to decrease the internal lack of structure. The team should therefore have a discussion on what activities that are valuable to document as it will save time in the future and decrease dependencies on individuals. This is valuable as the staff turnover is high. The documentation should be done in Confluence to not introduce more software systems and gather all information in the same place.

The CS&A-Team

The CS&A-team is young and new, which is not recommended by Conforto et al. (2014), who say that the team should have at least three years of experience. Most of them do not have any experience from another job or from using agile before joining the team,
which further connects to the team’s inexperience. However, there are other teams at Klarna that have more experience both as a team and within agile, which might benefit the CS&A-team and make up for their own inexperience through shared knowledge. The size of the team, seven people, works well with AM, as it is small, which facilitates better communication (Nicholls, Lewis & Eschenbach 2015). However, Chang (2010) mentions that a team with young members is often innovative, which is preferable in agile work.

The team also describes themselves as very flexible as a team, which is one of the nine SFs in this study. This is also something that Chang (2010) mentions as important for agile work. Another aspect that Chang (2010) brings up is that agile does not fit all people, and this is important to have in mind.

**The Agile Work at CS&A**

The team seemed generally positive about agile but they do not have much experience in it from before, neither did they get much information about the way of working when they started with it. This might be a reason for the lack of criticism. The general picture is that they want to continue using agile but with some adjustments. However, the ideas for adjustments brought up in the interviews were quite mixed depending on the interviewees.

**Kanban vs. Scrum**

One recurring point that was brought up during the interviews was whether it is beneficial to use time limited sprints as they are doing today. Salazar (2014) mentions why Stormpath switched from Scrum to Kanban and stopped using the time limited sprints. Some of the issues that Salazar (2014) mentions with Scrum are similar to the ones brought up in the interviews with the team. One example is the struggle with time estimations and how they create an environment where you either have to work very hard in the end of a sprint, or not being able to finish the task within the sprint.

There are other ways in which the team at Stormpath and CS&A are similar. They both operate in a fast paced environment and have similarities to start-up companies. However, there are a few differences between the teams. The team at Stormpath is operating within software development (Salazar 2014) while CS&A is not. Stormpath does not experience that collaboration is a weak spot (Salazar 2014), while it is something that
some members at CS&A mentioned as an improvement possibility. Still, it could be argued that CS&A would benefit from using Kanban instead of Scrum in their agile work. Another argument to stop using the time limited sprints is that they often have tasks that are not finished in the sprint. Many of CS&A’s tasks take longer time than two workweeks, and the team is not very good at breaking them down into smaller tasks.

It can be assumed that a Kanban methodology works better for processes where a team works with a more continuous flow of work and not clearly defined deliverables. If the case team keeps using time-limited sprints, it should be further researched how long a sprint should be to be able to finish the planned work within a sprint. If they start using Kanban, other questions arise. It would have to be decided how the planning should work with the new method, and who would be responsible for distributing tasks and prioritizing them. The easiest would most likely be if they could work similarly today, but without the sprints. Short planning sessions could be incorporated into the stand-ups and the retrospectives could be replaced with Kaizen meetings. The team manager and policy owners could still be responsible for collecting and distributing tasks. The team members would still be responsible for adding their own tickets to the backlog. Salazar (2014) mentions that his team needed two days of planning and discussing with the team to adapt to Kanban from Scrum. It could be assumed that this would be an easy transition for CS&A as well. An interesting and unexpected benefit that Salazar (2014) found in his team was that the quality of work was improved with Kanban compared to using Scrum. This could also be the case for CS&A.

If Scrum will remain the used method, implementing the role of a Scrum master could help the team’s agile work (Schwaber 2004). The role could be passed around the team with a new member being Scrum master every sprint and that he or she would be responsible for making sure that the agile method is working and be responsible for the stand-up meetings and retrospectives. This role could be put as a ticket in the sprint, so there is time for the Scrum master to administer the sprint. Also, the planning per team member should be less than 80 hours, which is how it is today. There are meetings and ad hoc work that are not included in these 80 hours, which makes it difficult to finish the work within a sprint. This is suggested by Chang (2010) where the team did not plan all working hours for agile work in order to fit in ad hoc assignments.
Agile Practices

A problem with the team’s work today is that some members often are late to the stand-up meetings and the team did not feel that there is much value added by holding the meetings. A solution could be to conduct the stand-ups later in the day, not when the members are coming to work in the morning. At the same time, it might be beneficial to have the stand-ups first thing in the morning as a natural way to start the day. Either way, the team needs to get better on being on time, especially regarding stand-ups since those meetings are very short. If one person is ten minutes late for all stand-ups, the wasted time that the others have to wait represents almost three full working weeks for one employee in one year. If other meetings that team members are late to would be included in this number, it would be even higher. The team also needs to understand the value of the stand-ups better and include this in the training for new employees, mentioned under “need of education”.

Regarding the retrospectives most seemed quite happy but had some ideas on how they could change. One thing that was brought up was that the team does not talk about the results of their work during the retrospectives. In Scrum, there is a meeting called Sprint review, where results are looked at and discussed (Schwaber 2004). An idea is that the team would implement a version of this meeting where they would discuss what they have done during the sprint and what the results have been from the tasks.

Something else that would simplify the use of agile is if management and stakeholders worked with the same methodology. This would lower the probability of new tasks coming up during a sprint. However, that is out of scope for this Master Thesis Project, and is therefore suggested as further research.

During the interviews, the difficulty of estimating the time consumption of tasks were discussed. If the team got better at always documenting the time consumption, for example through burn down charts, it would be easier to go back and look at previous need of time for similar tasks. However, Salazar (2014) asks himself if there really is a point of time estimating if task needs to be executed no matter what. One solution to this is to simply use Kanban and stop the use of time-limited sprints. If sprints still are used, CS&A should break down the tasks further so that it is be easier to estimate time
consumption for every subtask and aggregate it for the entire task. However, it would be more time consuming in the planning process. Another idea is to use points instead of hours, as suggested by Sullivan (2013). However, it might cause lower precision. Also, the idea of planning for less than 80 hours would most likely decrease the risk of not finishing within a sprint and decrease the importance of time estimations.

One of the Master Thesis Project’s SFs is collocation, which is also mentioned by one of the twelve principles of the agile manifesto (Fowler & Highsmith 2001). The CS&A-team is located in the same room, which is a good starting point for working with agile.

*Understanding and Acceptance*

The team also mentioned that it was difficult to get into the way of working in the beginning, which again could be due to the little amount of information they got about agile. According to an SF found it is important with a basic understanding of agile. Also for any change initiative to work there need to be ongoing training of both team members and the management (Näslund 2013).

Another SF is that there must be an acceptance for working with agile in the team and the management. The team said that they accepted working with the method, but as the question was posed “Do you accept working with agile?” it might be hard to say no. One of the reasons against acceptance is lack of understanding of AM. This further indicates that the team needs education in the subject, as well as that the team said that they do not understand the method and that they do not think that the rest of the team understands it either.

The team mentioned that they are happy with the support systems they have at the moment but expressed that they do not know enough about them. Furthermore, they wanted to learn more about how to break down and structure tasks and further understand the basic principles of agile. The education should thus cover those things, especially before initiating any new software. There should be an initial training as well as continuous education; to create the understanding but also give a long-term perspective, which also is an SF.
The Environment of the Team

As the internal and external environment is fast-changing and uncertain it is beneficial for CS&A to work with AM as it creates a balance between flexibility, which is needed to react and be innovative, and structure, which is needed for efficiency (Highsmith 2004). Furthermore, as the environment is fast-changing it is hard to standardize task, which is something that the team also said. This makes it beneficial to work with agile, as trying to standardize in fast-changing environments often is counterproductive (Highsmith 2004).

Many of the team members commented on that they found their process complex. However, the team is new to what they do so a lot of the complexity can be derived from the early stage they are in on the learning curve. Thus, no intervention is needed here.

Leadership

It would be good for the team to have an improved long-term planning as it would provide a more clear direction and would facilitate the leadership. As the team has a fairly good understanding of Klarna’s overall goals and the specific goals for each task, there is a gap in CS&A- and CR-specific goals. The goals should be measurable, both to know if the method is successful and for the morale of the team (Näslund 2013).

If choosing to continue with Scrum the leadership needs to be clearer that the sprint is a deadline. However, it is important that the leader does not become a micromanager (Hass 2007), which does not seem to be a risk in the case team. According to the agile manifesto it is important to build the team around motivated people, give the team a lot of freedom and support when they need it (Fowler & Highsmith 2001). Here, there does not seem to be a gap between the case team and best practice either.

The leaders, both internally and externally, need to support the initiative, as it is an SF. For the initiative to work the management must lead the team and go all-in regarding what practices to use (Putnam 2014). It was mentioned in the interviews that it would be beneficial for CS&A if the entire company worked with agile. The team does not feel like management needs to support the agile initiative more. However, only by knowing more about the basic principles of agile and how CS&A works, the team’s planning could
hold up much better since less tasks would come up in the middle of sprints. If the entire company worked with agile, it would probably create the same benefits for CS&A.

**Formal Roles and Collaboration**

When working with agile it is important with outspoken and documented roles to avoid misunderstandings (Chin 2004). As the different team members have specific areas of expertise and unofficial roles it would be easy to just document this to create transparency for all team members. This is in a way more formal, but it will only be about documentation and not any change in the way they work. This could be in the form of “cards” or short resumes where each team member’s competence is shown. At the same time, formal roles did not end up as one of the nine SFs and should not be weighted too heavily.

When it comes to collaboration, one suggestion is to have more high-level tasks that are planned by the team. This is in line with what Hass (2007) suggests, that a team manager should not give detailed instructions on how to solve tasks, but rather create collaborative relationships in the team and set ground rules for the team to follow. As the tasks have low dependency on each other the team members often can work independently with their own tasks, without having to wait for someone else to finish.

It is not problematic that the team does not collaborate more than they do at the moment as the collaboration in agile work is mostly while performing the agile activities such as stand-ups and planning sessions. Those activities improve the relationships within the team (Mahale 2011).

**Stakeholders**

Having a close collaboration with the stakeholders is an SF. If the team knew more about their stakeholders, and the stakeholders knew more about the team’s work this would be easier facilitated. This would mean that fewer tasks come up in the middle of a sprint and that the team more easily could communicate with them. Ideally, the team should be cross-functional, with stakeholders from for example Engineering and Commercial on it (Conforto et al. 2014). They should all also be collocated (Hass 2007; Awad 2005), as the most efficient way to communicate is face-to-face (Hass 2007). Since CS&A
mostly mention internal stakeholders as the most important, this should enable them to work well with their stakeholders and also enable their agile work.

As the above alternative is not possible within the scope of this Master Thesis Project, it is suggested as future research. The team should, however, get to know their stakeholders better through having coffee with them as they previously have had with a team within Engineering. This is further supported by the agile manifesto that states that communication between the business people and developers should be on a daily basis (Fowler & Highsmith 2001). The team also has mostly internal stakeholders that are on-site. It is suggested by Awad (2005) that it is good to have dedicated on-site stakeholders, mostly internal (Awad 2005).

Culture

AM is characterized by working in an entrepreneurial culture, according to the agile manifesto (Fowler & Highsmith 2001) and Conforto et al. (2014). In CS&A, the culture is entrepreneurial according to the team. However, it did not qualify as one of the SFs in this Master Thesis Project and as it is already in place nothing should actively be done to enhance the entrepreneurial culture.

Furthermore, the team feels that they are operating in a learning organization, as they need to learn by doing. Working in a learning organization is good for enabling work with agile (Conforto et al. 2014), even if it is not an SF. However, learning by doing is not always positive as the team multiple times mentioned that they do not know enough about agile. The learning organization should be more about constant knowledge improvement for the employees and not about having a hard time performing one’s work due to lack of knowledge. Education should be given to the employees as mentioned before. With the basic tools, it is easier to be creative and to create value.

Management Support

One of the nine SFs was management support, indicating that when working with agile it is important with that kind of support. At CS&A, the management is not very supportive and involved in how they work, but at the same time they do not work against the methodology either. It could be beneficial with more support from top management, which might increase the number of other teams working with agile as well. However, this would be quite difficult to require from other teams.
7 Conclusion

In this chapter the analysis will be synthesized into a conclusion of the research questions and recommendations are formed to CS&A about how they can improve their work in order to become best practice.

7.1 Answers to Research Questions

The answers to the research questions are listed below with references to the report.

RQ1: What agile practices are proposed in theory and previous case studies?
See chapter 3 and 4.

RQ2: What Success Factors does a team need to possess in order to benefit from working with agile compared to other management methodologies?
There are nine SFs identified in the Master Thesis Project and they are:

- Flexibility
- Acceptance
- Management support
- Understanding
- Leadership
- Small teams
- Dedicated stakeholders
- Long-term perspective
- Collocation

RQ3: What are the gaps between the Success Factors and practices that a team ideally should have to benefit from working with agile and the capabilities and practices case team currently have?
See 5.2, 5.3.4 and chapter 6.

RQ4: What can be done in order for the case team to fill the gaps mentioned in RQ3?
The case team should follow the recommendations posed in 7.2.
7.2 Recommendation

The recommendations can be seen in figure 7.

![Diagram of General Recommendations]

Figure 7. The recommendations for CS&A’s agile strategy. The blue represents the prioritized recommendation.

The authors recommend CS&A to implement the recommendations in 7.2.1 and to implement Kanban, see 7.2.2. If the team chooses to proceed with Scrum they are recommended to do it according to the recommendations in 7.2.3. Since Kanban and Scrum are quite different methods, recommendations for Scrum are also presented in case the team would not want to change to Kanban. Everything that is not covered in the recommendations is intended to stay the same.

7.2.1 General Recommendation

The general recommendations are listed below.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>How to do it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial basic training</td>
<td>Initial basic training and information about what agile is, how to use the software tools, and what the value of different parts of agile is.</td>
</tr>
<tr>
<td>Ongoing training</td>
<td>Ongoing training to remember and enhance skills.</td>
</tr>
<tr>
<td>Long-term planning</td>
<td>Create a long-term planning with measurable goals for the team and connect all tasks to this. The planning should be connected to the goals of CR and what CS&amp;A can do to contribute to these. The planning should be done collaboratively in the team but prepared by the policy owners.</td>
</tr>
</tbody>
</table>
Get to know teams
Get to know the other teams better through socializing together, for example have coffee together. Plan to meet one new team every two weeks.

Inform stakeholders
Inform the stakeholders about the team’s agile work, such as when planning meetings are held.

On time for stand-ups
The team needs to become better at being on time for the stand-up meetings. They should also make sure that the entire team understands the value of the stand-ups.

Competence card
Create competence cards where all team members get to specify what they are good at and what they want to do. These should be visible for the entire team to increase transparency and clarity around roles and responsibilities. See appendix 6 to see what this card could look like.

No added support systems
Do not add any new support systems, at least not until Jira and Confluece are understood thoroughly.

Support going all-in
Support and enforce the team to go all-in with all activities such as following deadlines, being on-time to meetings, reporting time consumption on tasks and join educations about agile.

Discuss documentation
CS&A should have a meeting and discuss what tasks that are value-adding to document. These tasks should for example be recurring and not affected by environmental uncertainty, as they otherwise might change fast and thus decrease the value of documenting them. The team should create a template on how to document these tasks in order to increase their efficiency in the future.

Implementation and measuring
The team should implement the changes and make sure to measure the results of these implementations. The results should be based on productivity, meeting of deadlines, quality, structure, and happiness of the team. This could be done through a survey before the implementation and the same one after using the methodology for two months.

7.2.2 Recommendation for Kanban
The big difference between Kanban and the way CS&A work today is that there is no time estimation of tasks and that there are no sprints, but instead an ongoing work process. The recommendations for Kanban are listed below.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>How to do it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile coach</td>
<td>Have an agile coach that is responsible for conducting agile meetings and making sure that agile is used. This role should rotate within the group with two weeks intervals and be put into the backlog as a task.</td>
</tr>
<tr>
<td>Policy owners add and prioritize</td>
<td>The policy owners add and prioritize tasks individually, as they come up, in the product backlog. They create the tickets in Jira, write down the basic information, and then the team member who is taking on the task will groom it.</td>
</tr>
<tr>
<td>Planning in stand-ups</td>
<td>The tasks are presented on the stand-up meetings and there is therefore no planning session.</td>
</tr>
</tbody>
</table>
### Agile Management Outside of Software Development

**Longer stand-up meetings**

The stand-up meetings should be held three times a week, Monday, Wednesday, and Friday, as today. The stand-up on Friday could be incorporated with the Kaizen meeting. The stand-ups should be around 15 minutes in order to have time for planning.

**Choosing task**

The team members should choose the task that has the highest priority on the product backlog and that they have the right competence card for. See 7.1.1 and appendix 6 for a closer description of the competence card.

**Kaizen meetings**

Instead of the retrospectives, Kaizen meetings should be held every two weeks to discuss similar things that are discussed in the retrospectives today. The big difference is that they will not be finished since sprints are no longer used. But since the focus of these meetings is on the agile work, the difference should not be significant.

**Kanban board**

The team should use a Kanban board, which is similar to the Scrum board used today, and it should be common for the entire team. The support systems used today support Kanban boards as well, which will make the transition easier.

**WIP-limits**

The team should decide on and evaluate a limit for WIP. The ad hoc assignments from the management should not be included in these.

**No time estimation**

There is no time estimation.

### 7.2.3 Recommendation for Scrum

Recommendations for Scrum are listed below.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>How to do it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan 70 hours</td>
<td>Plan 70 hours within a sprint instead of 80 hours, leaving one hour per day for meetings, grooming of tasks, underestimated tasks and ad hoc work. This number is based on an estimation of current ad hoc work during a sprint.</td>
</tr>
<tr>
<td>Scrum master</td>
<td>Have a Scrum master that is responsible for making sure that agile is followed and conducting the agile meetings. The role should rotate within the team and added as a task in the backlog of a half day task. This should be changed if the team finds that it takes shorter or longer time.</td>
</tr>
<tr>
<td>Clear deadlines</td>
<td>The team manager should be more clear that the sprint is a deadline and that all planned tasks should be finished within the sprint.</td>
</tr>
<tr>
<td>Break down tasks</td>
<td>Tasks should be broken down to the point where it is possible to finish them within a sprint.</td>
</tr>
<tr>
<td>Sprint review</td>
<td>In order to not only discuss the agile work as today at retrospectives, a sprint review will add discussions about the results as well. The sprint review should take place right before the regular retrospective and be around 15 minutes long.</td>
</tr>
<tr>
<td>Discuss sprint length</td>
<td>The team should investigate whether two weeks is the right length for them. Either, they could lengthen the sprints to better suit their tasks today, or become better at breaking down tasks so that they suit the sprint.</td>
</tr>
</tbody>
</table>
8 Discussion

In this chapter there is a discussion about how the Master Thesis Project has contributed to the academia, the chosen methodology, and propositions for further work within the subject.

8.1 Contribution to the Academia

There are a number of articles and case studies of agile within software development. However, the amount of articles of agile outside software development is smaller, especially regarding case studies on this subject in academia (Conforto et al. 2014). This is a popular subject outside of the academic world, which makes an academic case study within the subject interesting.

One result of the Master Thesis Project was the nine SFs that are the most important enablers for agile within a team, according to the study. These nine SFs can be used for understanding when agile can be used successfully.

The findings from the case study shows that CS&A is a good candidate for the use of agile, but also that there are gaps between their capabilities and practices today and best practice. The Master Thesis Project provides recommendations for how the team can fill these gaps by using agile in different ways. This is a contribution to the academia since there is little written about how specific teams can use agile (Conforto et al. 2014), but especially with teams that are already using agile and want to improve.

8.2 General Contribution

All teams that want to start working with agile both inside and outside the business society can use this case study to find similarities and differences between their team and the case team. They can use the SFs to analyze what their gaps are between themselves and best practice and take influence from the recommendations to CS&A. If a team wants more significance in the analysis of their team they could transfer the entire study to their own team using the same interview guide and so on in order to better find their gaps.
Especially Klarna could benefit from using the methodology and some of the findings from the Master Thesis Project to analyze if the other teams in the company could benefit from using agile and in that case how. The findings are more generalizable to them as the culture and other intangible factors probably are similar in other teams.

There is also a contribution to the agile society as a discussion about working with agile is lifted out of its normal context and into other communities such as the academic world. This increases the influences and the spread of the management methodology to reach a wider audience. It is especially broadened from the software community, which is where the management methodology has its stronghold.

### 8.3 Further Work Recommendations

The authors recommend that CS&A implement all recommendations in chapter 7 as soon as possible and measure the results of implementing the changes. Generally, theory does not provide much information regarding measurable results from using agile. There should be more focus on this in future work in order to see what the actual effects of agile are as well as create a more convincing case around why to use agile. The reasons for presenting all recommendations at once are partly because it is suggested to go all-in when using agile (Putnam 2014), and also that it does not have to take more than two days to change from Scrum to Kanban (Salazar 2014).

The authors also recommend that this study should be repeated within the entire company in order to investigate whether agile could be implemented or improved in other teams. By using the methodology in this Master Thesis Project, other teams can use the nine SFs and the interview guide to investigate their own gaps and how these could be improved. The recommendations could also be used to some extent if the team finds similar gaps as in the Master Thesis Project. The entire company should also look further into using cross-functional teams since this enables the use of agile.

There could also be a more significant survey conducted with more respondents and a more valid sample methodology. This survey could challenge the SFs found in the Mas-
A study like that could make up an entire Master Thesis for quantitative researchers. In that way possible industries for AM could also be found.
References

*A Brief Introduction to Kanban*, n.d. Available from:
<https://www.atlassian.com/agile/kanban>. [13 April 2016].


*About Us*, n.d. Available from:


Awad, M. A 2005. ‘A Comparison between Agile and Traditional Software Development Methodologies’. The University of Western Australia.


Highsmith, J 2004, Agile Project Management, Pearson Education Inc. Boston

Höst, M, Regnell, B & Runeson, P 2006. Att genomföra examensarbete. First ed. Studentlitteratur AB.


Javanmard, M & Alian, M 2015. ‘Comparison between Agile and Traditional software development methodologies’.


from: 

McLaughlin, M n.d. *Agile Methodologies*. Available from:  

*Metaphor*, n.d. Available from:  


OPS International LLC 2015, *Comparison of Agile Methods by the Agile PrepCast*.  
Available from: <https://secure-hwcdn.libsyn.com/p/7/3/7/73773eb1e8ac86ca/The_Agile_Methods_Comparison_by_The_Agile_PrepCast.pdf?c_id=5993202&expiration=1461832436&hwt=e5a45063ab5eb8fdeb1e3aa38831b436>. [28 April 2016].


*Planning Game* n.d. Available from:  


Appendix

In the Appendix, the interview guide is presented followed by the survey conducted with other companies and the full results of the survey.

Appendix 1 - Interview Guide

Intro

The goal with our master thesis is to come up with recommendations for how your team can work with agile. As part of our data gathering, we are doing these in-depth interviews with all members in your team to get a better understanding of how you work and how this could be changed. The structure of the interview is as follows: first some basic questions, a bit about how agile works today, how it could work in the future as well as follow up questions from the survey. The interview is anonymous and your name will not be mentioned anywhere in connection to you answers.

Basics

How long have you worked at Klarna?
What is your position at Klarna?
What is your official role in the team?
What is your unofficial role in the team?
How would you describe your team's culture?
Have you worked with agile methodologies before you started at Klarna? (within/outside software development)

How did it start?

Where you a part of the team when you started working with agile?
What was your first thought when the team started talking about working with agile?
What information did you get about agile in the beginning?
How has it been to get into the new way of working?

How does it work now?
What have the results been from using agile methodologies?

Would you prefer to use any other methodology? I so: which one?

**How will it work in the future?**

What possibilities for improvement do you see in the team’s agile work at the moment, when it comes to:

- a. Sprints
- b. Stand-ups
- c. Retros
- d. Planning meetings
- e. Anything else?

**Follow-up questions for the survey**

To what extent did the team have good and clear internal leadership?

Describe the leadership you have in your team.

Why is the leadership good/what could be better?

**To what extent did the team know why agile was used?**

Would you say that you understand the point of working with agile?

Would you say that the other team members do?

Do you think there is a need for more education about agile within the team?

**To what extent was there an acceptance for working with agile within the team?**

Do you accept that agile is used?

Why do you think the team accept working with agile?

Why do you think that the team does not accept working with agile?

What do you think about the attitude about agile in the team?

**To what extent did the team collaborate?**

How do you collaborate within the team?

Do you consider this to have changed since you started working with agile?
Do you believe that the team should collaborate more or less?
Do you have any ideas on how to change this?

**To what extent was the goal of the process clear to the entire team?**
Is there a common goal that your team is working towards?

**To what extent was the process complex?**
Why is your process complex?
Should and could this be changed?

**To what extent were key stakeholders involved in the process?**
Which are your team’s stakeholders?
How do you collaborate with them in the process?
Is there a difference between different stakeholders?
Is it good that stakeholders are involved in the process?
How could the collaboration with stakeholders be changed?

**To what extent were the tasks dependent on each other?**
If dependent: Is this a problem?
Is it possible to make the tasks less dependent, how?

**To what extent did the process contain reoccurring tasks that could be standardized?**
What tasks are standardized today?
What tasks could be standardized?
How would this fit into the agile way of working?

**To what extent did the team members have formal roles?**
Can you describe the different roles in the team?
Do you believe that the roles should be more or less formal?
To what extent was the process characterized by an entrepreneurial culture?
Is it possible to take own initiatives, and do they have impact?
What types of initiatives are possible to take?
What was the latest initiative you took?
How does this work with your role in the team?

To what extent was the process operating in a "learning organization"?
In what ways are your process a “learning organization”?

To what extent did top management support the agile initiative?
In what ways is the agile initiative supported by management?
Is there any resistance from management regarding the agile initiative?
Would you want this to change? How?

To what extent was information accessible in the process?
What kind of information do you need in your work?
What makes it easy/hard to get access to information?
Would you want this to change? How?

To what extent was the process demanding in terms of workload?
What are the reasons for the process being demanding?
Do you think this is a problem?

To what extent was it possible to prioritize between the different tasks?
What makes it easy to prioritize tasks?
What makes it hard to prioritize tasks?
Do you have any ideas how this could be changed?

To what extent was the time estimation before a task and time consumption of tasks the same?
How does the time consumption normally differ from the time estimation?
What makes it easy to estimate tasks regarding time?
What makes it hard to estimate tasks regarding time?
Do you have any ideas how this could be changed?

**To what extent were support systems for agile used?**
What do you think about Jira/Confluence?
Should the use of support system work better?
Do you wish to have any other support?

**To what extent was the external environment of the process fast-changing or uncertain?**
In what ways are your external environment fast-changing or uncertain?

**To what extent was the internal environment of the process fast-changing or uncertain?**
In what ways are your internal environment fast-changing or uncertain?

**To what extent was the process flexible to adapt to change?**
In what ways are you flexible?
In what ways are you not flexible?

**To what extent was using agile in the process successful? To what extent do you consider agile methodology to have been the best way to work in the process?**
Motivate your answer whether agile has been the best way to work!
Do you believe that it is possible to change the agile methodology to reach a higher number?

**Last Questions:**
What do you want our master thesis to result in?
Is there anything you would like to add?
Appendix 2 - Survey with Other Companies and CS&A

Enablers for working with Agile

(Please read this before answering survey)

This survey is one part of gathering information for a master thesis examining the enablers of working with agile in a business context. It will take approximately 5 minutes to answer. The answers will only be treated internally within the study and are anonymous.

If you have any questions, comments, or would like to get a copy of the master thesis report once it has been published, please feel free to contact us at:

kaisa.alenmyr.468@student.lu.se or antonia.nilsson.345@student.lu.se

Please answer the questions according to the most recent time you worked with agile. If it is an ongoing process, answer accordingly to how the process has been so far. If there is no perfect answer, choose the option that is closest to reality.

* Required

1. How big was the team working with agile? *
   \[Mark\ only\ one\ oval.\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]

2. How was the team located? *
   \[Mark\ only\ one\ oval.\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]

3. What was your role in the project/process? *
   \[Please\ choose\ the\ alternative\ that\ suits\ the\ best.\]
   \[Mark\ only\ one\ oval.\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]
   \[\]

https://docs.google.com/forms/d/1YXaOiCeB68IUCnwpYd78ve5cMPT5aOf9YhtpmMrPzljDw/edit?usp=drivelink
4. Within what function(s) did you work with agile? *
   Please choose the alternative(s) that suits the best.
   Check all that apply.
   - Software Development
   - IT (but not development)
   - Product Development
   - Sales/Marketing
   - Legal
   - HR
   - Logistics/Purchasing
   - Administration
   - Operations
   - Finance
   - Other: _______________________________________

5. To what extent did the team make own decisions, independent from higher management?
   *Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
</table>
   Not at all | | | | | | | Completely

6. To what extent did the team have good and clear internal leadership?
   *Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
</table>
   Not at all | | | | | | | Completely

7. To what extent did the team know why agile was used?
   This means both the reasons for using agile in general and the reasons for using the specific techniques within it.
   *Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
</table>
   Not at all | | | | | | | Completely

8. To what extent was there an acceptance for working with agile within the team?
   *Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
</table>
   Not at all | | | | | | | Completely
9. **To what extent did the team collaborate?**
   *Mark only one oval.*
   
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. **To what extent was the goal of the project/process clear to the entire team?**
    *Mark only one oval.*
    
    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
    |---|---|---|---|---|---|---|
    | Not at all |  |  |  |  |  |  | Completely |

11. **To what extent was the project/process complex?**
    *Mark only one oval.*
    
    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
    |---|---|---|---|---|---|---|
    | Not at all |  |  |  |  |  |  | Completely |

12. **To what extent were key stakeholders involved in the project/process?**
    Examples of key stakeholder are customers, suppliers, partners, senior management, other departments within the company, etc.
    *Mark only one oval.*
    
    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
    |---|---|---|---|---|---|---|
    | Not at all |  |  |  |  |  |  | Completely |

13. **To what extent were the tasks dependent on each other?**
    For example: Was it crucial to follow a particular order for execution of the tasks?
    *Mark only one oval.*
    
    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
    |---|---|---|---|---|---|---|
    | Not at all |  |  |  |  |  |  | Completely |

14. **To what extent did the project/process contain reoccuring tasks that could be standardized?**
    *Mark only one oval.*
    
    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
    |---|---|---|---|---|---|---|
    | Not at all |  |  |  |  |  |  | Completely |
15. To what extent did the team members have formal roles?
   Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Completely</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

16. To what extent was the project/process characterized by an entrepreneurial culture?
   Entrepreneurial culture means that the team members are both authorized to and supported when developing and taking responsibility of own initiatives.
   Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Completely</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

17. To what extent was the project/process operating in a "learning organization"?
   A learning organization means that the organization always strives for the members to earn new knowledge.
   Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Completely</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

18. To what extent did top management support the agile initiative?
   Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Completely</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

19. To what extent was information accessible in the project/process?
   For example online systems that increase accessibility to information regardless of the physical location of it.
   Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Completely</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

20. To what extent was the project/process demanding in terms of workload?
   Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Completely</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
21. To what extent was it possible to prioritize between the different tasks?
Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

22. To what extent was the time estimation before a task and time consumption of tasks the same?
The time estimation does not have to be a “formal” estimation.
Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

23. To what extent were support systems for agile used?
Support systems can be software systems like Jira or Confluence, offline “systems” like Task boards or any other system supporting agile.
Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

24. Did the team have any kind of knowledge management system?
The system could both be a software system and some other kind of information storage system.
Mark only one oval.

- Yes
- No
- I do not know

25. To what extent was the external environment of the project/process fast changing or uncertain?
External environment = the environment outside the organization working with the project/process
Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

26. To what extent was the internal environment of the project/process fast changing or uncertain?
Internal environment = the environment inside the organization working with the project/process
Mark only one oval.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question</td>
<td>Scale</td>
<td>Rating</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>-------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27. To what extent was the project/process flexible to adapt to change?</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark only one oval.</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28. To what extent was using agility in the project/process successful?</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark only one oval.</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29. To what extent do you consider agility methodology to have been the best way to work in the project/process?</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark only one oval.</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30. To what extent do you believe that the rest of your agile team would have answered the survey in the same way you have done?</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark only one oval.</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not at all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Completely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31. Would you like to add something else about working with agility?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Powered by Google Forms

https://docs.google.com/forms/d/1XaoOCe6B6G/Czdp/15VVe5cMPT5a6bIYtpbMrP2hхоDw/edit?usp=drive_web
### Appendix 3 - The 41 Enablers Suggested by Conforto et al.

<table>
<thead>
<tr>
<th>Organization</th>
<th>Process</th>
<th>Project team</th>
<th>Project type and Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational structure type</td>
<td>Capability of reconfiguration</td>
<td>Self-direct teams</td>
<td>Product succession planning</td>
</tr>
<tr>
<td>Organizational culture</td>
<td>Process automatization</td>
<td>Team autonomy to make decisions</td>
<td>Urgency to complete the project (pace)</td>
</tr>
<tr>
<td>Entrepreneurial culture</td>
<td>Process modularity</td>
<td>Team leadership</td>
<td>Goal clarity</td>
</tr>
<tr>
<td>Learning organization</td>
<td>Easy access to information</td>
<td>Team dedication</td>
<td>Project complexity</td>
</tr>
<tr>
<td>Agile-style work environment</td>
<td>Formalization</td>
<td>Team knowledge about agile</td>
<td>Project newness</td>
</tr>
<tr>
<td>Acceptance of agile methodology</td>
<td>Frequent development milestones</td>
<td>Team experience/expertise</td>
<td>Support systems</td>
</tr>
<tr>
<td>Adequate reward for agile use</td>
<td>Process concurrency</td>
<td>Project manager experience</td>
<td>Customer involvement</td>
</tr>
<tr>
<td>Emphasis on speed</td>
<td>External integration</td>
<td>Team size</td>
<td>Collaborative work</td>
</tr>
<tr>
<td>Performance measuring</td>
<td></td>
<td>Team location</td>
<td>Suppliers’ involvement</td>
</tr>
<tr>
<td>Knowledge management systems</td>
<td></td>
<td>Multidisciplinary team</td>
<td></td>
</tr>
<tr>
<td>Multidisciplinary teams</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource competition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strong executive support</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decentralized decision making</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table adapted from Conforto et al. (2014)*
Appendix 4 - Results from Survey with Other Companies

62 responses

View all responses  Publish analytics

Summary

How big was the team working with agile?

- < 5 people: 11 (17.7%)
- 5 - 10 people: 36 (58.1%)
- > 10 people: 14 (22.6%)
- I do not know: 1 (1.6%)

How was the team located?

- In the same room: 29 (46.8%)
- On the same floor: 6 (9.7%)
- In the same building: 4 (6.5%)
- On the same site: 3 (4.8%)
- It was a virtual team: 19 (30.6%)
- I do not know: 1 (1.6%)

https://docs.google.com/forms/d/1XkQgheCfIiG8BUC1i7kgY8wGqSdJZMTP549yc7Up0rPd9qP0/p1/viewanalytics
What was your role in the project/process?

- Corporate manager: 3 (4.8%)
- Intermediate manager: 4 (6.5%)
- Team leader: 15 (24.2%)
- Team member: 27 (43.5%)
- Consultant for the agile initiative: 6 (9.7%)
- None of the above: 7 (11.3%)
- I do not know: 0 (0%)

Within what function(s) did you work with agile?

- Software Development: 54 (87.1%)
- IT (but not development): 10 (16.1%)
- Product Development: 12 (19.4%)
- Sales/Marketing: 8 (12.9%)
- Legal: 0 (0%)
- HR: 4 (6.5%)

https://docs.google.com/forms/d/1kXaoOe868UC2zqY9vKeXcMPT5af06YhptpmMrPzhpDw/viewanalytics
### Agile Management Outside of Software Development

#### Enablers for working with Agile - Google Forms

<table>
<thead>
<tr>
<th>Logistics/Purchasing</th>
<th>1</th>
<th>1.6%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration</td>
<td>3</td>
<td>4.8%</td>
</tr>
<tr>
<td>Operations</td>
<td>4</td>
<td>6.5%</td>
</tr>
<tr>
<td>Finance</td>
<td>4</td>
<td>6.5%</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
<td>3.2%</td>
</tr>
</tbody>
</table>

To what extent did the team make own decisions, independent from higher management?

<table>
<thead>
<tr>
<th>Rating</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4.8%</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>8.1%</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>9.7%</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>35.5%</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>32.3%</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>9.7%</td>
</tr>
</tbody>
</table>

To what extent did the team have good and clear internal leadership?

<table>
<thead>
<tr>
<th>Rating</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.6%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.2%</td>
</tr>
</tbody>
</table>

[https://docs.google.com/forms/d/e/1FAIpQLScMPT5a6Vif19tO6Y6ztb7m61pX9yypnMWpDow/viewanalytics](https://docs.google.com/forms/d/e/1FAIpQLScMPT5a6Vif19tO6Y6ztb7m61pX9yypnMWpDow/viewanalytics)
Enablers for working with Agile - Google Forms

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>9.7%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>19.4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>22.6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>25</td>
<td>40.3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Completely: 7 2 3.2%

To what extent did the team know why agile was used?

Not at all: 1 0 0%
2 3 4.8%
3 3 4.8%
4 7 11.3%
5 14 22.6%
6 15 24.2%
Completely: 7 20 32.3%

To what extent was there an acceptance for working with agile within the team?

Not at all: 1 1 1.6%
2 1 1.6%
3 1 1.6%
4 4 6.5%

https://docs.google.com/forms/d/1XaOce88UjCzdpY8veScMPT5af06v9pbnmtPjhiy0vw/viewanalytics
Agile Management Outside of Software Development

2016-05-08

Enablers for working with Agile - Google Forms

<table>
<thead>
<tr>
<th>5</th>
<th>11</th>
<th>17.7%</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>17</td>
<td>27.4%</td>
</tr>
<tr>
<td>Completely: 7</td>
<td>27</td>
<td>43.5%</td>
</tr>
</tbody>
</table>

To what extent did the team collaborate?

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>18</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>Not at all: 1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>18</td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>Completely: 7</td>
<td>11</td>
<td>17.7%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To what extent was the goal of the project/process clear to the entire team?

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>16</td>
<td>17</td>
<td>10</td>
</tr>
<tr>
<td>Not at all: 1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>16</td>
<td>17</td>
<td>10</td>
</tr>
<tr>
<td>Completely: 7</td>
<td>10</td>
<td>16.1%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://docs.google.com/forms/dfbY1kA0Ce6B8U1C2qdp19kveS/MPT5at06Ytbpww4W/cviewanalytics
To what extent was the project/process complex?

Not at all: 1  1  1.6%
2  1  1.6%
3  5  8.2%
4  11  18%
5  22  36.1%
6  13  21.3%
Completely: 7  8  13.1%

To what extent were key stakeholders involved in the project/process?

Not at all: 1  0  0%
2  9  14.5%
3  8  12.9%
4  7  11.3%
5  14  22.8%
6  15  24.2%
Completely: 7  9  14.5%

To what extent were the tasks dependent on each other?
Agile Management Outside of Software Development

**Enablers for working with Agile - Google Forms**

**To what extent did the project/process contain reoccurring tasks that could be standardized?**

- Not at all: 1 5 8.1%
- 2 22 35.5%
- 3 13 21%
- 4 11 17.7%
- 5 8 12.9%
- 6 3 4.8%
- Completely: 7 0 0%

**To what extent did the team members have formal roles?**

https://docs.google.com/forms/d/1YaOCe866UC1z5pY18veScMPT5a9691tptmIVtC1p5vViewanalytics
To what extent was the project/process characterized by an entrepreneurial culture?

Not at all: 1 1 1.8%
2 7 11.3%
3 6 9.7%
4 12 19.4%
5 12 19.4%
6 16 25.8%
Completely: 7 8 12.9%

To what extent was the project/process operating in a "learning organization"?

https://docs.google.com/forms/d/1YxaOCe86UJ1zop19veScMPT5a60bY1zptmmP2z6pG8v/viewanalytics
Enablers for working with Agile - Google Forms

| 3  | 9  | 14.5% |
| 4  | 8  | 12.9% |
| 5  | 12 | 19.4% |
| 6  | 13 | 21%   |
| Completely: 7 | 14 | 22.6% |

To what extent did top management support the agile initiative?

| Not at all: | 1 | 2 | 3.2% |
| 2 | 1 | 1.6% |
| 3 | 3 | 4.8% |
| 4 | 5 | 8.1% |
| 5 | 7 | 11.3% |
| 6 | 17 | 27.4% |
| Completely: | 7 | 27 | 43.5% |

To what extent was information accessible in the project/process?
Agile Management Outside of Software Development

To what extent was the project/process demanding in terms of workload?

<table>
<thead>
<tr>
<th>Rating</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.2%</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6.5%</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>9.7%</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>40.3%</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>27.4%</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>12.9%</td>
</tr>
</tbody>
</table>

Completely: 7

To what extent was it possible to prioritize between the different tasks?

https://docs.google.com/forms/d/1YaOCeB68UC1zdp18veScMPT5at69r8ptbmWz2hisp9dvw/viewanalytics
Agile Management Outside of Software Development

To what extent was the time estimation before a task and time consumption of tasks the same?

Not at all: 1 6 9.7%  
2 8 12.9%  
3 10 16.1%  
4 13 21%  
5 12 19.4%  
6 11 17.7%  
 Completely: 7 2 3.2%

To what extent were support systems for agile used?
Agile Management Outside of Software Development

2016-05-08

Enablers for working with Agile - Google Forms

Not at all: 1 1 1.6%
2 4 6.5%
3 4 6.5%
4 5 8.1%
5 17 27.4%
6 14 22.6%
Completely: 7 17 27.4%

Did the team have any kind of knowledge management system?

Yes 29 46.8%
No 22 35.5%
I do not know 11 17.7%

To what extent was the external environment of the project/process fast changing or uncertain?

https://docs.google.com/forms/d/1fXJaOCeBe66UC1zp19BveSclMPT5t0b8fIypbmmP2lhvpOzw/viewanalytics

12/17

110
To what extent was the internal environment of the project/process fast changing or uncertain?

Not at all: 1 1 1.6%
          2 6 9.8%
          3 13 21.3%
          4 12 19.7%
          5 17 27.9%
          6 6 9.8%
Completely: 7 6 9.8%

To what extent was the project/process flexible to adapt to change?

Not at all: 1 1 1.6%
          2 3 4.8%
          3 5 8.1%
          4 8 12.9%
          5 20 32.3%
          6 16 25.8%
Completely: 7 9 14.5%
To what extent was using agile in the project/process successful?

<table>
<thead>
<tr>
<th>Not at all</th>
<th>1</th>
<th>1.6%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>24%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>25%</td>
</tr>
<tr>
<td>Completely</td>
<td>7</td>
<td>17.7%</td>
</tr>
</tbody>
</table>

To what extent do you consider agile methodology to have been the best way to work in the project/process?

<table>
<thead>
<tr>
<th>Not at all</th>
<th>1</th>
<th>0%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1.6%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.2%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>8.1%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>11.3%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>32.3%</td>
</tr>
<tr>
<td>Completely</td>
<td>7</td>
<td>43.5%</td>
</tr>
</tbody>
</table>
To what extent do you believe that the rest of your agile team would have answered the survey in the same way you have done?

<table>
<thead>
<tr>
<th>Not at all</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3.2%</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.6%</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>11.3%</td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>33.9%</td>
</tr>
<tr>
<td>6</td>
<td>28</td>
<td>45.2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completely</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>3</td>
<td>4.8%</td>
</tr>
</tbody>
</table>
Appendix 5 - Long Text Answers from Survey

Would you like to add something else about working with agile?

Case against agile https://vimeo.com/110554082

Not only the team can work agile, stakeholders also need to understand and adapt to agile

Client involvement and iterative customer/user feedback is key

Often the developers has no problem adapting to agile, its the old large organizations that is afraid and rather do waterfall. It is a problem if trying to do agile while organization is stuck and you can't just tell them to do right, they need to slowly adapt to the agile way of working. For example in SCRUM you start up by having 3 month sprints and slowly making the sprints shorter and on the way educating the business.

Awesome, if you do it a 100%, sucks if you do it 95%...

There's a lot of people, including in my team, that say that they work and support agile. Meanwhile they try to add more and more processes, "manager"-roles and hierarchical structures to the project/organization which is the opposite of going agile. Why people like processes is because when something goes wrong nobody can blame a person as long as that person followed the process, which removes personal accountability. A lot of processes also creates way more overhead than you can imagine, sometimes so much so to support it's own "manager" role. Bear in mind that even though people work hard and long hours on a project with a lot of processes, nearly no real work (i.e. the project goal) gets done because everyone spends their time with the different processes/documents. The best enabler for being able to work with agile is to get rid of these people. The agile motto is: "People and communication over processes and documentation" and that is fucking difficult with these kind of people around. In short.

Our whole company is adopting scrum (before it was only development teams), even top management now have an agile board (in a corridor, clearly visible for everyone) and an agile coach.

Eliminate environment complexities and interdependent teams. Otherwise there are too many moving parts to leverage agile the way its intended... i.e. the skill set to implement a given feature is available on one team.

It is very useful in software development when new tasks and requirements appear during the project


Accountability for the end result needs to be covered via innovation accounting

otherwise agile sucks

Working agile in teams are rather easy, it is scaling and offshore that give a higher degree of complexity (known from experience)

Agile is great in certain circumstances but needs to be chosen carefully. The problem with agile is scope creep which can occur due to fluid timelines/deliverables.

Our delivery project was "surrounded" had dependencies with a number of large, traditional Waterfall projects - which caused predictable friction. The project budgeting process wasn't totally aligned with the agile process or had a clear grip of the pull vs push aspects involved. Business likes agile / Traditional cost drivers/budget by 'silo' likes waterfall projects.

Agile product development effort was "successful" from the viewpoint of the internal sales team members and external customers who got the alpha/beta releases within four and seven months of agile team formation. The final product released was an agile failure as the executive team delayed final release of the "beta" as a full production release for 12 months while an SVP hijacked the effort and grew a non-agile team of 200+ to replace the agile program group of about 13 members.

The project had a fixed list of goals, a fixed budget, and a fixed timeline. Agile was a completely inappropriate methodology to use under those constraints. It was clear from day 2 that the project was completely impossible to deliver. The project manager decided to use Agile as the best way to achieve as much as possible with the resources available. Total disaster.

It's key to remember to only use the parts of the method that work for the team.

Don't be a slave to the method.
Appendix 6 - Competence Card

Klara Klarnasson
Credit Risk Analyst
Credit Strategy & Analytics

I am good at:
☐ Excel
☒ SQL
☐ Analysis
☒ Geographic region: Nordics
☒ Other: Agile, IT

I want to do more:
☒ Excel
☐ SQL
☐ Analysis
☐ Geographic region: 
☐ Other: Presentations