Cargo Cult in Agile Software Development
~ To Jaxon ~
the light of my life
Örebro Studies in Informatics 22

TANJA ELINA HAVSTORM

Cargo Cult in Agile Software Development
Cover image: Dubois, Cargo cult in Papua New Guinea during the Pacific war

© Tanja Elina Havstorm, 2023

Title: Cargo Cult in Agile Software Development

Publisher: Örebro University 2023
www.oru.se/publikationer

Print: Örebro University, Repro 08/2023

ISBN 978-91-7529-517-6
Abstract


Software development (SD) projects are complex endeavors, and organizations employ software development methods (SDMs) both to add structure and to propel their projects. Nowadays, agile SDMs are the most used methods. Still, inefficient and ineffective use of SDMs is a challenge, and existing research shows cases of SD teams failing in their use of SDMs due to deviations made. Of course, methods should not be used in rigorous and textbook-like ways. Among practitioners, some of these deviations are referred to as cargo cult behavior; without, however, providing much of theoretical depth or any analytical tools to identify and characterize such behaviors. Although existing research has investigated SDM deviations, they have not been investigated as cargo cult behavior.

Against this backdrop, the aim of this thesis is to understand the challenges in succeeding with SDM use by developing a theory called SDM cargo cult theory (SDMCCT), to identify, analyze, and describe the cargo cult phenomenon in agile SDM use. The journey of this thesis takes its starting point in practitioners’ use of the buzzwords “cargo cult” when referring to flawed SDM use. By returning to the field of social anthropology and its studies of cargo cult, a definition of SDM cargo cult and an analytical framework are iteratively crafted through a longitudinal ethnographic study to constitute the SDMCCT. The research approach has been abductive and the SDMCCT builds on social action theory and work motivation theory.

The ethnographical study took place at an international industrial manufacturing company in Sweden that is using agile SDMs. This study includes three years of data collection, which includes observations, interviews, and gathering of business documents. The ethnographic study focused on three SD teams and their daily work using agile SDMs. The analysis using the analytical framework includes four agile SDM practices: daily scrum meeting, sprint demo, continuous integration, and visualization. In total, the analysis uncovered 36 deviations in the SD teams’ use of these practices, structured into 30 SDM cargo cult categories. In addition, this study shows that the framework is applicable to analyze and characterize effective SDM use as well, although it is not its main focus. The framework can be used by researchers to make similar analysis of cargo cult situations in other organizations, and the catalogue of cargo cult situations can serve as background knowledge for other organizations to study and improve their SD teams’ use of agile SDM practices.

Keywords: agile software development, cargo cult, deviations, ethnography, SDMCCT, social action theory, software development method, work motivation theory, theorizing

Tanja Elina Havstorm, School of Business, Informatics Department
Örebro University, SE-701 82 Örebro, Sweden, tanja.havstorm@oru.se
Acknowledgement

As I write these final lines in this thesis, with a strange feeling of losing a close friend who has been with me intensely for a very long time, I am sitting on the cliffs looking out over the ocean from the place I call home, at the surf lifeguard station in Tylösand. I cannot believe I have reached this long journey’s end. I remember being here precisely seven years ago, celebrating having been accepted as a doctoral student and enrolled in the Swedish Research School of Management and IT. What I did not know was what was coming. These years have been the most challenging, insightful, and developmental of my life, and I would like to take this opportunity to acknowledge the ones who supported me and contributed to this work in different ways.

Foremost I want to thank Fredrik Karlsson, my main supervisor, for your tremendous support throughout these seven years. Fredrik, you have through your profound wisdom and professional work, guided me to learn the essential means. Most of all, you never left my side, spent countless hours, and kept pushing me to achieve this end. Second, I want to give my gratitude to Karin Hedström and Shang Gao, my co-supervisors; thank you for all the insightful input and pep-talks along the road. I could not have wished for a better team as you both helped me grow as a researcher. Thank you to Kai Wistrand who saw my potential and encouraged me to step into the world of research and start this journey.

I want to thank Jenny Lagsten, Per Ågerfalk, and Björn Johansson for your input on my thesis through the proposal, mid-term, and final seminar, respectively. You helped me to steer this thesis in a good direction.

To all my co-workers at the informatics department and the research group CERIS; what a great place and group to be a part of. You all have become what I call my professional family, creating a place where I felt safe and happy even through the toughest times. A special thanks to Andreas Ask, Annika Andersson, Hannu Larsson, Ella Kolkowska, and Ann-Sofie Hellberg for your encouragement.

I want to thank the Swedish Research School of Management and IT, for the support you provided and opportunity you gave me. A network of great scholars has given me wisdom as a researcher. To all the fellow doctoral students, thank you for the enlightening discussions during our courses, but also the annual conferences.

To the software engineers at the Industrial manufacturing company where the study took place, thank you all for participating. It was only possible to carry out this study with your involvement.
To all my friends, you have a special place in my heart. Thanks especially to my dear friend, co-worker, and ‘partner in crime’, Elham Rostami, for all the time we spent together through tears and laughter, and the unforgettable memories we made. *This journey would never have been possible without you by my side.* Also, Josefin Nilsson, Sabina Eklund, Christina Nilsson, Josefin Palm, Jim Asprou and his family, and Louisa Isomäki stayed by my side and cheered me up, making sure that I kept a balance between work and life outside work. To Ako Zahir and Mohammed Al-Ali: I made it!

To my family, with whom I have the strongest bond one could ever imagine. My dear mother, Virpi Koskenranta, and father, Juha Mäki-Runsas, your endless love and belief in me is invaluable. You raised me safely and well, and I will never be able to thank you enough. My little sister, Nina Mäki-Runsas, thank you for all laughter and our beautiful memories. To my grandparents and my godmother Heidi Helosaari, I am forever grateful for your love and support.

Finally, I want to say some final words to my beloved son, Jaxon. You truly came into my life at the perfect time. You are the light of my life and I love you more than anything else in the world. *Forever you and me.*

*Ohana.*

Mahalo,

Tanja Elina Havstorm, June 12, Tylösand
Table of Contents

PART I – INTRODUCTION ........................................................................................................ 1
CHAPTER 1. INTRODUCTION TO THE DISSERTATION ........................................ 3
1.1. Software development methods .............................................................................. 4
1.2. Cargo cult in the software development field ..................................................... 6
1.3. Research purpose and research questions ......................................................... 7
1.4. Concepts used in the dissertation ....................................................................... 10
1.5. Summary of contributions ................................................................................. 11
1.6. Structure of the dissertation ............................................................................... 11

PART II – THEORETICAL CONCEPTS .......................................................................... 13
CHAPTER 2. CARGO CULT AND SOFTWARE DEVELOPMENT

METHOD .......................................................................................................................... 15
2.1. Cargo cult phenomenon......................................................................................... 15
2.2. Software development methods............................................................................. 19
   2.2.1. Software development method and goals ...................................................... 20
   2.2.3. SDM: adoption, use, adaption, implementation, and diffusion ............... 22
2.3. Traditional software development methods ........................................................ 23
2.4. Agile software development methods .................................................................. 23
   2.4.1. Agile values and principles .......................................................................... 24
2.5. Relevance of cargo cult phenomenon for SDM ................................................. 26
2.6. Related research – summary of paper I ......................................................... 27
CHAPTER 3. THEORETICAL FOUNDATIONS ......................................................... 31

3.1. Social action theory ............................................................................................. 31
   3.1.1. Rational social actions .............................................................................. 32
   3.1.2. Irrational social actions ............................................................................. 33
   3.1.3. Non-rational social actions ....................................................................... 33
3.2. Work motivation theory ....................................................................................... 35
   3.2.1. Extrinsic motivation .................................................................................. 36
   3.2.2. Intrinsic motivation .................................................................................. 37
   3.2.3. Amotivation .............................................................................................. 38

PART III – EMPIRICAL SETTINGS AND RESEARCH APPROACH ... 39
CHAPTER 4. EMPIRICAL SETTINGS ........................................................................ 41
4.1. Criteria for case selection .................................................................................. 41
4.2. Alpha – An industrial manufacturing company ................................ 41
4.3. Delta – The division studied ............................................................ 42
  4.3.1. Delta Sweden’s ASDM practices ............................................. 44
  4.3.2. The software development teams at Delta Sweden .................... 49
4.4. Summary of Delta between 2018-2020 ............................................ 52

CHAPTER 5. RESEARCH APPROACH ................................................ 53
5.1. Philosophical considerations ............................................................ 53
5.2. Abductive ethnography of social phenomena ..................................... 54
  5.2.1. Collecting empirical material ................................................... 55
  5.2.1.1. Access to Delta’s daily activities ........................................... 57
  5.2.1.2. Data collection techniques .................................................. 58
  5.2.2. Theorizing SDM cargo cult ...................................................... 61
  5.2.3. Data analysis ........................................................................... 66
  5.2.3.1. Dimensions of cargo cult analysis ......................................... 66
  5.2.3.2. Use of the SDM CC framework for data analysis ............... 67
5.3. Procedure of systematic literature review (SLR) ............................... 69
5.4. Credibility and transferability .......................................................... 71
5.5. Ethical considerations ....................................................................... 72

PART IV – FINDINGS............................................................................ 75
CHAPTER 6. SOFTWARE DEVELOPMENT METHOD CARGO CULT THEORY .................................................................. 77
  6.1. Initial definition and first version of the theory ................................ 77
  6.2. Second version of the theory .......................................................... 79
  6.3. Third version of the theory .............................................................. 80
  6.4. Fourth and final version of the theory ............................................. 82
    6.4.1. Refined SDM CC definition .................................................... 85
CHAPTER 7. IDENTIFIED SDM CARGO CULT ................................. 87
  7.1. Identified SDM deviations as cargo cult behavior ......................... 87
  7.2. Prominent pattern in the SDM cargo cult results ......................... 90
    7.2.1. Fragmental ............................................................................ 90
    7.2.2. Conventional ........................................................................ 91
    7.2.3. Deficient ............................................................................... 91
    7.2.4. Spontaneous .......................................................................... 92
    7.2.5. Conception ........................................................................... 92
CHAPTER 8. IDENTIFIED SDM NON-CARGO CULT ....................... 93

8.1. Identified non-cargo cult behavior .......................................................... 93
   8.1.1. Focus on upcoming tasks .......................................................... 94
   8.1.2. Identify hindrances ................................................................. 94
   8.1.3. Executing meetings ................................................................. 95
   8.1.4. Set directions .................................................................... 95
   8.1.5. Provide status report ............................................................. 96

PART V – CONCLUSIONS ......................................................................... 97

CHAPTER 9. KNOWLEDGE CONTRIBUTIONS AND IMPLICATIONS ................................................ 99

9.1. Implications for the SDM research community ................................ 99
9.2. Implications for SD domain community........................................ 103

CHAPTER 10. CONCLUSIONS AND FINAL REMARKS .................. 105

10.1. How can the cargo cult metaphor be used to identify, analyze, and describe challenges with SDM use? .......................................................... 105
    10.1.1. To what extent has reasoning behind adoption and use of SDMs been investigated in prior research? ........................................................ 106
    10.1.2. How can cargo cult be theorized to understand challenges in SDM use? .................................................................................. 106
    10.1.3. What analytical steps should be undertaken to uncover cargo cult in SDM use? ............................................................................... 107
10.2. Limitations .................................................................................. 107
10.3. Future research ............................................................................. 108

REFERENCES ...................................................................................... 111

PART VI – ADDITIONAL MATERIAL ................................................ 119

APPENDIX A – CONSENT FORM FOR DATA COLLECTION........ 121
APPENDIX B – INTERVIEW GUIDE .................................................. 123
APPENDIX C – OBSERVATION PROTOCOL ............................... 135
APPENDIX D – ORGANIZATIONAL DOCUMENTS ....................... 137
APPENDIX E – ETHNOGRAPHIC DOCUMENTS ............................ 139
APPENDIX F – THEORIZING PROCESS IN DETAIL ....................... 141

PUBLICATIONS IN THE DISSERTATION........................................ 145
List of Figures

Figure 1. Research questions and publications ................................................. 9
Figure 2. Definition of term cargo cult (Oxford Dictionary of Cultural Anthropology) ......................................................................................... 16
Figure 3. Software development method .......................................................... 21
Figure 4. The focus of existing research on the reasoning behind SDM adoption and use (Havstorm & Karlsson, 2023) ........................................ 27
Figure 5. Identified relationships between the research themes (Havstorm & Karlsson, 2023) .................................................................................. 29
Figure 6. The self-determination continuum (Gagné & Deci, 2005)................. 36
Figure 7. Alpha’s organizational structure ........................................................... 42
Figure 8. Delta’s roles and relations ................................................................. 44
Figure 9. Room setting - scrum meeting, Delta management team .............. 46
Figure 10. Room setting – scrum of scrum meeting, Beta scrum masters. 46
Figure 11. Room setting – team A, Delta Sweden ........................................... 50
Figure 12. Room setting – team B, Delta Sweden ........................................... 50
Figure 13. Room setting – team C, Delta Sweden ........................................... 51
Figure 14. IS research framework (Braa & Vidgen, 1999) ............................... 53
Figure 15. Research strategy and publications .............................................. 54
Figure 16. Observation protocol ....................................................................... 59
Figure 17. Notes from unstructured interviews during observations ............ 60
Figure 18. Process of abductive theorizing ...................................................... 63
Figure 19. Abductive theorizing in iterations ..................................................... 65
Figure 20. Dimensions of cargo cult analysis ............................................... 66
Figure 21. Selection process in structured literature review (Havstorm & Karlsson, 2023) .............................................................................. 70
Figure 22. SDM CC framework version 1 (Mäki-Runsas et al., 2019) 78
Figure 23. SDM CC framework version 2 (Mäki-Runsas, 2019) ................. 80
List of Tables

Table 1. Summary of contributions ................................................................. 11
Table 2. Summary of results presented in the cover paper ......................... 11
Table 3. Agile principles ............................................................................. 26
Table 4. Overview of research themes and types of reasoning (Havstorm & Karlsson, 2023) ................................................................. 28
Table 5. Future research directions and research questions (Havstorm & Karlsson, 2023) ........................................................................ 30
Table 6. Rational, irrational, and non-rational social actions.................... 32
Table 7. Role descriptions in Delta ............................................................... 43
Table 8. Delta’s ASDM practices ................................................................. 45
Table 9. Delta’s meeting overview ............................................................... 47
Table 10. Timeline .................................................................................... 52
Table 11. Follow-up interviews ................................................................. 61
Table 12. Summary of input and output from theorizing ....................... 64
Table 13. Inclusion and exclusion criteria in SLR (Havstorm & Karlsson, 2023) ................................................................................. 70
Table 14. SDM CC framework version 3 (Havstorm et al., 2023) ............ 81
Table 15. SDM CC framework version 4 (Havstorm et al., submitted) .... 83
Table 16. ASDM cargo cult result .............................................................. 88
Table 17. Identified SDM goal fulfillments .............................................. 93

List of Definitions

Cargo cult – *its original definition*
### Recurring abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ASD)</td>
<td>Agile software development</td>
</tr>
<tr>
<td>(ASDM)</td>
<td>Agile software development method</td>
</tr>
<tr>
<td>(CC)</td>
<td>Cargo cult</td>
</tr>
<tr>
<td>(XP)</td>
<td>eXtreme programming</td>
</tr>
<tr>
<td>(IS)</td>
<td>Information systems</td>
</tr>
<tr>
<td>(MR)</td>
<td>Method rationale</td>
</tr>
<tr>
<td>(PO)</td>
<td>Product owner</td>
</tr>
<tr>
<td>(SM)</td>
<td>Scrum master</td>
</tr>
<tr>
<td>(SDT)</td>
<td>Self-determination theory</td>
</tr>
<tr>
<td>(SAT)</td>
<td>Social action theory</td>
</tr>
<tr>
<td>(SD)</td>
<td>Software development</td>
</tr>
<tr>
<td>(SDM)</td>
<td>Software development method</td>
</tr>
<tr>
<td>(SLR)</td>
<td>Systematic literature review</td>
</tr>
<tr>
<td>(WMT)</td>
<td>Work motivation theory</td>
</tr>
</tbody>
</table>
List of publications

This thesis is based on the following publications, referred to in the text by their Roman numerals.


Note: All publications are reprinted with the authorization of the respective publishers.
PART I – INTRODUCTION
Chapter 1. Introduction to the Dissertation

In 2020, the success and failure rate among IT projects was being measured. Based on modern measurements, only 16% of IT projects were considered successful, i.e., the projects reached high satisfaction level of between 13-19%, while 9-20% met the level of value criteria (Standish Group, 2020). Consequently, in the world of IT project management, projects fail all the time, with failure rates of 19% on IT projects that did not meet both the criteria of satisfaction and of value level, and 65% rated as challenged (Standish Group, 2020); they failed much more often than one would expect. Based on the traditional measurements i.e., time, budget, and on target, between 2015-2020, only 33% of the IT projects were reported as successful (Standish Group, 2020).

Of those failed IT projects, a large amount failed despite the use of project management techniques (Standish Group, 2020). In the field of information systems (IS), particularly software development (SD), project management techniques consist of various methods that describe how to run IT projects, here called software development methods (SDMs). SDMs focus on managing IT projects and contributing to effective SD (Avison & Fitzgerald, 2003).

When SDMs are being used, it is often taken for granted that the projects and the use of SDMs will work out fine. In 2021, barriers in agile SDM adoption were reported by Digital.ai (2021). Three barriers, among others, were reported: inconsistencies in processes and practices (46%), lack of skills and experience (42%), and cultural clashes (43%) (Digital.ai, 2021). Agile culture has been reported as the biggest challenge to successful agile implementation (Agile Business, 2022). Digital.ai (2022) reports of dissatisfaction with agile being at 40%, as agile it clashes with company culture. The same report showed that 34% responded that they faced inconsistent agile practices and/or processes as a factor in what does not work well. Agile, which is associated with flexibility, collaboration, and leaness (Cram, 2019), implies an adaptability to respond rapidly to changes. Furthermore, agile is based on three fundamental pillars: transparency, inspection, and adaptation (Beck et al., 2001). “Agile is most commonly contrasted against traditional, plan-driven development methodologies” (Cram, 2019)

From the IT companies’ perspective, projects must be completed on time, within budget, and with desired features. The software delivered from these projects can be of many kinds. However, when considering how software is used in today’s society, we find them not only as part of support organizations’ back-office procedures. We find software integrated in highly complex systems, medical equipment, self-driving cars, and space rockets. In
these application areas, other aspects are equally crucial; for example, for the sake of the life or death of people. The latter could be related to development projects with high safety demands, where the developed software could create significant damage if anything goes wrong in the development process.

For example, in space rocket projects, each launch costs roughly billions of dollars: SpaceX’s Falcon 9 costs around $62 million, while NASA’s space shuttle had a cost of $1.5 billion to launch (H. Jones, 2018), meaning that there is much money at stake. Moreover, the project has a high risk of affecting the surroundings if anything goes wrong during a launch. Johnston and Harris (2019) compiled lessons learned for SD organizations due to several airplane crashes due to the upgraded Boeing 737 MAX software. They conveyed that it is important for practitioners to pay attention to the risks with software development since there are interrelationships between software engineering, human factors, corporate behavior, and customer service. Johnston and Harris (2019) said: “Most engineers have worked on fast-paced engineering projects and have had to make compromises to meet deadlines. Some compromises may have been personally initiated, while others were suggested or ordered by management”.

Thus, the software that runs these machines cannot be flawed. Therefore, SDMs are essential in project management and guiding software developers in any SD project, but even more so in projects with high safety demands.

1.1. Software development methods

During recent decades, SDMs have become central for SD organizations. The SDMs have been used to structure the SD processes. In general, an SDM, according to Avison and Pries-Heje (2007), is "a prescribed way of carrying out the development process".

However, the idea of SDMs has been introduced previously. In the late 1960s, the complexity of information systems increased, and they took longer than expected to develop. This resulted in higher costs and inefficient results, i.e., the SD could have been more efficient, and the final system rarely met the stakeholder’s requirements. The term "software crisis" was coined due to these problems in SD (Fitzgerald, 1996).

Therefore, practitioners and researchers put much effort into developing standardized methods to resolve the cost, time, and quality issues (Chuang et al., 2014). Those traditional, plan-driven SDMs were introduced around the 1970s and based on the waterfall model. They were meant to guide and help developers in SD processes; they became the claimed response to the "software crisis". Traditional SDMs focus on structured project plans and documenting the work. Ultimately, these methods strive for good
transparency, traceability, and improved management handling of the SD. Traditional SDMs consist of several phases, and each must be completed to continue to the next phase. In addition, these methods have been called "document heavy" since the communication between the phases relies on documents. The most common traditional SDMs, such as rational unified process, are based on the waterfall model (Royce, 1970). From a researcher's perspective, Fitzgerald (1996) described these methods as "increased control and the more widespread adoption of rigorous and formalized system development methodologies".

In 2001, the agile SDMs (ASDM) developed as a reaction to the traditional SDMs. The purpose of ASDMs is increased flexibility and faster delivery (Beck, 2000; Boehm, 1988; Cockburn, 2000; Consortium, 1997). The ASDMs' designers have agreed on directing the software developers toward four central values that are supposed to guide and influence the SD process. These values manifest as follows:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan. (Beck et al. 2001)

By having these values in mind, the developers indirectly suggested, in the Agile Manifesto, 12 main principles to follow when developing software (Beck et al., 2001). Dingsoyr et al. (2012) explained these principles as an encouragement to use different practices that accommodate rapid requirement changes at any stage of the development process.

These values show that an SDM represents rationality, which aligns with, for example, how Ågerfalk and Wistrand (2003) conceptualized SDMs. The organizational choice and use of SDM, whether it is traditional or agile SDM, is often regarded as a rational decision. It means that the business values and goals correspond with the chosen and used SDM. However, the decision could also be an irrational (inconsistent with the SDM goals) or even a non-rational choice or action (unawareness of the SDM goals) to use a specific method and its different parts.

Despite the importance of cohesion between the chosen SDM and the business goals and values (i.e., intended goals and values with the use of SDMs), and the fact that ASDMs have been on the research agenda since the start of this century, existing research has mainly focused on successful SDM adoption and use; empirical inquiries of less successful SDM adoption and use have not received much attention (Dybå & Dingsoyr, 2008; Gregory et al., 2016). The attention to less successful cases is minor, although it is possible to find research on, for example, deviations (Mortada et al., 2020; Stray et al., 2016). Still, cases of less successful SDM adoption
and use, i.e., where the SDM or its parts do not correspond to the business goals and values, have not received much attention (Dybå & Dingsoyr, 2008; Gregory et al., 2016). Interestingly, even less attention seems to have been placed on how software developers' use of these SDMs aligns with the goals of the methods (Havstorm & Karlsson, 2023; Havstorm et al., 2023; Mäki-Runsas et al., 2019).

Returning to today’s situation, SD organizations still need help managing their SD processes; despite adopting ASDMs. In 2011, ASDMs were well-known in the IS field (Laanti et al., 2011), and several benefits have been reported in comparison to the traditional SDMs. At the same time, SD projects were reported to face difficulties in meeting their targets when adopting and using various kinds of SDM (Standish Group, 2000, 2008, 2012, 2020). Furthermore, customers continue to require faster delivery of working software, putting higher demands on their return on investment (ROI), and the pressure on SD organizations has increased (Chuang et al., 2014). The demands from the customers have also affected software developers' use of SDMs.

1.2. Cargo cult in the software development field

As discussed above, the IT industry reports on IT projects having low success rate using agile methods. Practitioners from the IT industry, such as Soria (2012), highlighted in a blog focusing on agile training and coaching some experiences faced by consultants in the IT industry. The blog consisted of practical situations presented as habits of faulty ASDM use. For example, "Replicate (wrong or incomplete) past circumstances, trying to obtain the same outcome," "Ignore how the practice works," and "Pay attention only to the success and ignore failure" (Soria, 2012).

These situations were referred to as “cargo cult behavior” or, later, as “cargo cult agile”. The topic has emerged in warnings by consultants who provide, for example, checklists on different webpages to review the state of agility in organizations to identify if the organization faces these "cargo cult agile" situations (Wolpers, 2020).

“Cargo cult programming” has also been observed in situations among practitioners as: “Cargo cult programming occurs when a programmer ritually includes code in a system, despite not understanding how this code interacts in said system” (M. Jones (2018).

Recently a practitioner, den Uijl (2023), working as an agile coach, posted on LinkedIn an experience from the IT industry about the ongoing discussion on cargo cult agile: “Organisations, leaders and teams that do not fully embrace the underlying principles of open and honest collaboration, customer focus and iterative development will not reap the benefits”. To gain readers' attention, the post was introduced with a question: “It sure looks agile; we are following all the steps; where are the planes we were
promised?”(den Uijl, 2023). The quote about the planes refers to the history of CC, where the Western army landed on the Melanesian islands during the Pacific War, as the islands' natives observed mysterious guests and how it resulted in "cargo" being delivered by ship or air (Worsley, 1957).

These abovementioned situations have the characteristics of buzzwords but pinpoint, among other things, situations where software developers involved in IT projects lack understanding of the SDM and faulty ASDM adoption (Shore, 2008; Soria, 2012). At the same time, the cargo cult is a borrowed concept with its origin in the field of social anthropology. It refers to a collective behavior when a group tries to imitate and perform certain rituals to reach the same outcome and success as others without adequately understanding the underlying reasons, i.e., goals and values, behind the actions (Worsley, 1957).

Later on, in 1974, the concept appeared in the scientific setting by Feynman. He tried to shed light on what he identified as a common situation among researchers: that scientific researchers used research methods without proper understanding of them, but did as the other researchers did (Feynman, 1974).

Thus, in the SDM context, faulty SDM adoption and SDM use occur when SD organizations have tried to adopt or use parts or entire SDMs with the hope of reaching the same success as others but failed to do so. It also aligns with the examples provided by the practitioners, such as (Soria, 2012), who have argued that there are situations when developers claim to be agile but are not doing agile. Furthermore, prior research has shown cases with successful adoption and use of SDMs (Heikkilä et al., 2013; Häggmark & Ågerfalk, 2006). Other studies have identified deviations, such as anti-patterns (Eloranta et al., 2015). Anti-patterns pinpoint situations where software developers deviate in their use of ASDMs practices. The deviations describe what has been carried out but not the reasons behind the actions.

Thus, these deviations have not been researched from the perspective of understanding the reasoning behind them, nor from the cargo cult metaphor perspective. This dissertation, therefore, elaborates on the cargo cult metaphor used by practitioners, and hereafter called CC, by uncovering what is behind these SDM deviation situations.

1.3. Research purpose and research questions

Despite the benefits that the ASDMs can contribute compared to the traditional SDMs, as shown above, there is still an ongoing discussion in the IT industry about the less successful use of ASDMs. Furthermore, the discussion above also shows that there is a scientific need for research focusing on these challenges, particularly software developers’ reasoning behind SDM
use. Therefore, this dissertation views CC as one potential challenge in deviations that may hinder an organization’s effective use of ASDMs.

Against this background, this study aims to develop a theory called SDM CC theory (SDMCCT), to identify, analyze, and describe the presence of the CC phenomenon in ASDM use. To achieve this end, this dissertation uses the cargo cult metaphor as a starting point to explain the phenomenon of flawed ASDM use. The suggested theory can be used to develop an increased understanding of challenges with SDM use. As will be shown in Chapter 6, the SDMCCT consists of a definition and a framework that includes ideal stereotypes. Furthermore, the SDMCCT is developed by combining two borrowed theories: social action theory (Weber, 1947) and work motivation theory (Gagné & Deci, 2005), and is iteratively refined using empirical material from an ethnographic study. SDMCCT can be used to uncover what is behind the metaphor of CC, i.e., the buzzword used by practitioners, when describing less successful use of ASDMs.

To propose a theory to complement previous research, the study will answer the overall research question:

**RQ: How can the cargo cult metaphor be used to identify, analyze, and describe challenges with SDM use?**

To complement the overall research question, three sub-questions were devised as illustrated in Fig. 1. The figure comprises the overall research question and sub-questions and illustrates how each publication addresses and contributes to specific sub-questions.

**SubRQ1: To what extent has the reasoning behind the adoption and use of SDMs been investigated in prior research?**

In order to theorize CC in the SDM context, there is a need to know what previous research on SDM use has focused on. The results from reviewing existing research are an essential input for the theorizing process and for better positioning the results of this thesis. Therefore, the scope for this sub-question was broadened, and investigating previous research and its focus (or lack of focus) on reasoning in SDM adoption and use; a structured literature review was used to provide such an overview. The reason for using both the concepts of adoption and use in the literature review is because existing research tends to use both these concepts simultaneously.

The overview further helped to understand what was needed to define the CC concept in the context of SDM. The overview of the current knowledge base of the scientific field is presented in Publication I in Figure 1 below, and the research question is answered in Chapter 2.5. “Related research – summary of Paper I”
SubRQ2: How can cargo cult be theorized to understand challenges in SDM use?

The phenomenon needed to be defined and placed in the SDM context to study the phenomenon of CC. In this dissertation, the context is the field of SDM. Therefore, sub-question number two focuses on the development of the SDM CC theory. As illustrated in Fig. 1, theorizing and its output can be found in publications II-V. Furthermore, empirically validating the developed theory is important. Therefore, the theory has been validated using the framework to analyze data from the ethnographic study. The validation is found in three publications: III, IV, and V.

SubRQ3: What analytical steps should be undertaken to uncover cargo cult in SDM use?

To identify, analyze, and describe SDM CC, a definition and a framework are not enough. Future users of the SDMCCT need to know how to use the framework to uncover potential CC situations. Thus, there is also a need for a set of analytical steps that describe how to execute this type of analysis. Analytical steps are presented in section 5.3.3.2. How these steps can be used to analyze CC in SDM is elaborated in publications III, IV, and V, where empirical data was analyzed using the SDM CC framework proposed in this dissertation.
1.4. Concepts used in the dissertation

This dissertation, both this cover paper and the appended papers, includes several concepts. Some of the publications use more than one concept to refer to the same things. A short description of the concepts and how these are used will be presented in this section.

Deviations – SDM practices performed by software developers in another way than first expected.

Cargo cult phenomenon – reflects faulty SDM adoption and SDM use when SD organizations have tried to adopt or use parts or entire SDMs, with the hope of reaching the same success as others but failed due to irrational or lack of reasoning behind their actions.

Cargo cult situations – situations identified in software development teams’ recurrent actions, in their use of SDM practices being deviations analyzed and classified as cargo cult situations by SDM CC framework. A cargo cult situation reflects failed use of an SDM practice, i.e., software development teams do not achieve the intended goals and values with their SDM use.

Non-cargo cult situations - situations identified in software development teams’ recurrent actions in their use of SDM practices, being analyzed and classified as non-cargo cult situations by SDM CC framework. A non-cargo cult situation reflects achieved goals and values with the intended use of an SDM practice.

Software development method (SDM) – in two of the publications “information systems development method” has been used. This cover paper uses “software development method”.

Reference point – software development teams’ SDM description, i.e., their intended goal with their SDM use.
1.5. Summary of contributions

This dissertation has two main contributions and one contribution each for the research domain and the SD domain community, respectively. These are summarized in Table 1. The main contributions are shared both for research domain and for SD domain i.e., practitioners. Each of these contributions are presented further in Chapter 9.

<table>
<thead>
<tr>
<th>Table 1. Summary of contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main contributions</strong></td>
</tr>
<tr>
<td>• A theory SDMCCT for identifying and explaining CC phenomenon in the SDM context.</td>
</tr>
<tr>
<td>o A definition of SDM CC</td>
</tr>
<tr>
<td>o A framework for analyzing SDM CC situations.</td>
</tr>
<tr>
<td>o A set of eight stereotypes capturing CC and non-CC situations.</td>
</tr>
<tr>
<td>• Catalogue of identified SDM CC situations.</td>
</tr>
<tr>
<td><strong>Research domain contributions</strong></td>
</tr>
<tr>
<td>• A data analysis method for identifying SDM CC.</td>
</tr>
<tr>
<td><strong>SD domain contributions</strong></td>
</tr>
<tr>
<td>• Advice on where to focus the attention on SD teams’ SDM use.</td>
</tr>
</tbody>
</table>

1.6. Structure of the dissertation

This dissertation consists of a cover paper and a collection of five research papers. The results of this thesis are presented in five chapters, summarized in Table 2. Chapter 2, section 2.4, presents an overview of previous research by summarizing the literature review conducted as a background to this thesis. Further on, in Chapter 5, section 5.3.3.2., a research method contribution can be found, in terms of the analytical procedure on how to use the SDM CC framework. Chapter 6 consists of theoretical contributions i.e., the results from the theorizing, presenting the SDMCCT (including the SDM CC definition, and the SDM CC framework consisting of eight stereotypes). Chapters 7 and 8 present the empirical results divided into two chapters. Chapter 7 focuses on identified CC situations, and Chapter 8 presents identified non-CC situations i.e., successful use of SDMs.

<table>
<thead>
<tr>
<th>Table 2. Summary of results presented in the cover paper</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of result</strong></td>
</tr>
<tr>
<td>Overview of previous research, and summary of literature review.</td>
</tr>
<tr>
<td>Analytical procedure i.e., how to operationalize the SDM CC framework.</td>
</tr>
<tr>
<td>The SDM CC definition and SDM CC framework i.e., results of the theorizing.</td>
</tr>
<tr>
<td>Identified CC situations, i.e., empirical findings from ethnographic study.</td>
</tr>
<tr>
<td>Identified non-CC situations, i.e., empirical findings from ethnographic study.</td>
</tr>
</tbody>
</table>
The remainder of the cover paper is structured into the following sections.

Chapter 2 presents an overview of related research on the CC phenomenon, SDM, and agile SDM. Finally, the chapter ends with a summary of Paper I, a structured literature review of empirical studies on rationality in SDM adoption and use.

Chapter 3 presents an in-depth description of this dissertation's theoretical foundations and theories, i.e., the phenomenon of CC, social action theory, and work motivation theory.

Chapter 4 presents the empirical settings by describing the context of the organization that has been studied in this dissertation.

Chapter 5 positions the dissertation and outlines its ontological and epistemological assumptions, which are then used to justify the methodological choices. Results from validated analytical steps on using the framework are also presented as a summary of papers II-V.

Chapter 6 presents results from the theorization of SDM CC theory, which is presented as a definition and a framework. This chapter summarizes parts of papers II-V that focus on theorization.

Chapter 7 presents the first of two parts of the empirical findings from the ethnographical study and summaries of identified deviations and cargo cult behavior, presented in papers III-V.

Chapter 8 presents the second out of two parts of the empirical findings from the ethnographical study and summaries identified SDM non-deviations and non-cargo cult behavior, which are additional findings that have not been presented in any publications.

Chapter 9 discusses the knowledge contributions and implications of this dissertation.

Chapter 10 presents this dissertation's conclusions, limitations, and suggestions for future research.

The five research papers are appended after the cover paper.
PART II – THEORETICAL CONCEPTS
Chapter 2. Cargo Cult and Software Development Method

SDMs have been developed for a long time as they already appeared as a response to the software crisis in the 1960s (Brooks, 1995). The SD organizations and their SD projects greatly needed structure and guidance in their development processes to produce working software. Organizations' needs resulted in SDMs with clear process instructions (Avison & Fitzgerald, 2003). These processes' lifecycle became linear with the "waterfall approach," with strict steps to plan and capture all possible components included in one SD project (Russo & Stolterman, 2000).

The linear way of working is still applicable in several organizations. However, for many SD organizations, an iterative and incremental approach has gained much more attention and is the most used approach today (Russo et al., 2013), the so-called ASDMs.

As mentioned in Chapter 1, SD organizations have faced challenges in succeeding with IT projects. Industry reports point, among other things, toward the challenges of using SDMs (Standish Group, 2020), as well as practitioners highlighting buzzwords, such as “cargo cult”, to label the faulty use of ASDMs.

This dissertation focuses on the notion of ‘cargo cult concerning SDMs. Therefore, this chapter will first provide an overview of the cargo cult phenomenon, which is the primary concept in this dissertation, to understand its origin; then provides an overview of SDMs and the different approaches to SD that organizations can use, with a primary focus on agile methods. After describing these two central concepts, CC and SDM, the following section explains the relevance of CC in SDM. Finally, this chapter ends with a summary of a literature review of prior research on empirical studies on software developers’ reasoning behind SDM adoption and use.

2.1. Cargo cult phenomenon

This section will go deeper into the term cargo cult, and its history from the scientific field of social anthropology, thereafter, connecting the term in SD organizations. The cargo cult term has been investigated over the years from different perspectives, making social anthropologists debate it. First, I introduce the definition from the Oxford Dictionary of Cultural Anthropology, in Fig. 2.
Cargo cult

A term originally from Melanesia that refers to a social movement during the nineteenth and twentieth centuries throughout the South Pacific in which native prophets promised the imminent arrival of 'cargo,' or manufactured goods and money if local people were to purify and harmonize themselves. During and after World War II, these movements gained a high international profile, and anthropologists explained them as a universal millenarian movement. By the 1970s and 1980s, anthropologists began to emphasize that they were grounded in local societies' cultural creativity and assertiveness in response to social changes brought by modernization and globalization. Many cargo movements still exist, as political organizations or in churches.

The identified social movements governed by cultural creativity are responses to different social changes worldwide. These movements have a common term: cargo cult. If one drew a timeline of the CC term, it would, according to Worsley (1957), start around the year 1935, when cargo movement was first reported. However, in Fiji, the cults began as early as 1885 (Kaplan, 1995; Worsley, 1957). Worsley (1957) conducted a study by interpreting the various reports on the social movements in Melanesian Islands. These reports consisted of studies done by anthropologists who cited the native people on the different islands as they “used religious and magical symbols derived from both Christianity and indigenous belief” (Worsley, 1957).

Thus, between the wars in the Pacific, the natives on the islands got visits from the Western army. As the army landed on the Melanesian Islands to station their equipment (cargo) the native people became inspired by them and their equipment, i.e., airplanes, weapons, and artillery. The natives saw the equipment as goods that came from the sky.

As the army continued its journey towards Japan, the natives on the island began to create equipment that had the same shape as that of the army’s, this fostered social change. The native believed that if they created equipment similar to that of the Western army and marched around in the same way, they would, in the end, become like the Western army, having the same goods and being able to do the same things, i.e., fly up in the sky with airplanes. Most importantly, Worsley concluded that the natives felt insecure due to social and economic conditions changing as the Westerners entered their villages. The millenarian movements became the symptom of the natives’ attempts to “assert their independence” (Worsley, 1957).

Other anthropologists have identified and discussed other similar movements over the years though, Worsley was, according to Kaplan (1995), building a theory that the context of cults was political-economic. As the
Oxford dictionary above defines, the Millenarian movements arose after World War II. Later, during the 21st century, the term CC arose in the IT industry as a buzzword, pointing towards less successful adoption and use of ASDMs (Shore, 2008). The buzzword has also been used as an input in agile training for practitioners to provide potential causes of failure in incorrect adoption and use of the ASDMs (Soria, 2012).

The American Anthropological Association presented a collection of papers in 1996. The call for papers was due to both questioning the term of cargo cult and investigating the inner workings of what appear to be "classic" CC activities, e.g., collectively performing rituals during ceremonies (Dalton, 2000b).

In Lindstrom (2000), the term CC was used to quash ingenious political movements that threatened the colonial state's power authority and gave examples of different “cargo horror stories” and “carnival cargo”. The term is also tightly associated with political and economic power, such as various political and economic activities and movements. However, the term has its historical origin in descriptions of suppressing legitimate native endeavors that threatened white colonial domination (Dalton, 2000b).

Dalton (2000a) discussed the recurrence of CC behavior as a specific “madness” which continues to inhabit CC discourse. He saw behavior as a liminal phenomenon that includes absurd and unworkable beliefs and behavior as fundamental. Therefore, he argued that “it needs to be comprehended in a way other than our usual rationalistic argument through dependency orders” (Dalton, 2000a). Western rationalism compelled the Melanesian mode of knowledge-power as a mimetic parody. The CC is viewed as a memetic phenomenon, which has a parodic enactment of particularly absurd aspects of Western colonial bourgeois culture with the image of unlimited good in its projects. Dalton further referred to Buck (1989), where CC was described as: "Cargo cult is a Western colonial category that serves to blame indigenous peoples for their own underdevelopment" (Dalton, 2000a).

The term “cult” has been viewed by some as religion "without power", according to McDowell (2000), since the term has been applied to human behavior and beliefs as negativity, otherness, or difference. When comparing a phenomenon or an object, it is most likely that its differences and similarities will be compared. However, only its differences have been treated when CC has been in focus (McDowell, 2000).

In most of the CC literature, the writers have used the "we" to compare and describe phenomena. McDowell (2000) emphasized that "we" needs to be more specified by anthropologists. Rationality and irrationality are terms frequently used with CC, but according to McDowell (2000): "without much explicit exploration of either their meaning or the consequences of their usage" (McDowell, 2000).
In Wagner (2000), disinformation and deniability are compared with CC, and these concepts somehow describe the world in general. There are theories about CC, and CC about those theories, according to Wagner (2000). From an abstract perspective, they investigate whom we are supposed to be or what we are supposed to be. That could also involve the public or scientists and lead to another type of CC. He related this phenomenon to social science.

For example, if software developers prepare for a coming sprint but do not know where to begin their development, it is not their fault that they do not find the backlog, as is supposed to be communicated from the product owner and the scrum master. This implies that if the software developers develop outside the intended scope of the sprint, it is due to misinterpretations of information and the end goal. It does not imply deniability, as the software developers had faulty instructions.

If we relate to the case of software developers developing solutions outside the scope of the sprint, deniability, and disinformation to this and having the wrong instructions, we could investigate misconceptions and misinterpretations in this disinformation. From a philosophical standpoint, Wagner (2000) said: "How one knows what one sees as a function of how one sees what one knows" Wagner (2000).

The reality of someone who is within a particular group is what they see and what they think, and what they believe. If someone from the outside looks at them, they might have another reality from their perspective. So, it is essential to understand that for CC to be viewed as something regarding information, we need to understand what the group, in this case, sees. Some or most cargo movements are secret forms of professional endeavor that have gone wrong. Wagner (2000) stated: "Caught somewhere between the objectification culture and conservation of the technological object, carnival called has grown to be something of a Loch Ness Monster" Wagner (2000).

It is not just Wagner (2000) who described the CC using a monster definition. Lindstrom (2000) also described cargo "culting" as some horror. The post-Mambu text was the fear monstrosity of cargo related to Frankenstein. So in a way, "cargo culting leads horribly to hell", and "cargo culting turns people into monsters" Lindstrom (2000).

All of these come from some mythical dreams, and we could also see some descriptions of cargo as something good; the "culting" part is the bad part. In Frankenstein, we can see in his history that he tries to have some self-hate and decides to be something other than himself. That has been the case; the cargo culting was a symptom, and Frankenstein wanted to have the cargo now, not to have the same things anymore; he wanted to create something new and better.

We know that people want to have cargo in the moment and do not want old and dysfunctional things (such as old processes and SDMs in SD)
anymore. The end time when they are not satisfied with this cargo is because the cargo has somehow failed, as the Melanesian natives found (Worsley, 1957). In SD, the desire of changing the dysfunctional processes and SDMs implies that the intent to reach the new ways of working have failed.

If we compare this to the Melanesian Islands, it is the case that even when the planes land and the cargo does arrive, these goods somehow fail to satiate the cargo desire, and eventually lead to unhappy endings. The unhappy endings are examples of when there are groups who want to become cargo people.

Furthermore, Lindstrom (2000) also said that we, in this century, are cargo people right now; we do not want “old stuff” anymore. Moreover, in the end, all this new stuff will later not satisfy us. After that, we will seek new cargo, and humans will not be that happy even though we have all these new things, because we will never be satisfied.

2.2. Software development methods

SD is a process where software (or, in systems development, information systems) are being developed. This dissertation will further not distinguish between these terms; "software" will include "information systems".

Within the spectrum of SD, several frameworks, methods, and models describe how these processes should be facilitated. Each approach has in common organizing an effective SD process by reducing the project’s complexity to become manageable (Avison & Fitzgerald, 2003). An SDM is a prescribed description to guide software developers’ development process. “One of the basic tools that practitioners need in order to accomplish their task is a well-defined engineering process laid out by a software development method” (Hazzan & Dubinsky, 2014).

Scholars have invested much effort into understanding and explaining what an SDM is. As a result, it is possible to identify several definitions (Brinkkemper, 1996; Hazzan & Dubinsky, 2014; Palvia & Nosek, 1993; Russo & Stolterman, 2000; Wynekoop & Russo, 1995; Wynekoop & Russo, 1997). If we look at these definitions, from 1993 we find a definition of SDM as follows: “An organized and systematic approach to systems life cycle or its parts. It will specify the individual task and their sequences” (Palvia & Nosek, 1993).

Palvia and Nosek emphasized the sequences of task performance, while Brinkkemper put emphasis on the perspective and systematic performance: “A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products” (Brinkkemper, 1996).
The systematic performance was in focus even in 1995 and 1997 in combination with development being viewed as a set of phases carried out in a specific order:

A systematic approach to conducting at least one complete phase (e.g., requirements analysis, design) of systems development, consisting of a set of guidelines, activities, techniques and tools, based on a particular philosophy of systems development and the target system. (Wynekoop & Russo, 1995; Wynekoop & Russo, 1997)

Russo and Stolterman (2000) used the definition of SDM from Wynekoop and Russo (1995) and added:

Information systems methodologies are promoted as a means of improving the management and control of the system design and development process by specifying and standardizing the activities to be performed and the documentation to be produced. (Russo & Stolterman, 2000)

If we look at the research from the 21st century, we can identify a similar SDM definition from 2014, which also emphasizes the roles and behavior:

A software development method is a set of activities and practices, as well as roles and norms of behavior, derived from a set of professional aims, which are carried out in a logical and specified order. (Hazzan & Dubinsky, 2014)

Karlsson and Ågerfalk (2004) concluded that even though existing definitions are slightly different, there seems to be a shared understanding that an SDM consists of three fundamental parts: (1) activities, i.e., operationalized activities, a process making sure developers are guided in their performance; (2) notations, i.e., documented results of the activities, and (3) concepts, i.e., what is vital in the SDM, description of the problem domain. Furthermore, they argued that an SDM's content is based on a method designer's decisions on what is essential, and it gives a description for systems developers that tells them what to do and how they should develop software (Karlsson & Ågerfalk, 2004).

2.2.1. Software development method and goals

The SDM concept, illustrated in Fig. 3, builds on a perspective including goals and values (Goldkuhl et al., 1998), which is essential for an SD team to have a common understanding so as to achieve a fruitful adoption and use. Furthermore, perspective is referred to as method rationale (Ågerfalk & Wistrand, 2003).
Since the content of SDMs is the result of the method designer’s decisions concerning the method, SDMs’ activities, notations, and concepts exist because of what the method designer considers essential. For example, Jayaratna et al. (1999) have argued that: “if […] the rationale for the action is implicit then by definition the activity set cannot be considered a methodology”– (Jayaratna et al. 1999).

So, these reasons are an essential part of SDMs, according to Goldkuhl (1998) and expressed as part of the SDM’s perspective.

Similar ideas are also found in Brinkkemper (1996). Hence, they all acknowledged method rationale as an essential part of SDM. Rossi et al. (2004) have defined method rationale with the meaning of all choices made when designing the SDM and the reasons for these choices; Ågerfalk and Wistrand (2003) have made a less inclusive operationalization, with the meaning of explicating goals and values behind the SDM. This dissertation uses the less inclusive notion of method rationale.

Regardless of the chosen operationalization of the method rationale concept, it can be used to analyze the rationale expressed in a method artifact. For example, Ågerfalk (2006) used the less inclusive notation to analyze the rationale in the Agile Manifesto. He identified 12 different goals anchored in five different values.

Method rationale has also been used as a starting point for situational construction or tailoring of SDMs. For example, Rolland and Prakash (1996) and Ralytė et al. (2003) used the concept of intention, while Brinkkemper (1996); Gonzalez-Perez et al. (2007); Harmsen (1997); Karlsson and Ågerfalk (2004); (Karlsson & Ågerfalk, 2009) used the goal concept.

Thus, the abovementioned research has mainly focused on the method artifact and has paid less attention to how software developers perceive SDMs and their rationale. However, the software developers must be aware
of the goals and values of the used SDM. Otherwise, they risk using the SDM's activities, notations, and concepts based on wrongful assumptions and/or with wrong interpretations (Conboy, 2009).

2.2.3. SDM: adoption, use, adaption, implementation, and diffusion

There are various concepts within the area of SDM research that link to this thesis. This section describes these concepts related to this dissertation's main focus, i.e., the use of SDMs.

When SD organizations are about to make a change in their SD process, the SD organizations introduce SDM. An organization could, in those cases, adopt one or more SDMs to support their SD process; then the organization management has decided to follow the SDM prescriptions. The software developers in the organization are the ones who use these SDMs by having certain SDM activities to pursue.

The SDMs are possible to adapt and implement in an organization. Looking at how innovation has spread within a group or community is referred to as diffusion of an SDM, i.e., how an SDM has been spread. Diffusion could also describe how the process of an SDM adoption went.

Gregory et al. (2016) stated limitations in empirical research on the challenges and shortcomings of agile SDM adoption and use; instead, the field consists of studies on successful use and adoption of SDMs. When Gregory et al. (2016) studied practitioners' experiences with agile SDMs, they identified different challenges. Since agile SDMs are supposed to be open and possible to adapt, practitioners need help adopting and using these SDMs properly. They found that the challenges could be based on misconceptions, not having a common understanding in the organization, lack of having the "right" environment or commitment, problems with the organizational culture, or difficulties in changing the mindset.

In this dissertation, SDM adoption is defined as when SDMs are about to be introduced to the SD organization, and use as when SDMs are used by software developers, and when the SD organization aligns with the intended business values and goals of introducing the SDM. For example, when companies achieve their planned strategy, an organization can succeed with the SD project, but achieving the planned strategy differs from meeting their business values and goals with their SDM. Therefore, when successful or unsuccessful scenarios are described further in this study, the scenarios refer to SDMs being introduced.

Fitzgerald et al. (2002) comprehensively explained the complex nature of information systems development and its use. They developed a framework called method-in-action that focuses on using formalized SDMs, such as rational unified process (RUP), scrum, and extreme programming (XP). The formalized SDMs could also include the one that an organization customizes by itself. Formalized SDMs are the fundament of how they will execute
in action. Fitzgerald et al. (2002) described these SDMs as the basis for method-in-action, but this does not mean that the SDM must be strictly followed. On the contrary, an SDM is adapted to suit developers, the development context, and the information system. The cause of this could be the nature of the project and different interpretations of individual developers on how to apply the SDM.

2.3. Traditional software development methods

One approach to SD is the plan-driven; one example of a plan-driven SDM is the waterfall. A waterfall SDM is one where software developers develop software by planning and executing the project in five separate phases, which are, for example: (1) requirements engineering - to identify the system needs and to create a requirement specification, (2) designing the architecture - of the system by creating a design specification, (3) implementation - of functionality based on the design specification, (4) testing and verifying - the developed software to the original requirements specification, and (5) installation and maintenance - where the software is handed over to the customer (Avison & Fitzgerald, 1995).

Such a method is "waterfall" because the phases are carried out in sequences, which means that one phase is supposed to be carried out before entering the next phase. Developing software following waterfall methods was, for a long time, a standardized way of working to ensure that everything within the project was covered.

Rapid development and delivery are now often the most critical requirement for software. The changing requirements of businesses evolve, which means that SD must also evolve quickly to reflect business needs. – Plan-driven development, such as the waterfall, is useful for some types of SD projects. The types of projects where plan-driven development is appropriate are those where the needs are set from the start, and the project plan does not change during the process (Fitzgerald & Stol, 2018). Nevertheless, for other types of projects, these plan-driven approaches do not meet business needs, and in the late 1990s, ASDMs emerged (Avison & Fitzgerald, 2003).

2.4. Agile software development methods

The purpose of ASDM is to radically reduce the delivery time for software development. Instead of developing the software in sequels, the requirement specification, design, implementation, and testing are developed in a series of versions of increments. In each incremental development, the stakeholders are involved in the specifications and evaluations to provide feedback and set directions (Conboy, 2009). Developing the software as smaller sets of increments allows frequent delivery of new software versions for
evaluation (Jacobson et al., 2022). In general, ASDM advocates minimal documentation and focuses on working software.

Since the plan-driven approaches did not consider changing requirements, a new mindset was needed in the 2000s. Several knowledgeable people in IS gathered in 2001, with the common goal of getting to the bottom of this issue (Beck et al., 2001). They developed common principles needed to reach high customer satisfaction and working software with a development process that includes interacting with the customer to meet rapidly changing requirements. The group of practitioners that met summarized these principles and values into the so-called "Agile Manifesto."

The individual members of the group who founded the Agile Manifesto separately developed different types of ASDMs that today are well-established SDMs. These ASDMs rely on the Agile Manifesto's fundamental values and principles, consisting of different practices that make each SDM unique. Since these SDMs share the same values and principles (described in section 2.3.2.1), it is common and advocated that software developers could combine different ASDMs, as there is "no one-size-fits-all solution" (Conboy, 2009).

For example, one of these ASDMs is eXtreme programming (XP), founded by Kent Beck (Beck, 2000). XP consists of practices focusing on programming skills, building software efficiently, and having good quality. One of the practices is pair programming, where one developer is coding on the computer as a driver, and the other developer acts as the navigator to find errors and come up with solutions. The navigator is supposed to tell the driver what to do. The XP method focuses very much on the specific practices the developers should use while programming.

Another ASDM is scrum, founded by Ken Schwaber and Mike Beedle (Schwaber & Beedle, 2001). Scrum aims to manage SD processes. It provides practice in managing project tasks, such as timeframes for software deliveries and planning work items. Today, many organizations use these two SDMs combined in their SD processes (VersionOne, 2018, 2019, 2020).

2.4.1. Agile values and principles
The purpose of the Agile Manifesto was to collectively come up with better ways of developing software. It stated, "We are uncovering better ways of developing software by doing it and helping others do it" (Beck et al., 2001). Therefore, the group valued specific aspects and presented them in bullet points. Each bullet point consists of preferences where the first items are considered the essential aspects followed by the second items of lesser priority. The members behind the Agile Manifesto answered the question "What do you value more?" and the result became the following four values:
V1: “Individuals and interactions over processes and tools” (Beck et al., 2001) – the interactions and skilled individuals are seen as of greater importance (Fowler & Highsmith, 2001).

V2: “Working software over comprehensive documentation” (Beck et al., 2001) – the primary focus must remain on the final product (Fowler & Highsmith, 2001).

V3: “Customer collaboration over contract negotiation” (Beck et al., 2001) – it is only through ongoing collaboration that the SD team can understand the deliver what the client wants (Fowler & Highsmith, 2001).

V4: “Responding to change over following a plan” (Beck et al., 2001) – this is when the SD team can respond to external changes repeatedly throughout the SD processes the projects becomes successful (Fowler & Highsmith, 2001).

These values describe what agile is and what focus is needed for SD project success. However, these values do not provide guidance to becoming and being agile. Therefore, the members behind the Agile Manifesto agreed on twelve guiding principles that can help SD teams to adhere to the four values. The fundamental principles are designed to ensure that SD teams prioritize the most crucial things in SD. The crucial things are customer satisfaction, collaboration, and adapting to change (Beck et al., 2001).
Table 3. Agile principles

<table>
<thead>
<tr>
<th>Agile principles (Beck et al., 2001)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1: “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”</td>
</tr>
<tr>
<td>P2: “Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.”</td>
</tr>
<tr>
<td>P3: “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”</td>
</tr>
<tr>
<td>P4: “Businesspeople and developers must work together daily throughout the project.”</td>
</tr>
<tr>
<td>P5: “Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.”</td>
</tr>
<tr>
<td>P6: “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”</td>
</tr>
<tr>
<td>P7: “Working software is the primary measure of progress.”</td>
</tr>
<tr>
<td>P8: “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”</td>
</tr>
<tr>
<td>P9: “Continuous attention to technical excellence and good design enhances agility.”</td>
</tr>
<tr>
<td>P10: “Simplicity the art of maximizing the amount of work not done is essential.”</td>
</tr>
<tr>
<td>P11: “The best architectures, requirements, and designs emerge from self-organizing teams.”</td>
</tr>
<tr>
<td>P12: “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”</td>
</tr>
</tbody>
</table>

2.5. Relevance of cargo cult phenomenon for SDM

The introduction of this chapter provided an overview of the social anthropological discussion about the term CC, which consisted of various perspectives on this term. Unfortunately, the debate on CC in social anthropology provides a dim view of the phenomenon and even questions its relevance for SDM. Therefore, it is timely to introduce how the CC metaphor will be used in this dissertation to understand the phenomenon behind it.

First, when social anthropologists use the term CC, they study different tribes (their study object) and compare their study object with their view of what seems rational, such as in Worsley (1957). Usually, the study object is a group of people from another culture than the observer themselves. The observer follows the tribe and studies their behavior and rituals to understand what they are doing, why they carry out certain rituals, and their repeated actions to reach goods.

SDMs represent, as mentioned in section 2.2.1, activities, notations, and concepts from a goal and value perspective. This means that the goals an SD team tries to achieve in its SDM use and the values an SD team seeks to adhere to, represents what CC literature refers to as “goods”. The rituals carried out by a “tribe” represents the activities SDMs consists of, which are bound to notations and concepts.
From the observer’s view, these rituals carried out by the tribe are often referred to as irrational because the observer does not understand or compare with what seemed to be rational, i.e., the intended goals, the “reference point”. The SD teams’ reference point is crucial in this dissertation to uncover SDM CC's underlying purpose. The motives and reference points of the SD team must be identified to analyze the SD team’s SDM use. This thesis defines SDM CC as:

A software development method cargo cult is a temporarily delimited dysfunction having the foundation of lack of intended reasons. It leads to misconceptions and/or malpractices that are driven by the collective motive as they unconsciously fail to replicate the circumstances and success of others.

2.6. Related research – summary of paper I
To understand to what extent the reasoning behind the adoption and use of SDM that has been investigated in prior research, (Havstorm & Karlsson, 2023) surveyed existing SDM research. They scrutinized the current knowledge base on what researchers had focused on when they studied software developers and their reasoning behind SDM adoption and use.

The literature review focused on empirical studies and the type of reasoning that was addressed regarding SDM adoption and use: rational, irrational, and non-rational. This review included 111 empirical studies on SDM adoption and use, where we found that most studies considered software developers reasoning behind SDM adoption and use as rational activities. Twenty-eight primary papers addressed software developer’s reasoning behind SDM adoption and use.

Figure 4. The focus of existing research on the reasoning behind SDM adoption and use (Havstorm & Karlsson, 2023)
Among these 28 papers, most investigated rational SDM adoption and use (see Fig. 4), which implies that less attention has been given to irrational types of activities, and almost no attention has been given to non-rational activities. The paper further inductively categorized the 28 primary studies into five themes. The existing research was also sorted using the two phases i.e., SDM adoption and SDM use, and type of reasoning (see Table. 4).

Table 4. Overview of research themes and types of reasoning (Havstorm & Karlsson, 2023)

<table>
<thead>
<tr>
<th>Research theme</th>
<th>Phase</th>
<th>Rational</th>
<th>Irrational</th>
<th>Non-rational</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choice of SDM</td>
<td>Adoption</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Competing rationalities</td>
<td>Adoption</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Use</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Method deviation</td>
<td>Adoption</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Use</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Understanding operationalization</td>
<td>Adoption</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Use</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Achieved contribution</td>
<td>Adoption</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Use</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>21</td>
<td>11</td>
<td>3</td>
<td>35</td>
</tr>
</tbody>
</table>

The five themes reflect the focus of the studies, as follows.
1. **Choice of SDM.** In the papers characterized as choice of SDM, *how* the rationale of the chosen SDMs fits the situation, has been in focus. As shown in Table 4, these five studies have only addressed rational SDM adoption.
2. **Competing rationalities.** The eight papers characterized as competing rationalities, focused on how software developers deal with and prioritize conflicting rationalities during SDM adoption and use. As shown in Table 4, the papers in this theme have addressed both rational and irrational SDM adoption and use.
3. **Method deviation.** The papers characterized as method deviation, identifies, and explains deviations from either the adoption approach and the adopted SDMs, or the used SDM. As shown in Table 4, the 10 papers in this theme have addressed rational and irrational SDM use, and non-rational adoption.
4. **Understanding operationalization.** The papers characterized as understanding operationalization capture the driving actor’s understanding of how the SDM activities implement the rationale of the SDM. As shown in Table 4, the seven papers in this theme have
addressed both rational and irrational SDM adoption and rational, irrational, and non-rational use.

5. **Achieved contribution.** The papers characterized as achieved contribution, focused on to what extent the potential contribution of a chosen SDM has been reached during adoption and use.

As shown in Table 4, the five papers in this theme have addressed irrational SDM adoption and rational SDM use. More details of the papers in each theme can be found in Publication I, Havstorm and Karlsson (2023).

In Havstorm and Karlsson (2023) the results described above show that both SDM adoption and use have been investigated in four of the themes: competing rationalities, method deviation, understanding operationalization, and achieved contributions. All the research themes covered rational reasoning; however, irrational and non-rational reasoning are less covered. Based on our findings, the research community seems to know less about software developers’ irrational and non-rational reasoning behind SDM adoption and use.

![Figure 5. Identified relationships between the research themes (Havstorm & Karlsson, 2023)](image)

To synthesize the results, four theoretical relationships (Fig. 5) between research themes are presented. These relationships aim to frame future research directions for researchers. The first relationship in Fig. 5 considers how the rational, irrational, or non-rational choice of SDM impacts competing rationalities in SDM adoption and use. We see that there is an opportunity to develop an increased understanding of how rational, irrational, and non-rational SDM choices impact competing rationalities in SDM adoption and use. The second relationship in Fig. 5 considers how competing rationalities in SDM adoption and use drive rational, irrational, or non-
rational method deviation. We see that there is a need to develop deeper understanding of competing rationalities as a driver for SDM deviations, foremost irrational and non-rational method deviations. The third relationship in Fig. 5 considers how software developers understand SDM impacts method deviations. We see that there is an opportunity to research how the software developers understanding of the SDMs impacts method deviations, as software developers can avoid misconceptions, and by that reduce irrational method deviations.

The fourth relationship in Fig. 5 considers how consequences of rational, irrational, and non-rational method deviations impact achieved contributions. We see those competing rationalities and understanding operationalization leads to lack of the achieved contributions. Therefore, we suggest future research to develop an increased understanding of how method deviations impact achieved contributions.

We concluded in the paper that there is a need for more research on software developers’ reasoning behind the adoption and use of SDMs when exploring irrational and non-rational SDM adoption and use activities. To reach increased understanding of each identified future directions described above, we posed four broad research questions in Table 5 as suggestions. More details can be found in Publication I, Havstorm and Karlsson (2023).

Table 5. Future research directions and research questions (Havstorm & Karlsson, 2023)

<table>
<thead>
<tr>
<th>No</th>
<th>Research direction</th>
<th>Research question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Consider how the rational, irrational, or non-rational choice of SDM impacts competing rationalities in SDM adoption and use</td>
<td>To what extent do rational, irrational, or non-rational SDM choices lead to competing rationalities in organizations?</td>
</tr>
<tr>
<td>2</td>
<td>Consider how competing rationalities in SDM adoption and use drive rational, irrational, or non-rational method deviation</td>
<td>How do software developers deal with competing rationalities when doing rational, irrational, and non-rational tailoring of SDMs?</td>
</tr>
<tr>
<td>3</td>
<td>Consider how software developers understanding of SDM impacts method deviations</td>
<td>How does software developers’ understanding of SDM contribute to rational, irrational, and non-rational method deviations during adoption and use?</td>
</tr>
<tr>
<td>4</td>
<td>Consider how consequences of rational, irrational, and non-rational method deviations impact achieved contributions</td>
<td>How do software developers’ rational, irrational, and non-rational method deviations impact achieved contributions of SDM adoption and use?</td>
</tr>
</tbody>
</table>
Chapter 3. Theoretical Foundations

This dissertation focuses on SDMs and combines one phenomenon and two main theories. The cargo cult phenomenon (CC) and memetic behavior was previously presented in Chapter 2 together with SDMs. The latter describes ways to structure the SD process and various work practices to support organizations in reaching a common understanding of how to carry out the work. In this chapter, the social action theory (SAT) (Weber, 1947) and the work motivation theory (WMT) (Gagné & Deci, 2005) are discussed.

The motives behind using SAT and WMT are the following. First, SAT focuses on the underlying rationality that motivates people’s actions. Using SAT, it is possible to identify what the SD teams were doing (their social actions), the reasoning behind the actions, and the type of rationale, if any. Second, WMT focuses on the underlying motivation behind the actions, i.e., what motivates people to act in a certain way. Combining SAT with WMT makes it possible to identify why the SD teams acted as they did.

WMT focuses on the collective perspective on motivation. The theory itself stems from the self-determination theory (SDT) (Ryan & Deci, 2000), which focuses on individuals’ motivation. Combining the two (SAT and WMT) theories will help to understand how, why, and in which context the CC exists. Therefore, this chapter will describe the two theories used in this dissertation, starting with SAT followed by WMT.

3.1. Social action theory

The CC phenomenon in the previous section describes situations when people seek to gain goods and perform rituals to reach their desired success, as explained in Worsley (1957). In order to understand the rituals, i.e., recurrent actions, the following section focus on SAT by Weber (1947). In his study, Worsley (1957) discussed the possibility of using SAT (Weber, 1947) but found that irrational behavior was not covered. However, this dissertation focuses on the concept of rationality and the typology of different social actions based on Kalberg’s (1980) interpretations of Weber (1947), which includes irrational behavior. Thus, using Kalberg (1980), SAT consists of three types of social actions: rational, irrational, and non-rational, as illustrated in Table 6.
Table 6. Rational, irrational, and non-rational social actions

<table>
<thead>
<tr>
<th>Type of Social Action</th>
<th>Rational</th>
<th>Irrational</th>
<th>Non-Rational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Means-end</td>
<td>An SD team succeeds in its reasoning behind its actions in relation to the intended goals</td>
<td>SD teams fail in its reasoning behind its actions in relation to the intended goals</td>
<td></td>
</tr>
<tr>
<td>Value-based</td>
<td>An SD team succeeds in its reasoning behind its actions in relation to the intended values</td>
<td>SD teams fail in its reasoning behind its actions in relation to the intended values</td>
<td></td>
</tr>
<tr>
<td>Traditional</td>
<td>There is no reasoning behind the action and the SD team sticks to old habits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affectual</td>
<td>There is no reasoning behind the action and the SD team acts upon emotions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.1.1. Rational social actions

Rational social actions are founded in reasoning, which means that someone (in this dissertation, an SD team) acts towards specific goals or values they intend to achieve. Consequently, rational actions are based on underlying reasoning. Actions based on underlying reasoning represent SD team members' awareness of why a certain action is carried out (Weber, 1947).

In Table 6, two social actions can be the foundation for rational actions: means-end and values-based. The rational means-end social actions, represents actions where a collective, an SD team, chooses and executes a relevant action to achieve an SDM end. The SD team persists with an understanding of the underlying reasoning behind its choice of actions, and the SD team succeeds with its execution, i.e., achieves the desired end. In this way, the SD team carries out different actions when using SDM practices, to achieve the goals with its SDM use.

The second rational value-based social action, see Table 6, represents actions performed by an SD team, understanding the underlying reasonings behind its actions. The software developers’ actions are derived from certain values. Rational values-based actions are successful when the intended values are adhered to, which means that the software developers act in accordance with a set of values. Thus, the SD team does not focus on the techniques needed to carry out certain tasks, but rather on the values these actions incorporate. Even though the different techniques and goals are not in focus, these are still known (Weber, 1978).
3.1.2. Irrational social actions
As described above, social actions performed by an SD team where the intended reasoning succeeds are called rational social actions. On the other hand, situations, where the SD team fails in its reasoning of either fulfilling the goals (means-end) or aligning with the values (value-based), are irrational social actions. Although Weber (1978) explains that actions are viewed from an idealistic goal-oriented perspective, it is essential to understand that irrational actions are not less meaningful than rational actions. Irrational social actions are founded in reasoning, where the software developers’ judgement is based on misconception. As shown in Table 6, two social actions form the basis for irrational actions: means-end and value-based.

The irrational means-end social actions, see Table 6, represents actions performed by a collective, an SD team, to achieve an end. The SD team persists with a partial understanding of the SDM, founded in some type of reasoning. However, the SD team fails with its reasoning with the intended goals i.e., does not achieve the desired end. It means the SD team carries out actions that are not suitable for the claimed SDM goal, based on misconceptions when using SDM practices. The actions carried out based on misconceptions result in malpractices and irrational action.

The irrational value-based social action, see Table 6, represents actions performed by an SD team, which are based on misinterpretation of the SDM values. The software developers’ actions are, as the rational value-based social actions, derived from SDM values. Thus, the SD team does not focus on the techniques needed to carry out certain tasks, but rather on the values these actions could provide or achieve. However, the SD team fails to understand how these values are supposed to steer their actions and as a result the SD team does not act in accordance with the set of values.

Interestingly, the underlying understanding of the assertive motive or purpose of why something happened is forgotten. In addition to why something has happened, it is crucial to understand what has happened; once the reason for why and the understanding of what has happened, it is possible to understand if an SD team acted rationally or irrationally.

3.1.3. Non-rational social actions
To continue the previous explanation of different types of social actions, we will look at the final and third types of social actions: the non-rational actions. In contrast to rational and irrational social actions, non-rational social actions are not founded in reasoning. Non-rational social actions could be actions based on traditions, which means that the SD team sticks to its old routines or habits, and acts upon affect/emotions. As shown in Table 6, traditional and affectual social actions form the basis for non-rational social actions.
Traditional social actions could, for example, be an SD team which previously followed a specific SDM practice and has now adopted a new SDM practice. A non-rational social action will be if the SD team, without any conscious decision, continues to follow its old routines and the old way of working instead of adhering to its new SDM practice. It means the organization, or the specific SD team, has created a specific pattern and a particular way of working for a very long time, and changing something one is used to is difficult. In this case, traditions are rooted in the organization and how the SD team carries out its daily work. The traditions the SD team acts upon are old routines and habits rather than reasoning behind how it should act. Therefore, these are non-rational traditional social actions; see Table 6.

For actions to become routine, they need to be carried out repeatedly several times for a more extended period, where these goals and values have been placed in the background, no longer given any attention. These routines might from the beginning have included explicit goals and/or values in a written description of how things should be carried out. Eggers and Kaplan (2013) describe routines and capabilities as based on a precise understanding of how things should be done. Routines, as a sort of cognitive process, are experiences that are encoded into routines.

One example of a routine could be a meeting, which consists of a specific expectation of how this meeting should be carried out; it is based on how things have been repeatedly done before. These meetings consist of plans and rules supporting how the meetings should be carried out. In Feldman (2000), routines are defined as "repeated patterns of behavior that are bound by rules and customs". Feldman and Pentland (2003) further viewed routines as coordination mechanisms as they were coordinated by different tasks following a template to complete them—the coordination in enacted by bringing people together and creating a common perspective across groups.

The final type of social action in Table 6 is affectual action. This type of action captures when SD teams act upon emotions, so there is no reasoning behind their action. Instead, they go for the “gut feeling”, what their body tells them to do now. Again, one could say that this could be related to cognitive aspects of human beings, that humans follow a feeling that they do not think about at all (e.g., do something that the body just tells them to do).

When we look closer into how software developers might react and act based on emotions, we can identify how they could follow a simple feeling from their gut, in other words, a “hype”. For example, there could be times when different departments in an organization compete; as when one department becomes successful in one way of working and takes the lead. Competitions might create the other departments in a “competition mode”. So, this mode could lead SD teams to embrace a hype based on seeing
something happen somewhere else, and they want to do the same without thinking or reasoning at all. Trend following could also be very much related to this, and when people jump on the bandwagon, following the others without thinking why and how they were supposed to do those things.

Regardless of being traditional or affectual actions, they have one thing in common; there is no reasoning behind these actions. Therefore, a collective, e.g., an SD team, which acts in a certain way, either because it is based on old habits previously founded in routines, or following emotions, acts non-rationally.

### 3.2. Work motivation theory

When software developers carry out different actions, their actions are somehow related to motivations. This section will describe work motivation theory (WMT) and the related self-determination theory (SDT) and its distinction between the three main types of motivations, illustrated in Fig. 6.

First, the type of motivation could depend on feeling of control; this is when a software developer feels demanded to do things. For instance, the software developer is to present at a business meeting since its work description demands it. Their work description includes all the tasks this software developer should carry out daily. Therefore, the motivation behind this is control from the outside, in other words, extrinsic motivation. Extrinsic motivation could also be related to goal fulfillment. (Ryan & Deci, 2000).

Jansson (2015) studied how a particular work task could achieve higher productivity with motivation among software developers who use a whiteboard during daily scrum meetings. When IT organizations are about to adopt, for instance, agile methods, they refer to discussions about putting more effort into getting higher employee motivation.
It is most likely that most employees in organizations act accordingly to extrinsic motivations since extrinsic motivation is more related to goal fulfillment. Although what is the difference between goal fulfillment and control from the outside, external control? The second type of motivation is when a software developer does not feel external control. Instead, they feel self-determination, in other words, intrinsic motivation.

The third type of motivation is when a software developer’s action does not depend on external control or self-determination. Instead, the software developer is unmotivated, i.e., amotivated (Ryan & Deci, 2000).

At the core of self-determination theory, Ryan and Deci (2000) defined motivation as "to be motivated means to be moved to do something". Therefore, being motivated implies two different perspectives (extrinsic and intrinsic), which represents levels of self-determination, i.e., how much self-determination a software developer has.

There is also the orientation of that motivation, i.e., the type of motivation. It is worth noting that it is possible to categorize a software developer's action as unmotivated; these are the situations when a software developer does not feel motivated or inspired to act (Ryan & Deci, 2000). These are pointed out in Fig. 6 as amotivation.

3.2.1. Extrinsic motivation
Software developers who feel demanded to do things due to external control act based on extrinsic motivation. “Extrinsic motivation is a construct that
pertains whenever an activity is done in order to attain some separable outcome” (Ryan and Deci (2000)).

Extrinsic motivation is the opposite of intrinsic motivation, consisting of instrumental value activity. Both extrinsic and intrinsic motivation can be viewed as different levels or degrees of autonomy. The autonomy level of extrinsic motivation is either that a software developer has some feeling of choice compared to when the software developer is more compliant with external control.

Perceived external control results in a behavior where a software developer is extrinsically motivated. It means that the software developer executes a particular activity for some consequence separate from the activity, and this is called deviating. Extrinsic factors are, for example, tangible rewards, deadlines, surveillance, and evaluations. These extrinsic factors are controlled motivation (Gagné & Deci, 2005).

Internalized regulations could have three characters: introjected, identified, or integrated. Thus, these actions are not considered to have an intrinsic motive since SDM CC situations are related to work practices. SDM CC situations mean that software developers’ actions are part of their work and that the organization will expect their use of SDM practices. Thus, it is not their own choice entirely. Instead, the focus is on autonomy and volition to pursue a specific action.

The self-determination continuum (Ryan & Deci, 2000) contains three types of internalized regulated motivations to describe the level of autonomy. First, introjected regulations are situations when the regulation controls the software developers, which is a controlled form of internalized extrinsic motivation (Gagné & Deci, 2005). Second, identified regulations are when software developers feel greater freedom and volition as the behavior is more congruent with their personal goals and identities. In this case, software developers identify with the value of behavior for their own self-selected goals (Gagné & Deci, 2005). Software developers’ actions are defined as being of the character identified regulations, which means they are autonomously extrinsically motivated (Gagné & Deci, 2005). Third, integrated regulations are when software developers are motivated by their view that the activity is instrumentally vital for personal goals. Thus, they are not interested in the activity itself.

### 3.2.2. Intrinsic motivation

Doing an activity for inherent satisfaction rather than for some separable consequence is spontaneous behavior (Ryan & Deci, 2000). For human beings, being intrinsically motivated is very important. The intrinsic motivation in Fig. 6 captures an interested and enjoyable relationship between the software developer and their activities (Ryan & Deci, 2000). There is a difference between being intrinsically motivated and intrinsically interesting
tasks. Intrinsically related to such actions are feelings of competence, autonomy, challenge, positive feedback, and enjoyment (Gagné & Deci, 2005). These intrinsic factors are called autonomous motivations. “For a high level of intrinsic motivation people most experienced satisfaction of the needs both for competence and autonomy” (Ryan and Deci (2000)

Studies have shown that when a person gets a choice and opportunity for self-direction, it enhances intrinsic motivation since they receive greater autonomy. However, intrinsic motivation only occurs for activities with intrinsic interest for an individual, according to Ryan and Deci (2000). These activities must have either: novelty, challenge, or aesthetic (i.e., be interesting or enjoyable) value for the specific software developer.

3.2.3. Amotivation
When a software developer does not find a situation or activity motivational, the software developer will act without intention. Such situations are predicted by the work context and the software developer's impersonal orientation (Gagné & Deci, 2005).

The state of lacking an intention to act. When amotivated, a person’s behavior lacks intentionality and a sense of personal causation. Amotivation results from not valuing an activity […] not feeling competent to do it […] or not believing it will yield a desired outcome. (Ryan and Deci (2000)

In contrast to extrinsic (external and internalized regulated) and intrinsic (autonomous) motivation, amotivation refers to situations where both lack of intentions and motivations are behind a software developer’s pursued activity. This lack of intentions means that the software developer feels that neither external control, such as goal achievement, nor autonomous self-determination, such as a sense of freedom, are motivating factors for the activity.
PART III – EMPIRICAL SETTINGS AND RESEARCH APPROACH
Chapter 4. Empirical Settings

This chapter outlines the empirical settings of this study, i.e., where the study has taken place and the organization’s characteristics. Thus, this description is an essential part of clarifying the transferability of the result. The chapter starts by presenting the criteria for case selection, followed by a description of the overall organization. Thereafter, a description of the division studied, i.e., more detailed information to provide a context of where the empirical material was collected, follows.

4.1. Criteria for case selection

To collect a case within the scope of the study area, ASDM use, the following criteria were used in the case selection:

1. There was a need to study an organization doing SD using ASDM, to align with the purpose of this thesis i.e., to identify and understand the challenges in ASDM use.
2. The organization had to use SDM practices as the CC phenomenon is related to activities performed as part of an SDM practice and to be in line with its justification.
3. The software developers in the SD organization had to work in teams, as the CC phenomenon refers to collective behavior. It would thus not be possible (or relevant) to capture such a phenomenon when studying individuals developing software. The following sections describe the overall organization and the specific SD teams studied.

4.2. Alpha – An industrial manufacturing company

This study researched three SD teams in Sweden at a large international industrial manufacturing company between 2018-2020. The company was founded in Sweden, and has sites in six different countries worldwide and over 16,000 employees. The company will be given the acronym “Alpha” to preserve its confidentiality. Today, Alpha has customers in more than 150 countries. Alpha has a leading success in productivity and advanced technology in its business area. In Sweden alone, the company has a total of 5,000 employees. Software is a well-integrated part of Alpha's products; software being developed in the digital solution unit hereafter referred to as "Beta" (see Fig. 7). Beta has approximately 1,100 employees worldwide, and 170 employees in Sweden.
Today, Beta consists of several divisions (see Fig. 7) and develops different system modules for its products in different countries. In this thesis, the focus is on one of the divisions, from now on referred to as “Delta” and its Swedish sub-division. As shown in Fig. 7, Delta is divided into two sub-divisions that include SD teams. Although the SD teams belong to different sub-divisions, the researched sub-division in Sweden had close cooperation with another sub-division in the US, mainly working on testing, hence the acronym "Delta Sweden" and "Delta US". Fig. 7 illustrates the overall organizational structure to give an overview of the case studied.

In terms of software development, Beta has moved from a traditional plan-driven SDM to ASDMs. This transitioning means all of Beta’s divisions working with SD use ASDMs. Furthermore, the change of SDM means that all of Beta’s divisions have previously undergone an organizational transformation and an adoption process. The following section will briefly describe the SDM transformation of the sub-division Delta, and further focus primarily on the use of ASDMs by the three SD teams at Delta Sweden.

4.3. Delta – The division studied
Before this study took place, Delta went through an ASDM adoption. The previous way of working followed traditional plan-driven SDMs, consisting of fixed-release plans and top management decision-making. Due to an overarching decision to move Beta towards ASDM, Delta Sweden was one of the frontiers in Beta to adopt the agile way of working. This change is highly relevant to understand and have as background knowledge. The
transformation made, and its effects, help one to understand the reasons behind the company’s current use of SDMs.

Delta’s transformation aimed to: (1) increase the quality of its products, (2) visualize the work and progression, (3) increase interaction with team members, (4) get better stakeholder communication, (5) get better possibilities to prioritize and focus on the “right” things, and (6) focus on increasing productivity and eliminating impediments. As stated in Delta’s development method process description: “Department […] will introduce Scrum/Kanban to visualize the work development teams are doing. It is obvious that a certain standardization is required and the main requirements from Line management is given in this document.”

After the transformation, Delta consists of the roles summarized in Table 5. The division has a management team, and several SD teams. The management team consists of two managers (one in each office), three product owners (PO) in the US, one customer in the US, and four scrum masters (SM). Each SD team is led by an SM, and the SM acts as the link between the management team and the SD team. This is shown by placing the SM between the management team and the SD team in Table 7. Delta Sweden had three SD teams, and Delta US had one SD team. The SD teams had technical architects, test responsible, developers, and integration testers.

Table 7. Role descriptions in Delta

<table>
<thead>
<tr>
<th>Team level</th>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>Manager</td>
<td>Manages the unit and facilitates management meetings.</td>
</tr>
<tr>
<td></td>
<td>Product owner</td>
<td>Facilitates iteration planning and prioritization of backlogs.</td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>Provides business needs and decides focus. Gives feedback on demonstrations.</td>
</tr>
<tr>
<td></td>
<td>Scrum master</td>
<td>Facilitates daily scrum meetings, and demonstration.</td>
</tr>
<tr>
<td>SD team</td>
<td>Technical architect</td>
<td>Facilitates architectural design issues.</td>
</tr>
<tr>
<td></td>
<td>Test responsible</td>
<td>Runs tests at team level: unit tests and regression tests.</td>
</tr>
<tr>
<td></td>
<td>Developer</td>
<td>Executes research and development of software.</td>
</tr>
<tr>
<td></td>
<td>Integration tester</td>
<td>Responsible for system integration tests on the industrial machines.</td>
</tr>
</tbody>
</table>
Fig. 8 illustrates the relations between the SD teams and the different roles in Delta. This research focuses on the three SD teams at Delta Sweden. Delta Sweden’s SD teams are shown in Fig. 8: team A, team B and team C. Moreover, a fourth, team D, evolved during the final year of this study, in 2020, as an extra resource. However, this team is not included in the study. As said above, both sub-divisions in Delta had a tight collaboration. The team in the US was not explicitly studied; however, it is an essential part of the study context. Thus, Delta US is mentioned for the sake of completeness.

4.3.1. Delta Sweden’s ASDM practices
The following text describes Delta Sweden’s 25 ASDM practices. Delta Sweden used the ASDM practices from *scrum*, the most used ASDM worldwide (Jacobson et al., 2022), *eXtreme programming* (XP), *kanban*, and *DevOps*. Table 8 summarizes all of Delta’s different ASDM practices and presents: practice number, the ASDM practice, which ASDM each practice belongs to, and which team used which ASDM practice.

The first six ASDM practices presented include a more detailed description in the following text. The reason for this is because those practices are “meeting practices” and more details relating to them were possible to capture during this study. This allows for a rich contextual description of Delta regarding these practices.
Table 8. Delta’s ASDM practices

<table>
<thead>
<tr>
<th>Practice no.</th>
<th>ASDM Practices</th>
<th>Scrum</th>
<th>XP</th>
<th>Kanban</th>
<th>DevOps</th>
<th>Management team</th>
<th>Beta Scrum Masters</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Scrum meetings</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>Scrum of scrums</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>Daily scrum meetings</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P4</td>
<td>Sprint retrospective meetings</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P5</td>
<td>Sprint planning meetings</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P6</td>
<td>Sprint demo</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P7</td>
<td>Sit together</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P8</td>
<td>Incremental design</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P9</td>
<td>Incremental deployment</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P10</td>
<td>Code and test</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P11</td>
<td>Sprint</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P12</td>
<td>Shared code</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P13</td>
<td>Continuous delivery</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P14</td>
<td>Continuous flow</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P15</td>
<td>Rapid changes</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P16</td>
<td>Flow</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P17</td>
<td>Continuous integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P18</td>
<td>Use of visualization</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>P19</td>
<td>Product backlog</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P20</td>
<td>Project backlog</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P21</td>
<td>Sprint backlog</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P22</td>
<td>Estimation</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P23</td>
<td>Forecasting</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>P24</td>
<td>Burndown chart</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>P25</td>
<td>Burnup chart</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
This study observed six primary formal types of meetings (see Table 9.), at management, SD team, and cross-division level. As mentioned, Delta Sweden and Delta US had close collaboration, especially the management team (including managers, product owners, customers, and scrum masters), who had digital (P1) scrum meetings (Fig. 9) three times per week (Mondays, Wednesdays, and Fridays). The purpose of the scrum meetings was to provide status reports as well as to keep track of the ongoing process to set directions and prioritizations for the projects.

Delta Sweden’s SMs were, from time to time, involved in several projects with other SD teams from other divisions in Beta. Some of these SD projects were cross-division, which required the different SMs from each project to gather in a “stand up” (P2) scrum of scrum meetings (Fig. 10). The scrum of scrum meetings focus on reporting the ongoing progress. The purpose of the meetings was to ensure that all developed systems were compatible and followed the same standard. The reason was that over a period of time, all divisions in Beta underwent a major digital transformation by introducing the Jira method tool and the Git and Github version management tools, referred to as the “digital workflow project”. The tools were introduced as a pilot project for a few SD teams and sub-divisions, to ensure that they worked. In 2020, the digital workflow was rolled out to the entire Beta.
### Table 9. Delta’s meeting overview

#### Observed Formal Meetings

<table>
<thead>
<tr>
<th>Type of meeting</th>
<th>Participants</th>
<th>Duration</th>
<th>Time period</th>
<th>Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum Meetings</td>
<td>Management: unit managers, product owners, customer, scrum masters</td>
<td>2h</td>
<td>2018-2020</td>
<td>Three times per week (Monday, Wednesday, and Friday)</td>
</tr>
<tr>
<td>Scrum of Scrums</td>
<td>Scrum masters from different divisions in Beta</td>
<td>30min</td>
<td>2018-2019</td>
<td>Sporadically, based on the need</td>
</tr>
<tr>
<td>Daily Scrum Meetings</td>
<td>SD teams including scrum master and sometimes the manager</td>
<td>15-30min</td>
<td>2018-2020</td>
<td>Daily</td>
</tr>
<tr>
<td>Sprint demo</td>
<td>Management: unit managers, product owners, customer, scrum masters SD teams</td>
<td>4h</td>
<td>2018-2020</td>
<td>Irregular, approximately every 4-6 weeks</td>
</tr>
<tr>
<td>Sprint retrospective</td>
<td>SD teams and the scrum masters</td>
<td>2h</td>
<td>2018-2020</td>
<td>Team A – monthly Team B and C – never</td>
</tr>
<tr>
<td>Sprint planning meetings</td>
<td>SD teams and the scrum masters</td>
<td>1h</td>
<td>2018-2020</td>
<td>Team A and B – never Team C – three times</td>
</tr>
</tbody>
</table>

#### Observed Informal activities

<table>
<thead>
<tr>
<th>Informal activities</th>
<th>Participants</th>
<th>Duration</th>
<th>Time period</th>
<th>Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coffee breaks</td>
<td>SD teams and scrum masters</td>
<td>30min</td>
<td>2018-2020</td>
<td>Daily</td>
</tr>
<tr>
<td>Board games</td>
<td>SD team members</td>
<td>30min</td>
<td>2018-2019</td>
<td>Daily (I participated sporadically)</td>
</tr>
<tr>
<td>Lunch breaks</td>
<td>SD teams and scrum masters</td>
<td>30min</td>
<td>2018-2020</td>
<td>Daily</td>
</tr>
<tr>
<td>Lunch walks</td>
<td>SD team members</td>
<td>30min</td>
<td>2018-2020</td>
<td>Daily (I participated 2-3 times per week)</td>
</tr>
</tbody>
</table>

#### Interviews – carried out by researcher

<table>
<thead>
<tr>
<th>Type of interviews</th>
<th>Participants</th>
<th>Duration</th>
<th>Time period</th>
<th>Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstructured interviews</td>
<td>SD team members (architect, testers, developers) scrum masters manager</td>
<td>5min-30min</td>
<td>2018-2019</td>
<td>Sporadically</td>
</tr>
<tr>
<td>Semi-structured interviews</td>
<td>SD team members</td>
<td>1-2h</td>
<td>2020</td>
<td>9 interviews</td>
</tr>
</tbody>
</table>
It was possible to observe each SD team having (P3) daily scrum meetings (DSM), carried out in three different forms. The coming section (4.3.2) provides details on each team. DSM is a "15-minute time-boxed event" (Stray et al., 2016) focusing on the SD team members answering three questions daily to provide a status report. Each SD team also had (P4) sprint demos (SPD) to demonstrate their developed solutions in each sprint. For example, Delta used the ASDM scrum, both DSM, and SPD, and organized the SD process in sprints, i.e., developing software iteratively, resulting in smaller increments, demonstrated in SPDs, seen as core activities.

Delta Sweden’s three SD teams used different ASDM practices in combination with DSM and SPD, making each team unique. For example, (P5) sprint planning meetings (SPM) were carried out in one of the teams, while another team had (P6) sprint retrospective meetings (SRM). A third team did not carry out either of these two ASDM practices. From the original SDM description of scrum, both DSM and SPD are core practices alongside SPM and SPR. Nevertheless, all these ASDM practices were, according to Delta’s SDM description, expected to be used by each SD team.

In addition to the abovementioned ASDM practices, Delta Sweden further used ASDM practices from XP, such as: sit together, incremental design, incremental deployment, test and code, and shared code. For example, the SD team members (P7) sat together in their team rooms, and the software (both the design and code) was developed through (P8) incremental design and (P9) deployment. The solutions were aimed to be (P10) coded and tested within each (P11) sprint, and the code was (P12) shared among the team members. Delta further used ASDM practices from kanban such as: (P13) continuous delivery, (P14) continuous flow, and (P15) rapid changes. Delta’s main reason for combining scrum and kanban was to get a process with a constant (P16) flow.

Delta Sweden used the ASDM DevOps for (P17) continuous integration (CI) to complement scrum, XP, and kanban with ASDM practices: continuous delivery, working software, automation tests. Beta introduced, at the end of 2018, the integration and version control tool Github for the entire SD unit. The aim was to help reach continuous delivery, working software to the customer, and testing the solutions to achieve this end (flow).

An overarching ASDM practice was the (P18) visualization of artifacts retrieved from both the ASDMs scrum and kanban. Delta (both Sweden and US) had an SDM visualization tool. The visualization tool before 2017 to late 2019 was called Fogbugz, which later, at the end of 2019, was replaced
by Jira and the version control system Git. Delta used the tools to document (P19) product backlogs, overarching (P20) project backlogs, and SD teams' (P21) sprint backlogs in Jira. The versions of the software solutions were documented and controlled in Git. The tool also included a scrum board at both project and team levels. Both Fogbugz and Jira provide possibilities for Delta to keep track of (P22) estimation, i.e., estimate the project and the SD team's sprints, and (P23) forecasting with (P24) burndown, i.e., receive information on the work completed in the sprints and the remaining work to predict the SD teams' probability of achieving their work in time, and (P25) burnup charts, i.e., a visualization of completed work in a sprint compared with its scope. The chart helps to identify problems such as scope creep or a deviation from the project plan.

**Delta Sweden’s software testing environment**

To ensure the high quality of Delta’s digital solutions, Delta Sweden tested solutions in a specific test lab, besides the tests in each software developer’s development environment. Compared to Delta US, which ran tests on industrial machines, Delta Sweden wrote regression tests to be used in the simulators. The test lab consisted of automated tests in the simulators to allow running tests for several hours straight, executed at all times. Most of the industrial machines which Delta US tested, together with the developed software, are situated in the US. Few of the machines were found in Sweden, which meant the team members sometimes had to visit the US.

**4.3.2. The software development teams at Delta Sweden**

**Team A**

Ten team members focused on Delta’s old version of their software, to maintain its capabilities for sustainability to ensure the worldwide customers can continue use the machines in industry. The team consisted of a mixture of pure software developers, 70% of whom were consultants, and 30% were employees of Delta with an engineering background. The team included software testers who mainly focused on regression testing on their computers, as well as testing the solutions in Beta’s testing lab with their simulation machines.

Every SD team had its own SM who worked both as a software developer and as a team manager. Team A’s SM also acted as project leader of another unit’s project at Beta.
The 10 software developers were divided into two rooms, as illustrated in Fig. 11. The room illustrated on the left was also team A’s “meeting room” where all team members gathered for their DSMs. Since the number of team members exceeded the chairs available in the room, the team pursued these meetings as daily “stand-up” meetings. The room on the lefthand side of Fig 11. illustrates one of the DSMs with the empty circles showing the participated team members, and the filled circle showing myself, the researcher.

**Team B**

Five team members focused on Delta’s future products and their digital solution i.e., to create an upgraded version for its industrial machines. A future digital solution requires the SD team to do both research and development to be at the forefront with the solutions. The team consisted of a mixture of pure software developers, 28% of whom were consultants and 72% were employees of Delta, with an engineering background; some as former SD team members experienced in scrum teams, and as software testers.
Team B borrowed team members from team A when extra recourses were needed in development work. The reason for borrowing staff was team A had a complete understanding of the older software, which was needed for future projects. The team members of team B were divided into two rooms. The four software developers worked in the room illustrated in Fig. 12. and their SM in a separate room next door. Team B’s DSMs took place as a “sitting” meeting, where all software developers and the SM sat down.

Team C

Eight team members focused on Delta’s innovative research and development (R&D). It meant mainly working on autonomous software solutions, for the industrial machines. The team consisted of software developers and testers, of whom 28% were consultants and 71% were employees of Delta.

The team members of team C were divided into two rooms. Three software developers and the team’s SM worked in the room illustrated to the left in Fig. 13. and the four other software developers in a separate room, illustrated to the right. The lefthand-side illustration shows where team C’s DSMs took place, as a “standup” meeting, using their visual scrum board on the screen.
### 4.4. Summary of Delta between 2018-2020

A longitudinal ethnographic study focused on Delta between 2018 and 2020 (please find the details in Chapter 5, section 5.3). Table 10 provides a timeline of the overarching activities observed at Delta, structured using years and quarters.

Table 10. Timeline

<table>
<thead>
<tr>
<th>Year</th>
<th>Quarter</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 2018 | Q2      | - Initial meeting with manager in Delta Sweden.  
- Participated in two days of agile training – cross-sectional among several Beta units. Training was led by a consultant as agile coach. |
|      | Q3      | - Start of the longitudinal study and the observations.  
- Introduced myself to all the SD teams during their daily scrum meeting.  
- Introduced myself to the management team in their scrum meeting.  
- Met with “digital workflow” project managers, project aimed to replace the visual tool Fogbugz and introduce a digital visual tool Jira with Git and Github. |
|      | Q4      | - No specific activities than regular SD procedures were observed |
| 2019 | Q1      | - Digital workflow: the visual tool Jira was planned to be tested by two teams at Beta, i.e., a pilot project.  
- Digital workflow: scrum of scrum hosted with representatives (all scrum masters) from all units from Beta.  
- Team B received a new product owner. |
|      | Q2      | - Delta US started to use the visual tool Jira. |
|      | Q3      | - Delta Sweden implemented the visual tool Jira. |
|      | Q4      | - Team B rolled out the new digital solution to the market. |
| 2020 | Q1      | - A fourth SD team were set up, a collaboration between Delta Sweden and another Beta unit. Team members from team C joined the fourth SD team.  
- Team A’s scrum master was appointed as a project manager for an additional project at Beta. |
|      | Q2      | - Beta went full remote due to Covid-19, the meetings (both SD teams and management) were kept digital using digital meeting solutions. If anyone needed to enter the facilities, they had to contact their manager to get confirmation to enter the building. |
|      | Q3      | - The SD teams still worked remotely, unless the solutions needed to be tested in the lab. |
|      | Q4      | - The future solution by SD team B was up and running. |
Chapter 5. Research Approach

“Theory is a statement of concepts and their interrelationships that shows how and/or why a phenomenon occurs” - (cf. Gioia & Pitre 1990).

In this chapter, I start by providing the philosophical stance taken in this dissertation, followed by the research strategy. This is followed by the qualitative, ethnographic study perspective including the research setting, my role, the empirical material, and how the empirical material was collected. Thereafter, an outline of the theorization of the SDM CC phenomenon and the theory development is presented. After the outline of theorization, the analytical procedures, how the SDM CC framework was used, is presented. Further on is the procedure of the conducted literature review is presented. At the end of this chapter, I outline the credibility and transferability of research results as well as the ethical considerations.

5.1. Philosophical considerations

The nature of this dissertation has the characteristic of understanding a phenomenon, which according to Braa and Vidgen (1999) is described as existing in the interpretative paradigm, illustrated in Fig. 14. Interpretative research can be viewed as a contrast to the two other paradigms: interventions i.e., where the research aims to suggest and create a change, and reduction i.e., where the research aims to result in predictions. This dissertation focuses on understanding and explaining SDM CC, i.e., focusing on a

![IS research framework (Braa & Vidgen, 1999)](image-url)
phenomenon of collective behavior, where SD teams’ use of ASDMs are viewed as social actions. This study builds on the qualitative research traditions in information systems (Walsham, 2006), and particularly the ethnographic research approach (Myers, 1999).

5.2. Abductive ethnography of social phenomena

I started this research project by having a research strategy focused on the social phenomenon of CC. To fit the puzzle together, the theoretical understandings were altered based on empirical findings. By doing so, the theoretical knowledge and empirical findings gradually evolved.

According to Bajc (2012), abduction can be used to identify a social activity that has been theoretically interesting. The data analysis in abductive ethnographic approaches focus on letting the collected material be compared with theory. This dissertation consists of an abductive ethnographic research strategy, which consists of theoretical knowledge being constant compared with empirical material throughout several phases so as to develop a theory.

Below in Fig. 15, the research strategy and approaches used in this dissertation are visualized in relation to each publication. As shown in the figure, the research strategies in this dissertation include both literature review and abductive ethnography. The literature review is presented in Publication I. The theorizing is represented in four publications (papers II-V). Each publication contributes to the theory since the research process is an iterative process between both the theoretical and empirical aspects. The initial theorizing and the first version of the theory set the starting point for the data collection, data analysis, and the iterations of theorizing.

![Figure 15. Research strategy and publications](image-url)
The data analysis contributes to further development of the theory. Publications I and II both use SAT based on the need to understand the perspective on the non-rational and irrational results from empirical studies on SDM adoption and use. Paper III in this figure is connected to both the theorizing, i.e., the SDM CC framework, and to data collection since this paper provides the first test of using the SDM CC framework to analyze empirical material. Papers IV and V include both data collection and further tests on refined versions of the SDM CC framework by analyzing a new set of empirical material. Both papers IV and V are connected to theorizing.

Ethnographic research is one of the in-depth interpretative approaches in qualitative research, used when researchers study people and culture (Czarniawska, 2007; Myers, 1999; Van Maanen, 2011). The CC phenomenon, when studied in the field of social anthropology, was identified using an ethnographic approach by anthropologists (Worsley, 1957). To be able to capture the SDM CC phenomenon, extensive time in the field was needed so as to understand the context in the bigger picture. This provided the opportunities to follow the organization, its ongoing processes, and the action taken by the SD teams.

Thus, the ethnographic approach gives opportunities to study, reflect, and understand a phenomenon as the researcher spends an extensive amount of time on site, taking part in the natural setting of the subjects, observing what the people are doing and what they say. Since the researcher takes part in the “life” of the people at the research site, this part is called participant observation. As the researcher is supposed to not disturb the subjects, the research instrument is the researcher themself. I have collected different kinds of data, using interviews and observations (my field notes), and collecting an extensive number of organizational documents.

There are three types of ethnographical approaches: holistic, semiotic, and critical. First, the holistic approach means that the researcher must “go native”. Second, the semiotic approach focuses on providing a thick description of the finding’s significance. The researcher does not need any empathy with their subjects. Third, critical ethnography means the researcher tries to uncover what is hidden and unspoken in an organization. (Czarniawska, 2007). The critical type of ethnography was chosen for this dissertation to uncover the phenomenon of SDM CC behavior in the organization.

5.2.1. Collecting empirical material

The subjects in this dissertation are three SD teams that I as a researcher observed pursuing their daily work in relation to what they claimed that they were doing. An important aspect to note is that the CC phenomenon appears as a group behavior. Thus, my interest has been in the SD team and their activities as a collective, and that is what was observed. Thus, I did not assess individuals in the teams, as it was the collective recurrent action
patterns carried out by the developers that were addressed. My position was therefore to observe the practice of an organization’s daily work.

Czarniawska (2007) stated that ethnography provides the flexibility of seeking certain types of structured data. The richness of an ethnographical study is the details in the data; everything is written down in a diary e.g., field notes, with different character. The diary further develops over time, throughout the observations as each action takes place. I took field notes that later were written down in an observation protocol; more details will be discussed later about the data collection techniques. In general, it takes time to understand a certain language and also understand what is going on (Czarniawska, 2007). This study was no exception. I spent the first months in understanding Delta’s business language and the activities carried out; both in the separate SD teams and the general activities carried out at Delta. However, spending time to understand their setting helped me to understand what activities I should focus my attention towards.

Observations are fundamental in ethnographical studies (Van Maanen, 2011). This approach was chosen primarily to enable an in-depth understanding of the entire decision chain and the company’s value stream. As Myers (1999) stated,

> Ethnographic research is characterized by the researcher spending extended periods of time at the research site observing what people are doing there as well as listening to what they say they are doing.

In terms of observations, there are two different types. The first type of observation is indirect observation. It is observations done from behind a mirror or using a hidden camera. This is usually used in social work or psychology. Also, in criminology this happens without the study subjects being aware of it. The second type of observation is direct observation, which includes e.g., open video. It can be divided into either participant or non-participant observation.

I used direct participant observation for my data collection. Participant observation is described as when the researcher is onsite, taking part in the subjects’ work (Czarniawska, 2007). In this case, I participated in Delta’s meetings and shadowed its SD teams in their daily work by observing them and asking questions, as unstructured interviews (Czarniawska, 2007).

Since the focus of this study was to understand what the SD teams were doing, daily observations were needed for the same reason of identifying recurrent actions. If the observations were not carried out on a regular basis, it would be difficult to grasp the bigger picture of the SD teams’ actions in their use of SDM practices. Also, it would be difficult to understand both what happened and the reasons why if I did not ask questions to explore the observed situations.
To grasp the SDM CC phenomenon and identify what the three SD teams were doing, participant observations were carried out by the researcher for three years onsite. Observations of the SD teams’ daily work and the company’s management meetings were conducted. The latter gave the opportunity to understand how Delta used the SDMs at an overarching level and the relations between the SD teams and the management. Regarding the observations, I took a passive role in the room, observing the team members’ discussions and actions. If I had taken an active role, there would have been a higher probability of me affecting their actions, which could have led to not capturing their actual SDM use. These observations enabled me to identify what happened and to some extent why. I took notes using an observation protocol (see Appendix C), capturing actions related to the SDM practices and any other actions carried out in the room. An overview of the ethnographic documents created can be found in Appendix E.

To better understand the reasons why these actions occurred, interviews were conducted with the SD team members as a "follow-up" session to clarify/follow-up developers’ thoughts and reasons for why they pursued these types of actions. The follow-up interviews were necessary to collect data allowing the reasons behind the actions to become visible and, further, to analyze and identify potential CC situations. The interviews were semi-structured to both keep them focused on the main questions of what and why, but also open for the SD team members to share additional input that could have provided fruitful information for the study (Kvale and Brinkmann 2009). The questions asked were based using an interview guide (see Appendix B) developed based on the previous observations (Kvale 1994).

In addition to the empirical data collected from observations and interviews, documentation about the company’s SDM was collected. The business documentation entails important information about Delta’s SDM goals and intended use of the SDMs. The documentary sources stating the intended SDM use, specifically the elicited goals, are, in this dissertation, referred to as Delta’s reference point.

5.2.1.1. Access to Delta’s daily activities
From an ethnographer’s view, to capture the phenomena or the actions carried out, you need to observe and shadow the subjects. At the same time, you should not influence the subjects, which means that you need to take a step back and not interfere.

One crucial aspect allowing a researcher to carry out such a study is to gain access (Czarniawska, 2007). For this case, and as discussed in Chapter 4, I was given access to a large international manufacturing company which welcomed me into its daily work. I started out by first being open about what I was supposed to do as a researcher at their company. I provided
information about the purpose of the study and what they could expect from having me there.

It is highly important to have a common understanding of the mutual agreement about what a researcher is supposed to do, as the observation could be seen as uncomfortable for employees. Therefore, when I started my empirical work, I introduced myself to all the employees in the SD teams at Alpha’s unit Delta. I told them that when I shadowed, my focus would be on the teams’ collective actions in relation to the SDMs that they were using. Thus, I explained clearly from the start that my role was not to study them individually and with any type of judgement.

To become part of Delta and the observed teams I, already from start, reflected on the appropriate dress code from the start. This meant that I observed the informal dress code at Delta and adopted it. This was important for becoming a part of the group and blending in.

I began by participating in Delta’s ASDM training three months before I started my observations. This was a good opportunity to get to know them and learn their business language. I also participated in their coffee breaks where they discussed various topics, topics often not related to their work. In addition, I always brought my own lunch, i.e., lunchbox, to be able to sit down together with them during lunchtime. I participated in their outdoor walks after lunch, when they took a break to get fresh air and discuss their ongoing thoughts. I always showed up on time to their meetings and listened to whatever they had to say.

5.2.1.2. Data collection techniques
As mentioned earlier, the data collection was conducted through participant observations, unstructured interviews, collection of business document sources, and follow-up semi-structured interviews.

Observations
One difficulty throughout the data collection was notetaking, since I was constantly on the move and shadowing software developers in their daily work. I therefore created an observation protocol (Fig. 16) during the first six months that guided my notetaking. Whenever I sat down, I took notes. At times when I was alone, I tried to audio record my own reflections and, in the end, write up as much as possible about what I saw throughout the days of observations.
Figure 16 shows an example of my observation notes (in Swedish, see Appendix C for an English version). I used the observation protocol, filling the template with details of each observation. Details from the observations included: the specific activity, when the observation took place, who I observed, what the team members did, and the reasons why. The right column contains “Other comments”, where I wrote the ideas I had at that time, e.g., if the activity was considered formal or informal. A formal activity would be for example, a daily scrum meeting, i.e., a planned ASDM practice. An informal activity, on the other hand, could be, for example, a conversation during a coffee break. In Figure 16, some parts are masked so as not to disclose the organization and/or any of the SD team members.

As Czarniawska (2007) described, the technique of shadowing becomes fieldwork on the move, a way of doing research that captures the mobility of contemporary life. It offers a unique opportunity for self-observation and self-knowledge. The difficulties of the shadowing technique are that it requires constant attention and continuous ethical decisions. I elaborate further on the major ethical concerns and decisions in section 5.5 Ethical considerations.

Unstructured interviews
Along with the observations, I carried out several unstructured interviews. These are another example of informal activity, as they were rather spontaneous. My purpose with them was to ask questions to clarify the

---

**Figure 16. Observation protocol**

<table>
<thead>
<tr>
<th>Aktivitet</th>
<th>Välj</th>
<th>Värde</th>
<th>Ovrigt</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>Gär igenom de nya priorititeringarna som kommit upp på föredagens Scrum/Scrum notes,</td>
<td>Teamet har fått utv, för att allokeras mer resurser till teamet. Kommer från teamet.</td>
<td></td>
</tr>
<tr>
<td>Pilsar problem med testing, ska jämföra med handling, för att se till de får samma eller olika utfall. Kan av. börja på versionerna.</td>
<td>Besöker en vad som har gjorts och vad som är tänkt att genomföras under dagen, samt vem som eventuellt kan hjälpa till för att lösa problemet.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unstructured interviews:
Along with the observations, I carried out several unstructured interviews. These are another example of informal activity, as they were rather spontaneous. My purpose with them was to ask questions to clarify the
Observations 181016 - tisdag

<table>
<thead>
<tr>
<th>Aktivitet</th>
<th>Intervju</th>
<th>Team</th>
<th>Formellt/Informellt</th>
<th>Ovrigt &quot;frivillighet&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vem</td>
<td>Vad/Svar</td>
<td>Varför</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRÅGOR:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fråga: Vilka färger som används för att få upp den aktuella</td>
<td>Se anteckningar</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* SM = Scrum master

- Kanbantavlan / Sprinten / Backloggen?
  Har en sk. Levande tavla där alla case ligger inlagda (just nu totalt 80 case). Som härstammar från Fira.

- Hur fungerar tavlan?
  Daily scrum

- Vilka typer av möten används?
  Kan inte svara på det, vet ej.

- Vilken är din roll?
  Testare, "huvudtestare"

- Finns det något planeringsmöte?
  Nej, det vi har är daily scrums. Vilket vi haft sedan innan jag började jobba här (där sedan).

- Upplever du att dagens sprint är tydlig i vad som ska göras?
  Nej, just nu är det väldigt oövergrip. Bara för ett år sedan var det stor skillnad. Dock inte gällande möten, men i vad som ska göras (i vilken ordning), vad som är prioriterat osv.

- Har teamet något retrospektiv, där man tittar tillbaka och utvärderar hur sprinten gått?
  Nej.

- Finns det någon stor Kanban tavla som ni kan titta på
  Nej, det har vi inte, det skulle vara bra att ha. Det är inte upplagt på det sättet för oss, det finns tydligare

- Finns det någon stor Kanban tavla som ni kan titta på
  Detta skulle hjälpa utvecklarna en hel del för att öka

Figure 17. Notes from unstructured interviews during observations

Documentary sources

Along with my observations, documentary sources were collected. The documentary sources (see Appendix D) consisted of Delta’s organizational structure, SDM descriptions, and guidelines for the software developers. The documentary sources were used to understand the organization and, most importantly, to identify Delta’s goals with ASDMs. The SDM descriptions found in Delta’s business documents were used as the reference point in the data analysis when identifying SDM CC situations.

Semi-structured interviews

During the final year of data collection, follow-up semi-structured interviews were conducted with nine of the SD team members. I developed an interview guide (see Appendix B) which focused on each ASDM practice that Delta was using. The reasons for these interviews were to both confirm the observations during the three years period and to capture the reasons why the recurrent actions happened.
I reached out to all the 24 SD team members by email, asking for their participation in follow-up interviews. I ended up with nine SD team members, three from each SD team. These nine semi-structured interviews took about 1-1.5 hours each. They were held digitally through Zoom, as the COVID-19 pandemic in 2020 did not allow physical meetings. Table 11 shows details of the follow-up interviews. To keep the confidentiality of the team members, their individual roles are not presented. Each interview was recorded and thereafter transcribed.

At the beginning of the interviews, each SD team member was asked for consent to be recorded. In general, the observations focused on understanding what the SD teams were doing (their social actions during SDM activities), and the interviews focused on understanding why the SD teams acted in a certain way (their motive behind their actions during the SDM activities). Both data collection techniques were needed to study the SDM CC behavior.

### Table 11. Follow-up interviews

<table>
<thead>
<tr>
<th>Interview no.</th>
<th>Interview length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1h</td>
</tr>
<tr>
<td>2</td>
<td>50 min</td>
</tr>
<tr>
<td>3</td>
<td>1h</td>
</tr>
<tr>
<td>4</td>
<td>1,5h</td>
</tr>
<tr>
<td>5</td>
<td>1,15h</td>
</tr>
<tr>
<td>6</td>
<td>1h</td>
</tr>
<tr>
<td>7</td>
<td>1,05h</td>
</tr>
<tr>
<td>8</td>
<td>1,5h</td>
</tr>
<tr>
<td>9</td>
<td>1h</td>
</tr>
</tbody>
</table>

5.2.2. Theorizing SDM cargo cult

In my ethnographic study, the process of theorizing the SDM CC phenomenon was an integrated part of the abductive strategy (Bajc, 2012). The abductive strategy allowed me to constantly compare theory and empirical material. How theorizing was situated in my research is illustrated in Fig 18, which is described in detail in this section.

First, there are a couple of fundamental aspects that must be outlined to explain the content of my theorizing. In short, this section summarizes the process of how the SDMCCT was developed. SDMCCT draws on a combination of three theories: social action theory (SAT)(Weber, 1947), self-determination theory (SDT)(Ryan & Deci, 2000), and work motivation theory (WMT)(Gagné & Deci, 2005). This theorizing put forth a definition of the CC phenomenon situated in the field of SDM, called SDM CC behavior.
The SDM CC behavior is operationalized in a framework, including a set of ideal stereotypes. These stereotypes capture effective and non-effective use of SDMs by software developers. As the framework evolved during my theorizing, so did the number of stereotypes and their content, and the theorizing results are found in Chapter 6. The theorizing process is described below, although the details of the different stereotypes are provided in papers II-V.

Second, I want to give a short account of the initial starting point of the abductive process, and the input for my theorizing. It started at the entry point “buzzwords” in Fig 18. I collected statements from practitioners in the IT industry who referred to failed projects of agile adoption as cargo cult situations. However, they did not substantiate these statements with further theoretical investigations. The result from this part of the theorizing process is found in Paper II Mäki-Runsas et al. (2019).

By having these statements at hand, I created an overview of the current IS field and found no research had been done on this topic. Further, in a more structured way, I wanted to investigate if anything related to this topic had been researched and so I carried out a structured literature review. Results from this are presented in section 2.5 Related research and in Paper I (Havstorm & Karlsson, 2023).

Third, the phenomenon of CC is originally from the field of social anthropology, as described in section 2.1 Cargo cult phenomenon. The original CC definition was used as a metaphor and input to the first phase of my theorizing to suggest a definition of the CC phenomenon in the field of information systems (IS), and software development method (SDM) in particular. The initial definition of the phenomenon was elaborated into the SDM CC framework, and evolved during four phases of theorization.
Figure 18. Process of abductive theorizing
The four phases of theorizing are summarized in Table 12 below. The leftmost column shows the version of the theory, followed by the input used and thereafter the output for each version. The righthand column shows the publications where the results are presented. The iterations of these phases will now be described further.

Table 12. Summary of input and output from theorizing

<table>
<thead>
<tr>
<th>Phase</th>
<th>Version of SDM CC theory</th>
<th>Input</th>
<th>Output</th>
<th>Output presented in publication no.</th>
</tr>
</thead>
</table>
| I     | V1                       | - Origin. CC definition  
- Buzzwords  
- SAT (Weber, 1947)  
- SDT (Ryan & Deci, 2000) | - SDM CC definition  
- V1: SDM CC framework  
- Four SDM CC stereotypes | II |
| II    | V2                       | - SDM CC definition  
- V1: SDM CC framework  
- 4 SDM CC stereotypes  
- Lessons learned from V1 | - V2: SDM CC framework  
- Two new stereotypes  
a) rational means-end  
b) rational value-based | III |
| III   | V3                       | - V2: SDM CC framework  
- Six stereotypes  
- Empirical data  
- Lessons learned from V2  
- WMT (Gagne & Deci, 2005) | - V3: SDM CC framework  
- Stereotype names | IV |
| V4    | - V3: SDM CC framework  
- Stereotype names  
- Empirical data  
- Lessons learned from V3 | - V4: SDM CC framework  
- Two divided stereotypes  
- Four new stereotypes  
- Two new stereotype names | V |
| Final version | - V4: SDM CC framework  
- Eight stereotypes  
- SDM CC definition  
- Lessons learned from V4 | - Final version of the SDM CC behavior definition | Cover paper |

The abductive theorizing iterations followed the five iterative steps illustrated in Fig 19: theory search, theory proposition, data collection, data analysis, and lessons learned. A detailed description of each iteration can be found in Appendix F). Each iteration started with the theory search step, where complementary theories were searched for in existing research (see Fig. 19). The selected theories were important parts of the theory grounding. For example, at the start of the first iteration the “cargo cult” buzzword and the situations the IT consultants referred to when mentioning cargo cult acted as input for searches. In a later iteration, for example, the lessons
learned, remaining unclassified data or comments from reviewers acted as input in searching for theories. This could be, for example, a need to focus more on group motivation, resulting in WMT.

The theory search helped to, in the second step, create a theory proposition based on the complementary theories and what had been learned so far from the empirical grounding. The output from this conceptual work is: a) the SDM CC definition, and b) the SDM CC framework. Together they constitute the SDMCCT, and each version of the theory acted as the guide for refining the empirical grounding. The third step in the abductive strategy is data collection. This fieldwork included observation, interviews, and gathering of business documents, where the observation protocol and interview guide were based on the latest version of SDMCCT. Thus, in Fig. 19 the SDMCCT is illustrated as “guiding” the data collection step.

In the fourth step, data analysis, the collected data was analyzed using a combination of inductive thematic analysis (Braun & Clarke, 2006; Patton, 2015) and with theory-driven coding. For the theory-driven coding, the framework in SDMCCT was used. A detailed description of the data analysis and the use of the SDM CC framework is presented in section 5.3.3. The (intermediate) results from the data analyses were used as input to write papers (for example, Paper III) and make presentations to researchers and practitioners. The fifth and final step for each iteration is lessons learned. This step focuses on evaluating experiences from executing the data collection and the data analysis as well as feedback from paper reviews and presentations. The aim of this step was to identify any remaining data that could not be classified in a satisfactory manner, i.e., anomalies existed in the classifications, or to identify inconsistencies in the framework and/or the definition. These lessons learned guided the start of the next iteration.

The five iterative steps were conducted in full for each phase, except for the first version of the framework. When developing the first version of the framework, this iteration only included theoretical grounding without
empirical grounding (see Publication II). The main reason was the need to receive feedback from the research community before starting any data collection so as to, if possible, remove initial flaws.

5.2.3. Data analysis
Empirical data from the participant observations, interviews, and documentary sources provide a comprehensive view for uncovering CC in SD, using the SDM CC framework situations developed in this dissertation.

5.2.3.1. Dimensions of cargo cult analysis
Before we go through the analysis carried out in this dissertation, it is necessary to clarify that analysis of CC could be done by focusing on three different areas, as illustrated in Fig. 20.

First, the analysis can be done between the organization’s SDM description, and the recurrent social actions executed by the software developers. This is illustrated as A in Fig. 20. Thus, this type of analysis focuses on the use of the intended goals of the SDM, which could be a tailored version of the SDM. The intended use of the SDM is called the reference point.

Second, the analysis can address the relationship between the SD team’s SDM use, i.e., its recurrent social actions, and the selected formalized SDM description. Such analysis would mean accessing whether the software developers’ use follows the method by-the-book. This is illustrated as B in Fig. 20.

Third, the analysis can focus on the relationship between the organization’s SDM description and the selected formalized SDM description, illustrated as C in Fig. 20. This type of analysis addresses any potential conflicts.
that can occur in SDM tailoring. Thus, CC is identified between the original SDM description and the study objective’s method description, analyzing whether the actions of the organization’s method designer do not meet the goals of the original SDM description.

This dissertation focuses on the CC assessments in SD teams, addressing any potential CC between the recurrent social actions and the organization’s SDM description. Thus, this dissertation focuses on the relationship A in Fig. 20, and the analytical procedure addresses how to uncover SDM CC behavior between “recurrent social actions” by SD teams and “organization’s SDM description” i.e., its reference point. The following section will present the analytical procedure carried out in this dissertation.

5.2.3.2. Use of the SDM CC framework for data analysis
The data analysis in this dissertation was qualitative. It involved several iterations, including moving back and forth between the empirical data and the theorizing of both the framework and the definition of SDM CC.

The analyzed empirical material in this dissertation focuses on four ASDM practices used by all three of Delta’s SD teams on a regular basis. Taken together, these practices generated extensive empirical data. The four analyzed ASDM practices are daily scrum meetings (DSM), sprint demos (SPD), continuous integration (CI), and visualization (VN).

First, to identify recurrent actions, the analysis procedure was carried out using inductive thematic analysis (Braun & Clarke, 2006; Patton, 2015). Second, once the recurrent actions were identified, they were analyzed and classified using a theory-driven approach, using the different versions of the SDM CC framework; the different versions of the framework can be found in publications II-V as well as in Chapter 6, Table 12.

To identify and classify the recurrent actions within the empirical material, I followed four analytical steps iteratively. Below, I present these steps in detail. Also, it is important to note that the analytical steps were carried out in parallel with my data collection.

Step 1 – Identifying reference point
The first step focused on identifying the reference point for each studied ASDM practice, i.e., Delta’s claimed SDM goals for each practice. The reference point was found in Delta’s SDM description. Thus, I used the documentary sources as input to understand the contextualized versions of scrum, kanban and DevOps. One important note to remember is that the focus of this analysis is to understand the ASDM use in relation to the organization’s method description (relationship A in Fig. 20). Thus, I acknowledge that an SDM can be customized to organizational characteristics and needs.
In publications III-V, the intended goals with the analyzed ASDM practices were identified. For example, in Publication III (Mäki-Runsas, 2019) the studied ASDM practice was DSM, and one of the goals with Delta’s DSM was: “to communicate daily status, progress, and plans”. In Publication V, (Havstorm et al. (submitted), one of the studied ASDMs practices was SPD; one of Delta’s goals with SPD was: “The purpose of the sprint demo is for the team to present to the product owner and the stakeholders functionality that is done”.

Step 2 – Eliciting recurrent actions and identifying categories
The second step focused on eliciting recurrent actions performed by the SD teams, i.e., to identify categories. The inductive thematic analysis procedure was used to create categories (Braun & Clarke, 2006; Patton, 2015). These categories were created by coding the observation protocols. According to Weber (1947), social actions are carried out “for someone and something else”. The social actions of interest were recurrent ones executed by software developers with the intention to contribute to the SDM practice (i.e., for someone) performed by the SD teams.

For example, in Publication IV (Havstorm et al. (2023), one identified recurrent action was DSMs that got frequently cancelled by the SD teams. During my thematic analysis, categories were identified, which were both merged and demerged during the iterative analysis to capture uncovered details.

Step 3 – Classifying categories into deviations and non-deviations
The third step focused on classifying the recurrent SD action categories into deviations and non-deviations from the reference point. The coding was done by using the most abstract level of the SDM CC framework, i.e., structuring of the stereotypes as CC and non-CC (see Table 15 in section 6.4 “Final version of SDM CC Theory” for details).

This meant that the recurrent actions were sorted into two separate clusters:

1. Recurrent SD action themes that deviated from the reference point (i.e., deviations), for example, “We just receive an email notification in the morning if our team manager cannot come and then we assume that it is cancelled.” – Team member 6.
2. Recurrent SD action categories that aligned with the reference point (i.e., non-deviations); for example, “I upgraded to Windows 10 and ended up with the same problem as … before. I just want to indicate that this will probably take the whole day to fix.” – Team member 3.
Step 4 – Refining and sorting analysis
The fourth step focused on refining the analysis of the recurring SD action categories by sorting them using the fine-grained elements of the SDM CC framework, i.e., the actual stereotypes (see Table 15 in section 6.4 for details). The aim of this step was to understand why the recurring actions were performed.

I used the interviews and, where possible, the observations for this interpretation. In accordance with the analytical structure of the SDM CC framework, interpretations were based on: a) the type of rationality behind the recurrent action (if any), and b) the type of motivation underlying the recurrent action. This step resulted in each recurrent action category being associated with stereotypes from the SDM CC framework.

In some cases, this step involved identifying sub-categories within the categories, when it was found that certain recurrent actions belonged to different stereotypes.

5.3. Procedure of systematic literature review (SLR)
Publication I Havstorm and Karlsson (2023) presents a systematic literature review (SLR). This study surveyed existing SDM research to scrutinize the current knowledge base on software developers’ types of reasoning behind SDM adoption and use. The literature review was conducted between 2016-2023 and included peer-reviewed journal and conference articles and book chapters regardless of the geographic region. The articles were published, in English, and up until the first quartile 2022. Scopus was used as the search database, which includes IEEE and ACM papers, and is the largest database (Franke & Brynielsson, 2014).

The search queries ended up with 72 combinations of keywords, which evolved during the reading of papers as additional keywords were needed. One example of a search query was “software development method” AND “use”. Havstorm and Karlsson (2023) include an appendix summarizing all the 72 keyword combinations.

The identified research was selected in four stages (Fig. 21) where the first stage consisted of a search result with a gross list of 3,227 papers. The gross list was sorted in the second stage and duplications were removed, to end up with 1,619 papers. In the third stage, the net list became 516 papers that were further assessed and analyzed with an analytical framework developed in the paper.
During the detailed analysis in stage four, we used inclusion and exclusions criterions (Table 13); for example, the papers must be empirical studies. We ended up with 111 primary papers that were finally included in the actual analysis.

Table 13. Inclusion and exclusion criteria in SLR (Havstorm & Karlsson, 2023)

<table>
<thead>
<tr>
<th>Inclusion criterion</th>
<th>Exclusion criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1. The paper is written in English.</td>
<td>E1. The paper is an editorial.</td>
</tr>
<tr>
<td>I2. The paper is a peer-reviewed journal paper, conference paper, or book chapter.</td>
<td>E2. The paper is an introduction to a conference proceeding.</td>
</tr>
<tr>
<td>I3. The paper focuses on SDM as a study object.</td>
<td>E3. The paper is a literature review.</td>
</tr>
<tr>
<td>I4. The paper provides an empirical account, i.e., data, measurements, observations, or descriptions about adoption and/or use of SDMs.</td>
<td></td>
</tr>
</tbody>
</table>

The data extracted from each paper were: (1) full reference (author, year, title, source name), (2) source (journal, conference, book chapter), (3) research question/aim, (4) research method as classified by the authors of the paper, (5) phases addressed (adoption, use), (6) type of reasoning (rational, irrational, non-rational, unable to identify), and (7) research themes addressed. For the more complex data extraction (5 to 7), an integrative thematic synthesis was conducted following Cruzes and Dybå (2011) to learn more about the characteristics of the existing research. The thematic process consisted of five steps: 1) extract data, 2) code data, 3) translate code into themes, 4) create model of higher-order themes, and 5) assess the
trustworthiness of the synthesis. For the theory-driven part of the integrative thematic synthesis, we developed a classification framework consisting of phases (SDM adoption and/or use) and type of reasoning. The latter part draws on SAT (Weber, 1947). For a more detailed description of the SLR procedure, please see Publication I (Havstorm & Karlsson, 2023).

5.4. Credibility and transferability

To reach credibility, this dissertation demonstrates findings from a longitudinal ethnographic study, where the research took place onsite for three years. Those years helped provide an understanding of the organization and the SD teams’ culture, to build trust onsite with the software developers, and to repeat the observations continuously by shadowing similar activities, e.g., scrum meetings by the management team, coffee breaks, SD teams’ DSMs. The repeated observations helped in studying the CC phenomenon.

The data collection followed the research method “ethnography” tradition in qualitative research, which included shadowing, observations, and interviews, besides collecting documentary sources. This method allowed for empirical material triangulation, i.e., field notes, observation protocol, interviews, and business documents. The triangulation also enrolled co-authors in peer debriefing by describing the analysis and conclusion. Other co-workers were also involved, as the work in progress was presented to them so as to receive input and to be transparent with the research.

During the empirical material collection, consent forms were provided to all software developers to guarantee the trustworthiness of informants. In addition, the consent form provided information about the research study and how to withdraw the consent for any reason. The consent form can be found in Appendix A.

Both the research setting and the conducted research method have been presented, consisting of necessary detailed information to understand the research context, Alpha, Beta, and Delta, of this dissertation and how both data were collected and the SDMCCT was developed.

To meet the criteria of transferability, this dissertation provides an in-depth description of the case study, the overall organization, its unit, and the characteristics of the SD teams and its software developers. Unfortunately, the nature of qualitative methods does not provide randomized samples isolated from a situation or specific context, which means that the knowledge created in this dissertation cannot be generalized (Tracy, 2010).
Qualitative research engages in-depth studies that generally produce historically and culturally situated knowledge. As such, this knowledge can never seamlessly generalize to predict future practice. (Tracy, 2010)

However, the knowledge created and generated in this dissertation can be transferred and used in other settings or organizational cases, as the dissertation generates theoretical and practical knowledge (Tracy, 2010).

5.5. Ethical considerations
To reduce and prevent fraudulent scientific research, ethical scrutiny is crucial. When entering an organization, it is highly important not to disclose any kind of information that is regarded as classified. Therefore, a confidentiality agreement was signed with Alpha including its expectations from this project and vice versa.

Under the European legislation, General Data Protection and Regulation (GDPR), all data processing activities shall be registered. For research where personal data is processed, consent must be collected from each person involved. Therefore, a formal consent form consisting of the procedure for obtaining and withdrawing the consent was described, distributed, and collected from everyone in the different SD teams at Delta.

Since this dissertation consists of an ethnographical approach for data collection, there are different aspects to consider in terms of ethics, which can be found in “Research Ethics in Ethnography/Anthropology” (AcSS, 2021).

(1) The trust relationship. The ethnographic approach entails a form of relationship with research subjects that tends to have a more humanistic nature. This relationship might affect the subject’s actions.

I did not, during the observations, act as one of the systems developers, I was only acting as a part of the team and observed how members carried out their work. I did not interfere or report my observations until the final analysis was made. The reason for this was that I did not want to affect their work.

(2) Funders’ motives. Who is funding the research is crucial in terms of what the different actors might expect from the findings. Therefore, the theoretical perspective of this dissertation was explicit from the start of the collaboration with Alpha and its Delta unit. The funder of this research is not the company in the selected case. The research funder, the National Research School of Management and IT, is completely independent of the company.

(3) Judgements. Balancing between being a member of the subjects i.e., the SD teams in Delta Sweden, and collecting data could be a complex work. This longitudinal study was carried out for three years and with three SD teams and one management team. It took time to understand the business
language and their routines, which was a challenging task. After some time, a couple of months onsite, I became familiar with the teams’ daily work and kept myself focused on understanding it, rather than trying to compare or judge their procedures. Hence, the observation protocol and the interview guide helped me to keep this focus so as to minimize judgements.

(4) *Professional training.* It is important to get to know the people the researcher plans to study. In this case, I built up a relationship over four months, starting out by having a meeting first with one of the managers at the Delta unit, thereafter a meeting with the manager again, this time together with the head of Delta and one of my research colleagues, as described earlier in the section about gaining access. After these meetings, I followed the Delta unit’s training in ASDMs, which was a full day with lectures and exercises for each employee at Delta. During this training, I had a chance to interact with the subjects and study their culture in terms of dress code and language, which I later applied when I started the observations.

(5) *Respecting the subjects.* When understanding the culture of the subjects, i.e., the team members in Delta, it is easier to formulate research questions and how to research these. My responsibility was to avoid and minimize any harm in each step taken in the data collection process. Hence, I was continuously favorable toward all their concerns, thoughts, and procedures. For example, when approaching the SD team members, I always asked if they had time to reflect, and I kept open minded when they came by to ask questions. The same applied to the times that team members wanted to brief and reflect upon previous meetings or situations; I let them talk without interrupting them and, after that, I asked follow-up questions to make sure I understood their views or feelings.
PART IV – FINDINGS
Chapter 6. Software Development Method Cargo Cult Theory

In the process of understanding and characterizing the phenomenon of SDM CC, this dissertation has theorized this phenomenon by combining SAT with WMT. Based on the abductive research process, the theorizing has resulted in a definition and eight stereotypes that describe SDM CC situations and non-CC situations. This chapter presents the SDMCCCT i.e., the definition and the framework, including its stereotypes, that evolved in four publications, papers II-V. The chapter, therefore, summarizes the theoretical results from the four papers and ends with a refined version of the SDM CC definition.

6.1. Initial definition and first version of the theory

The first theorizing phase and iteration resulted in an early definition of SDM CC together with a framework which was presented in Publication II (Mäki-Runsas et al. (2019)).

The aim of the paper was twofold: a) to define the concept of information systems development method cargo cult, and b) to suggest an analytical framework that could be used to identify information systems development methods cargo cult situations in SD organizations. The paper builds on the idea that situations of unsuccessful SDM use, where SD teams lack understanding of the SDMs or misinterpret the SDM, can be characterized as SDM CC behavior.

The conceptual development was a result of combining SAT (Weber, 1947) and SDT (Ryan & Deci, 2000) with the CC phenomenon to frame CC in relation to SDM use. The combined theories, SAT and SDT, are described earlier in Chapter 3 “Theoretical foundations”, as well as the CC phenomenon in section 2.1 “Cargo cult phenomenon”. The following definition of SDM CC behavior was presented in Paper II:

An information systems development method cargo cult is a temporarily delimited dysfunction that can have a non-rational or an irrational foundation. It leads to misconceptions and/or malpractices and can be intrinsically and/or extrinsically motivated as an information systems development team unconsciously fails in an attempt to replicate the circumstances and success of others. (Mäki-Runsas et al. (2019)

In short, based on this definition, the paper put forth four distinct SDM CC stereotypes in a conceptual framework. This framework is shown in Fig. 22. The framework aims to be used for analyzing and assessing if an SD
organization faces CC situations in its SDM use. This was the first version of the framework.

Starting in the upper-right corner of Fig. 22, the first SDM CC stereotype is the non-rational affectual. This CC behavior represents situations where software developers act upon emotions, where it is possible to observe a so-called bandwagon effect. A bandwagon effect means that the SD teams imitate others by following a current trend without any underlying reasons for the actions. SD teams act upon emotions, and in this paper, such actions are described as non-rational affectual and having intrinsic orientation of motivation.

Moving clockwise in Fig. 22, the second SDM CC stereotype is the non-rational traditional. This CC behavior represents situations where SD teams act upon old traditions without thinking. By keeping old habits, the SD teams follow their old traditions without any underlying reasons for their actions. SD teams that act upon old traditions are, in this paper, described as employing non-rational traditional actions that have extrinsic orientation of motivation.

The third SDM CC stereotype, in the bottom-left in Fig. 22, is the irrational value-based. This CC behavior represents situations where SD teams seek to, but fail in adhering to, the SDM values and ignore the practices needed to achieve the values. When SD teams act but fail to adhere to the values, such actions are, in this paper, described as irrational value-based actions and having intrinsic orientation of motivation.
Finally, the fourth SDM CC stereotype is the **irrational means-end**. This CC behavior represents situations where SD teams replicate and imitate others’ SDM use but fail in their attempts due to not having a proper understanding of the SDM. Actions where the SD teams replicate others or use the intended SDMs in a faulty way are, in this paper, described as irrational means-ends actions having extrinsic orientation of motivation.

The conceptual paper was the first attempt to frame the CC concept in relation to SDM use, where each type of social action was combined with an applicable type of motivation i.e., intrinsic or extrinsic. Each SDM CC stereotype was connected to practitioners’ buzzwords from the IT-industry, as shown in Fig. 22. These buzzwords were used by the practitioners when describing different situations of failed SDM use.

The definition of SDM CC behavior only focuses on CC situations, which were turned into the framework. Reviewers of the paper commented that possibilities to capture successful SDM use were missing. In terms of the framework, such use would be characterized as non-CC behavior that is based on rational social actions.

### 6.2. Second version of the theory

The second theorizing phase and iteration resulted in a refined SDM CC framework which was presented in Publication III (Mäki-Runsas (2019). The aim of this paper was to “as a first attempt, test the ISDM cargo cult framework by using data collected throughout observations and interviews in an ISD organization using agile methods”.

This paper presents an extended version of the theory. I received review comments from researchers on the first version of the theory presented at the ISD2018 conference, on how it was possible to identify successful SDM use situations. Thus, I decided then to add non-CC situations to the framework, to capture the **rational** type of social actions in accordance with Weber (1947). This paper was the first attempt to empirically test the framework. Therefore, I wanted to provide situations of non-CC behavior to show how this framework could be useful for distinguishing between CC behavior and non-CC behavior.

The framework was adjusted further by removing the dimension of rationality from the first version of the framework due to non-rational, irrational, and rational not being considered to belong to one dimension. Instead, they refer to three distinct types of social actions, as shown in Fig. 23. Thus, the rationality dimension in Fig. 22 was replaced by “type of social action”, having rational social actions placed in the middle to capture the...
difference between non-rational and rational, as well as between rational and irrational social actions.

The empirical results in the paper only covered one motivational aspect of the framework: extrinsic motivation. The data used in this paper was very limited, focusing on only one SD team and its use of one ASDM practice. An important lesson learned from the empirical work was the difficulty of observing intrinsic motivation; in this case, it was not possible to observe. This difficulty was fed into the next theorizing phase. This paper ended with a suggestion for future research, which was to extend the analysis with empirical data from multiple teams and SDM practices.

### 6.3. Third version of the theory

The third theorizing phase resulted in another refined SDM CC framework and a set of six stereotypes, presented in Publication IV; (Havstorm et al., 2023).

The aim of this paper was “to refine the SDM CC framework and empirically test this version of the framework”, and the paper presents the third version of the SDM CC framework. As shown in Table 14, the SDM CC framework now includes six stereotypes: four CC stereotypes and two non-CC stereotypes. Each stereotype has a specific type of rationality based on interpretations of Weber’s (1947) SAT and a specific type of motivation based on interpretations of Gagne and Deci’s (2005) WMT.

The major refinement of the framework is the names given to each stereotype and the replaced motivational aspects. Based on the lesson learned, that intrinsic motivation was not present and therefore difficult to capture,
it was replaced with internalized regulated motivation, and amotivation, based on WMT (Gagné & Deci, 2005). The description of each stereotype was also refined, as presented after Table 14 below.

Table 14. SDM CC framework version 3 (Havstorm et al., 2023)

<table>
<thead>
<tr>
<th>Stereotypes</th>
<th>Rationality</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cargo cult</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conventional</td>
<td>Non-rational</td>
<td>Amotivation</td>
</tr>
<tr>
<td></td>
<td>Traditional</td>
<td></td>
</tr>
<tr>
<td>Spontaneous</td>
<td>Non-rational</td>
<td>Amotivation</td>
</tr>
<tr>
<td></td>
<td>Affectual</td>
<td></td>
</tr>
<tr>
<td>Conception</td>
<td>Irrational</td>
<td>Internalized regulated</td>
</tr>
<tr>
<td></td>
<td>Value-based</td>
<td></td>
</tr>
<tr>
<td>Fragmental</td>
<td>Irrational</td>
<td>External regulated</td>
</tr>
<tr>
<td></td>
<td>Means-end</td>
<td>Internalized regulated</td>
</tr>
<tr>
<td>Non-cargo cult</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credence</td>
<td>Rational</td>
<td>Internalized regulated</td>
</tr>
<tr>
<td></td>
<td>Value-based</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>Rational</td>
<td>External regulated</td>
</tr>
<tr>
<td></td>
<td>Means-end</td>
<td>Internalized regulated</td>
</tr>
</tbody>
</table>

The stereotypes are briefly summarized as follows.

**Conventional** – This stereotype refers to situations where it is possible to identify that software developers’ actions are based on old routines and habits. It means the rationale has been forgotten, and old routines and habits are followed without the software developers knowing why or being unaware of why a certain action was carried out previously. Thus, the pursued action lacks intention, which is the same as behaving amotivated (Gagné & Deci, 2005).

**Spontaneous** – This stereotype addresses affectual actions where the software developers’ action is carried out based on emotions, brief ideas, and a desire of the software developers to achieve the success of others. It could also be due to following a current trend without thinking and/or being aware of the deviation from the claimed SDM. Having a sense of being onboard with the current trend, the behavior is considered as amotivating. Consequently, the behavior is an emotive response without rational reasoning.

**Conception** – This stereotype addresses irrational value-based actions. Irrational value-based actions are situations where software developers claim to adhere to the values of the SDM. However, while having internalized regulated motivation, the software developers fail in their understanding of how the values need to shape their work. Misunderstandings or misinterpretations of the SDM lead to irrational actions, as the SD team fails to comply with the SDM values.
**Fragmental** – This stereotype captures means-end actions where the software developers are trying to achieve a goal (end) by choosing between different actions (means). However, in this case, the software developers’ chosen mean is not suitable for the claimed goal. The software developers perceive external control as the goals affect their actions. The perceived self-determination depends on the software developers’ volition to pursue the SDM practice. The software developers do not follow their own ideas or goals, but are instead told by the management to carry out certain practices within the SDM. Misinterpreted selection of relevant actions leads to irrational actions, which are malpractices, as the fundamental understanding of the SDM goals is non-existent or limited.

**Credence** – This stereotype addresses value-based actions where the software developers’ action is based on a belief in the values of the SDM and the executed action successfully aligns with these values. Consequently, the software developers understand the SDM values claimed by the organization. Furthermore, there is no malpractice. In these cases, the software developers perceive external control, and self-determination is characterized as internalized to the level of identified regulations, where the importance is focused on values and regulations.

**Achievement** – This stereotype addresses means-end actions where the software developers’ action contributes to fulfilling the goal(s) in the SDM. The perceived self-determination depends on the software developers’ volition to pursue the SDM practice. The motives in these situations are categorized as external regulated situations where the SD team perceives external control, or internalized regulated to the level of integrated regulated – which are situations where there is a coherence between the goals, values, and regulations. The latter means the software developers are interested in the goals of an activity, not the activity itself (Gagné & Deci, 2005).

The SDM deviations that were uncovered in Havstorm et al. (2023) were successfully characterized using the SDM CC framework. In total, three out of four CC stereotypes were identified. The empirical data did not provide any social actions related to the conception stereotype. Also, the results in the paper only focused on SDM deviations, i.e., situation captured by the cargo cult situations of the framework. Therefore, Chapter 7 in this dissertation presents empirical findings where non-CC situations have been characterized.

**6.4. Fourth and final version of the theory**

The fourth theorizing phase and iteration resulted in a refined SDM CC framework that includes two additional stereotypes, presented in Publication V (Havstorm et al. (submitted). The final version of the SDMCCCT
includes a refined version of the SDM CC definition, which is presented in section 6.4.1.

The aim of this paper was to refine the SDM CC framework and empirically validate it. This paper presents the fourth and final version of the conceptual framework. Table 15 presents the eight SDM CC stereotypes found in the framework. This version of the framework includes five CC stereotypes and three non-CC stereotypes anchored in Weber’s (1947) SAT and Gagné’s and Deci’s (2005) WMT.

The major refinement of the framework is modifications related to the stereotypes “achievement” and “fragmental”. In Havstorm et al. (2023) these stereotypes included both internalized and externalized types of motivations, which made them ambiguous. These two stereotypes were therefore split into two stereotypes, so each of them only referred to one type of motivation. As a result, two new stereotypes, called “accomplishment” and “deficient”, were added to capture the internalized type of motivations. They capture rational and irrational means-end social actions, respectively. The description of each revised stereotype was also refined; the new and the refined descriptions of the stereotypes achievement, accomplishment, fragmental, and deficient, are presented after Table 15.

Table 15. SDM CC framework version 4 (Havstorm et al., submitted)

<table>
<thead>
<tr>
<th>Cargo Cult</th>
<th>Stereotypes</th>
<th>Type of social action</th>
<th>Type of motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Achievement</td>
<td>Rational means-end</td>
<td>External regulated</td>
</tr>
<tr>
<td></td>
<td>Accomplishment</td>
<td>Internalized regulated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Credence</td>
<td>Rational value-based</td>
<td>Internalized regulated</td>
</tr>
<tr>
<td>Yes</td>
<td>Fragmental</td>
<td>Irrational means-end</td>
<td>External regulated</td>
</tr>
<tr>
<td></td>
<td>Deficient</td>
<td>Internalized regulated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conception</td>
<td>Irrational value-based</td>
<td>Internalized regulated</td>
</tr>
<tr>
<td></td>
<td>Conventional</td>
<td>Non-rational traditional</td>
<td>Amotivation</td>
</tr>
<tr>
<td></td>
<td>Spontaneous</td>
<td>Non-rational affectual</td>
<td>Amotivation</td>
</tr>
</tbody>
</table>
Achievement – The achievement stereotype addresses rational means-end actions where the software developers’ action contributes to fulfilling the goal(s) in the SDM. The perceived self-determination depends on the software developers’ volition to pursue the SDM practice. The motives in these situations are categorized as external regulated situations where the SD team perceives external control – which are situations where there is a coherence between the goals, values, and regulations.

Accomplishment – This stereotype also focuses on rational means-end actions. It share similarities with the achievement stereotype in that the actions are rational. The difference is that the actors’ selected goals are internalized regulated to the level of integrated regulated. Thus, these are situations where the SDM goals have been fully assimilated by the software developers. It means the software developers are interested in the goals of an action, not the action itself (Gagné & Deci, 2005).

Fragmental – The fragmental stereotype captures irrational means-end actions where the software developers are trying to achieve a goal by choosing between different actions. However, in this case, the chosen action is not suitable for the claimed goal. The software developers perceive external control, as the goals are the controlling factor that affects their actions i.e., external regulated. The software developers do not follow their own ideas or goals but are instead told by the management to carry out certain practices within the SDM. Misinterpreted selection of relevant actions leads to irrational actions, which are malpractices, as the fundamental understanding of the SDM goals is non-existent or limited.

Deficient – This stereotype addresses irrational means-end actions. Again, means-end actions mean that the software developers are trying to achieve a specific goal by choosing between different SD actions. However, in these cases, the chosen SD action is not suitable for the claimed goal. Misinterpreted selection of relevant actions leads to irrational actions. Thus, these SD actions are malpractices, as the fundamental understanding of the SDM goals is non-existent or limited. The SDM goals are not perceived as a controlling factor. Instead, they have been fully assimilated by the software developers, i.e., internalized regulated.
6.4.1. Refined SDM CC definition

The SDMCCCT consists of ideal stereotypes of software developers’ execution of SD actions. The stereotypes capture situations of rational, irrational, and non-rational use of SDM, driven by either internalized regulated or external regulated type of motivation, or lack of intentions i.e., amotivation. In other words, these stereotypes capture both effective use of SDMs and CC.

The initial definition of SDM CC behavior, defined in Paper 1 (Mäki-Runsas et al., 2019), was further elaborated, and is now based on the four phases of the theorizing, which hereby is defined as:

<table>
<thead>
<tr>
<th>Software Development Method Cargo cult</th>
</tr>
</thead>
<tbody>
<tr>
<td>A software development method cargo cult is a temporarily-delimited dysfunction, resting on the foundation of lack of intended reasons. It leads to misconceptions and/or malpractices that are driven by the collective motive as they unconsciously fail to replicate the circumstances and success of others.</td>
</tr>
</tbody>
</table>
Chapter 7. Identified SDM Cargo Cult

The empirical findings of this dissertation have been divided into two parts: identified SDM CC situations and identified SDM non-CC situations. The first part will be presented in this chapter. Here, I summarize the results presented in publications III-V. Publication III provides results from one SD team’s use of the agile practice DSM. Publication IV provides results from three SD teams use of daily scrum meetings. Thus, Publication IV can be seen as an extension of Publication III in terms of data. However, the compilation in this chapter is based on the most elaborated version of the SDM CC framework. Publication V provides results from three SD teams’ use of multiple practices, which are CI, SPD, and VN. This chapter will discuss the overall findings and the patterns found regarding the identified SDM CC situations across these practices. This entailed assessing the results to find prominent patterns among the stereotypes.

7.1. Identified SDM deviations as cargo cult behavior

The four ASDM practices that were analyzed were DSM, SPD, CI, and VN. All these ASDM practices were used by Delta’s three SD teams. The data collected in this thesis consisted of both Delta’s reference points (its method description) for each ASDM practice, identified in its business documents, and recurrent activities in the SD teams’ use of each ASDM practice. The empirical data was analyzed using the SDM CC framework, following the analytical procedure presented in section 5.2.3.2. The analysis resulted in identified deviations compiled in Table 16 as “deviation categories”, followed by a deviation number and finally the classified CC stereotype. Each ASDM practice is further explained below along with its reference point and the results from the analysis, using Table 16.

DSM is an everyday activity “status report” where each SD team meets and discusses the status of the project (Stray et al., 2016). The identified reference point of Delta’s DSM was: “(1) Support multi-team SW [software] development for one application – staged development, (2) Consistency in build process, (3) Enable fast platform development, (4) Support agile development by: fastest possible feedback by automated tests, confidence in quality of releases, support fast delivery of releases by increased quality” (Delta’s SDM description).
<table>
<thead>
<tr>
<th>Agile practices</th>
<th>Deviation categories</th>
<th>Deviation no.</th>
<th>Cargo cult stereotype</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deviation no.</td>
<td>Fragmental</td>
<td>Deficient</td>
</tr>
<tr>
<td>Daily scrum meeting</td>
<td>Bringing irrelevant information</td>
<td>D1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D2</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Canceling meetings</td>
<td>D3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D4</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Disturbing the team</td>
<td>D5</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Receiving unclear information</td>
<td>D6</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D7</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Bringing new requirements</td>
<td>D8</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D9</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Problem-solving</td>
<td>D10</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Task distribution</td>
<td>D11</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D12</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Missing feedback</td>
<td>D13</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D14</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not participating in the demo</td>
<td>D15</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D16</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D17</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not discussing sprint goals</td>
<td>D18</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Unclear responsibilities</td>
<td>D19</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not having opportunity to have a say</td>
<td>D20</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Sprint demos on irregular basis</td>
<td>D21</td>
<td>X</td>
</tr>
<tr>
<td>Sprint demo</td>
<td>Testing the code manually</td>
<td>D22</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D23</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not integrating the code frequently</td>
<td>D24</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not following continuous integration guidelines</td>
<td>D25</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Being inconsistent with continuous integration</td>
<td>D26</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not testing the code</td>
<td>D27</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not fulfilling the definition of “done”</td>
<td>D28</td>
<td>X</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Not updating the scrum board</td>
<td>D29</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D30</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Missing guidelines</td>
<td>D31</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not using forecasts</td>
<td>D32</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not having a continuous flow</td>
<td>D33</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D34</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Not using scrum board</td>
<td>D35</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D36</td>
<td>X</td>
</tr>
</tbody>
</table>
In Havstorm et al. (2023), 12 deviations were captured relating to DSM, divided into seven deviation categories. In Table 14, these deviation categories are: (1) bringing irrelevant information, (2) canceling meetings, (3) disturbing the team, (4) receiving unclear information, (5) bringing new requirements, (6) problem-solving, and (7) task distribution. The CC classification captures prominent CC stereotypes among the deviations. The found CC stereotypes were fragmental, deficient, conventional, and one deviation analyzed as spontaneous.

SPD is an agile event focused on inspecting and adapting. The inspection involves reviewing the sprints, product backlog, increment, marketplace, budget, timeline, and capabilities. The adaption is focused on product backlog and release plan (Jacobson et al., 2022). The identified reference point of Delta’s SPD was: “The purpose of the sprint demo is for the team to present to the product owner and the stakeholder’s functionality that is done, based on the defined sprint goals”. Delta’s SDM description also stated: “The team should call for demos when applicable” (Delta’s SDM description).

In Havstorm et al. (submitted), nine deviations were captured related to SPD, divided into six deviation categories. In Table 16, these deviation categories are: (1) missing feedback, (2) not participating in the demo, (3) not discussing sprint goals, (4) unclear responsibilities, (5) not having opportunity to have a say, and (6) sprint demos on irregular basis. The CC classification captures two prominent CC stereotypes among the deviations: fragmental and conventional.

The CI practice is viewed as one of the cornerstones in agile SD. In CI, each SD team member and the SD team work with code integration following a common structure in both design and development where solutions are supposed to be integrated daily (Soares et al., 2022). The identified reference point of Delta’s CI was: “(1) Support multi-team software development for one application – staged development, (2) consistency in build process, (3) enable fast platform development, (4) support agile development by: fastest possible feedback by automated tests, confidence in quality of releases, support fast delivery of releases by increased quality” (Delta’s SDM description).

In Havstorm et al. (submitted) seven deviations were captured related to CI. These deviations were structured into six deviation categories, as shown in Table 16: (1) testing the code manually, (2) not integrating the code frequently, (3) not following continuous integration guidelines, (4) being inconsistent with continuous integration, (5) not testing the code, and (6) not fulfilling the definition of “done”. The CC classification captures three prominent CC stereotypes among the deviations: fragmental, deficient, and conventional.
VN is a set of agile project artifacts to facilitate project members having a shared view of what is happening in the project. To support this practice, Delta foremost used backlogs and scrum boards. Delta used the software Fogbugz, and later Jira, to support these parts of scrum and kanban. The uncovered reference point of Delta’s VN was: (1) The work should be visualized on a whiteboard or a screen (if it is a multisite development. (2) The team should work with the backlog and have it estimated in Fogbugz or a similar system. (3) The team should identify and invite stakeholders on a regular basis. The transparency in the backlog should be at a level so stakeholders can prioritize between epics/stories. (4) The mandatory score card for each team is the burn-down chart for scrum. (Delta’s SDM description).

In Havstorm et al. (submitted), eight deviations were captured. These were structured into five deviation categories as shown in Table 16: (1) not updating the scrum board, (2) missing guidelines, (3) not using forecasts, (4) not having a continuous flow, and (5) not using scrum board. The CC classification captures three prominent CC stereotypes among the deviations: fragmental, deficient and conventional.

In publications III, IV, and V, the recurrent actions carried out by Delta’s SD teams were classified as deviations, as presented above. Thus, quotes and the detailed analysis can be found in these papers. In total, 24 deviation categories were found, consisting of 36 deviations identified as CC situations across the four ASDM practices analyzed in this dissertation. In the following section, the prominent pattern of these findings is presented and further discussed.

7.2. Prominent pattern in the SDM cargo cult results

The findings in Table 16 consist of classified CC situations. In this section, the prominent patterns of the identified CC stereotypes are presented. I will walk through each CC stereotype in the following order: fragmental, conventional, deficient, spontaneous, and conception. The presented order of the CC stereotypes reflects their prominence.

7.2.1. Fragmental

The fragmental CC stereotype reflects situations where SD teams select between different actions, based on external control, to reach an SD goal (Havstorm et al., submitted).

The findings from Delta’s use of the four ASDM practices (illustrated in Table 16) show that the fragmental stereotype is prominent among the deviations. The fragmental stereotype appears in all of Delta’s four ASDM practices and in relation to several SDM deviations.

The SD teams, according to the analysis, acted irrationally by carrying out malpractices i.e., did not worked in line with their intended use of the
ASDM practices. One such example relates to CI. Even though the SD teams at Delta had guidelines on how, when, and with what frequency the solutions were supposed to be integrated, the results show that the code did not become integrated by the SD team members until they considered it to be fully complete. This is illustrated by the following quote: “Several of us do not integrate until we are satisfied, especially with larger tasks” – Team member 2.

Not integrating the code frequently led to the SD teams having team members working on several different versions of the system, and not the most current version.

The prominent pattern of fragmental indicates that Delta’s use of ASDM practices includes a significant lack of understanding of these practices. This lack of understanding further resulted in inefficient situations that affected goal achievement through SDM use. The SD was negatively affected by not meeting the objectives of the SDM.

7.2.2. Conventional
It is evident from Table 16 that Delta’s previous way of working with SD had not been replaced completely. Conventional situations where the old way of working is prominent in the results. For example, Delta did not use the forecast in its use of visualization that the ASDM provides. Previously, the software developers relied on their project managers, by being told what to do next without having full insight into the project status. This is illustrated by the following quote: “We don’t have burndowns and stuff, and we don’t do any estimation or planning” – Team member 5.

The conventional CC stereotype reflects situations where SD teams act based on habits. These activities are amotivated as the SD teams do not reflect upon their actions and why they are executing them. A tradition and old habit that had been kept, and created CC behavior at Delta, was, for example, that the roles of managers had been kept but renamed as SMs. Another example is confidence was not shown in the teams solving the tasks themselves within the time frame, nor giving new directives and requirements during an ongoing sprint. In these situations, the SD teams were not being motivated i.e., amotivated, as they carried out the SDM practices based on old habits.

7.2.3. Deficient
In contrast to the fragmental stereotype, the deficient stereotype derives from internalized regulated motivation. The deficient stereotype was identified in three out of four ASDMs. For example, during the DSM, when the status reports on the current situations were supposed to be discussed, the SD teams tended to discuss possible solutions. Thus, this meant switching the focus of the DSM to problem-solving. This is illustrated by the following
quote: “We tend to almost every day go into details to solve issues and the meetings gets extended” – Team member 3.

On one hand, the deficient stereotype is not the most prominent pattern. On the other hand, the deficient situations show that Delta’s SDM use consisted of malpractices due to lack of understanding of the SDMs, similar to the fragmental situations. Thus, the difference is the internalized motivation compared to the external control in the fragmental situations. It means that the lack of understanding of the practices becomes a prominent pattern.

Deficient was not found in Delta’s use of SPD practice. One possible explanation for this was the SD teams’ low presence on these occasions.

7.2.4. Spontaneous
The spontaneous stereotype reflects actions based on emotions. As can be seen in Table 16, CC situations belonging to the spontaneous stereotype were identified only during Delta’s use of DSMs. These occasions were when the SD team members got interrupted while they provided their status report. A DSM is, compared to the other three studied ASDM practices, a meeting focused on the status of the SD project. Consequently, this setting probably encourages the participants to naturally engage in the ongoing discussion and opens the possibility of acting spontaneously. The prominent pattern that appears is the absence of actions classified with this stereotype.

7.2.5. Conception
The conception stereotype reflects actions where the software developer’s ambition is to act in accordance with the set of values of the specific ASDM practice but fails due to misconceptions.

As can be seen in Table 16, situations of conception were not found in the empirical results. This non-existence of conceptional CC situations implies that there were no irrational value-based actions in Delta’s SDM use. The conception stereotype was the only CC stereotypes that was not found during Delta’s use of ASDMs, and therefore becomes a prominent pattern in the results based on its non-existence.
Chapter 8. Identified SDM Non-Cargo Cult

This chapter discusses the empirical findings regarding the identified SDM non-CC situations in Delta’s DSMs. The purpose of this part of the analysis is to show that the SDMCCT is applicable to successful SDM use as well, although it is not its main focus. Consequently, the analysis of non-CC situations has not been done on the CI, SPD, and VN practices at Delta.

In Publication III, the first results of non-CC situations were presented from one of Delta’s SD teams. This chapter extends these initial results by analyzing Delta’s three SD teams’ use of DSM, and the analysis has been done using the final version of the SDM CC framework.

8.1. Identified non-cargo cult behavior

From Havstorm et al. (2023), we ascertained three goals with Delta’s DSM from its scrum description, which constitute the reference point for the analysis. The following goals were ascertained: (1) to communicate their daily status, progress, and plans, (2) to identify obstacles so that the team can remove them, and (3) to set direction and focus.

The findings presented in Table 17 show classified non-CC i.e., recurrent actions were coded as being in line with the reference point. In the analysis, it was possible to identify five categories of non-deviations from the coded recurrent actions.

<table>
<thead>
<tr>
<th>Non-deviation categories</th>
<th>Non-deviation no.</th>
<th>Non-cargo cult stereotype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus on upcoming tasks</td>
<td>ND1</td>
<td>X</td>
</tr>
<tr>
<td>Identify hindrances</td>
<td>ND2</td>
<td>X</td>
</tr>
<tr>
<td>Executing meetings</td>
<td>ND3</td>
<td>X</td>
</tr>
<tr>
<td>Set directions</td>
<td>ND4</td>
<td>X</td>
</tr>
<tr>
<td>Providing status report</td>
<td>ND5, ND6</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 17. Identified SDM goal fulfillments
The non-deviation categories are: 1) Focus on upcoming tasks, 2) identify hindrances, 3) executing meetings, 4) set directions, and 5) providing status report. Each of these categories are found in Table 17 where the non-deviation categories are presented in the left column, followed by a non-deviation number, thereafter the non-CC analysis. The non-CC analysis is structured using the three non-CC stereotypes.

8.1.1. Focus on upcoming tasks
During Delta’s DSMs, I observed how the team members were focused on answering the question “What to do next?”, which meant telling the other team members about the upcoming tasks. The identified non-deviation belongs to the stereotype *accomplishment*.

The non-deviation ND1 captures situations where the intended goal and underlying reasoning was to communicate the plans for the upcoming 24 hours. Delta’s SD team members tended to provide reports on their upcoming tasks to the rest of the SD team by telling which task they planned to work on. This is illustrated by the following quote: “I will continue the day by finishing the film (the demo) that will be presented to US” – Team member 2.

Thus, these actions adhere to the goals and values of the SD teams’ SDM reference point. The ND1 situations were identified as *means-end* actions. The underlying motivation was identified as *internalized regulated*. The SD teams was focused on self-worth contingent on performance. In other words, the SD team members were motivated to do so because they also wanted to provide the information for their own self-worth, by communicating their upcoming tasks.

Situations were also identified where the SD team members had no clear plans for what to focus on. In those situations, the SD team members did communicate that they were going to choose a new task from the backlog after the DSM. By communicating intention to select a new task, they still adhered to the goal of DSM. “I will look for another one [task] to start with during the day” – Team member 7.

The SD team members were in these situations in a high level of autonomous motivation as they had higher self-determination to choose their own coming task and provide the SD team with accurate information about what the focus would be for the day.

8.1.2. Identify hindrances
During Delta’s DSMs, it was possible to observe that the SD teams were focused on reporting unforeseen or faced hindrances. Communicating hindrances meant telling the other team members about faced challenges that could affect the project’s pace. The identified non-deviation belongs to the stereotype *accomplishment*.
The non-deviation ND2 captures situations where the identified hindrances were shared with the rest of the team and, in some cases, also explains to what extent the hindrance would affect the sprint. “I upgraded to Windows 10 and ended up with the same problem as ... did before. I just want to indicate that this will probably take the whole day to fix” – Team member 3.

The underlying reasoning behind these actions was to inform the rest of the team if there are any current hindrances in their ongoing sprint – in the case above, it was a software-related problem. The actions related to ND3 were identified as *rational means-end* actions having the motivational character of *internalized regulated* where self-worth was the motivation for the team members to provide the team with this information.

### 8.1.3. Executing meetings
At Delta’s DSMs, it was possible to observe that the SD teams occasionally held the DSMs even when the SM was absent. This category of non-deviations *executing meetings* consists of the non-deviation ND3. The identified non-deviation belongs to the stereotype *accomplishment*.

ND3 captures situations where the SD teams carried out the DSMs without their SM. The underlying reasoning for these occasions were identified as rational means-end, as the SD teams felt the need of having the DSM to keep all the team members updated with the project status. These *rational means-end* actions were also identified as having *internalized regulated* motivation, as the motivational aspects behind these actions were related to coherence between goals, values, and regulations.

The coherence between the goals, values, and regulation of the DSM shows that the SD teams on these occasions had autonomous motivation as they had high self-determination.

### 8.1.4. Set directions
During Delta’s DSMs, it was possible to observe that the SD teams were, besides answering to the three main questions, also focused on reporting new directions that they would undertake based on knowledge of their project tasks. This category includes two non-deviations belonging to two stereotypes: (ND4) *achievement* and (ND5) *accomplishment*.

The non-deviation ND4 captures situations where SD team members reported on new directions of their tasks and solutions based on the knowledge gained from their research and development work. “Based on these changes in the development environment, I will work on finding a solution so that everyone can work with the same version” – Team member 4.

Since the underlying reasoning behind these actions was to provide important information to the other team members, these situations were
identified as *rational means-end* actions having *external regulated* motivation. The motivation to provide this information to the SD team was that not sharing it would lead to consequences, as the information could be important for the others to know. Such information often tended to affect other SD team members’ tasks as well.

The non-deviation ND5 captures situations where the SMs identified a need in their SD teams’ DSMs to discuss a topic further. For example, it could result in holding a strategy meeting afterwards and discussing the latest information on either new requirements or priorities of the sprint. “We will have a strategy meeting for the team directly after this morning break. The focus will be on how to move forward on the current sprint” – Scrum master.

The underlying reasoning behind these actions was *rational means-end* having *internalized motivation*. The purpose of the actions was to focus on the goal of the DSM during the current meeting; this was achieved by advertising that the strategy would be discussed at a later meeting. By doing so, the DSM stayed focused on the SD team members’ reports.

### 8.1.5. Provide status report

During Delta’s DSMs, it was possible to observe that the SD teams were focused on answering the question: “What they currently working on?” These non-deviations belong to the *accomplishment* stereotype.

The non-deviation ND6 captures situations where SD team members reported to the rest of the SD team on their current tasks that either they have finished or are currently working on. “I have been focused on the network problems that I still haven’t solve, I will continue with that today” – Team member 16.

Another team member said, “You really get the feeling that by contributing, you’re actually influencing a lot, how it goes” – Team member 15.

Since the underlying reasoning behind these actions was *rational means-end*, focusing on the importance of the goals of the DSM, these situations were identified as having *internalized regulated* motivation.
PART V – CONCLUSIONS
Chapter 9. Knowledge Contributions and Implications

This dissertation contributes with knowledge to both research and practice in the field of SD. This chapter aims to discuss the knowledge created in this dissertation and its implications for both the SDM research community and the SD domain community. The chapter will discuss the following areas as contributions:

- Understanding of the reasoning behind SDM deviations
- SDM CC theory, consisting of:
  - the SDM CC definition.
  - the SDM CC framework, including stereotypes
- The analytical method to use together with the SDM CC framework.
- Identified CC situations – catalog with deviations
- Identified non-CC.

9.1. Implications for the SDM research community

Complement current studies on reasoning of SDM deviation occurrence

According to Gregory et al. (2016) the research community seeks more empirical studies on SDM adoption and use. In this dissertation, the SDMCCCT is a theoretical contribution that builds on empirical work, providing results from a longitudinal study. It contains descriptions of what, how, and why SD teams act in a certain way in their use of SDMs. Scholars have mostly focused on to what extent SDMs are claimed to be used (Paivaärinta et al., 2010) and situational adaption of SDMs (Avison et al., 1992; Fitzgerald et al., 2002; Karlsson & Hedström, 2008).

In the structured literature review (Havstorm & Karlsson, 2023), this dissertation provides an overview of the status of the existing research in SDM adoption and use. The literature review addresses the reasoning behind adoption and use of SDMs in empirical studies, finding that there is a lack of research focus on reasons why SDM deviations occur.

Developed theory

This dissertation foremost contributes with a new theory, the SDMCCCT, to the field of SD, and it can be used to explain flawed SDM use. In the existing research, it has become evident that SD organizations have difficulties with
their use of SDMs (Conboy & Carroll, 2019; Dikert et al., 2016; Soares et al., 2022). Existing research has studied SDM use by investigating situational SDM use (Karlsson & Ågerfalk, 2004), challenges (Gregory et al., 2016), and deviations from the intended SDMs (Mortada et al., 2020). These studies investigated SDM use, but did not investigate software developers’ reasoning behind the situational use; the reasoning behind these occurrences. SDMCCT contributes to the existing literature as one explanation for the reasons behind the faced challenges in SDM use, by providing an understanding of the patterns and dynamics regarding ASDM deviations (Eloranta et al., 2015; Heikkilä et al., 2013; Mortada et al., 2020; Stray et al., 2016).

This dissertation provides a definition of SDM CC behavior that can be used when referring to identified SDM deviations and sorting them as SDM CC. SDM deviations identified as SDM CC means that an SD team face flawed use of their SDM practices due to their irrational reasoning or lack of reasoning behind their actions. Klein and Hirschheim (1991) built on Weber’s (1947) concept of rationality and defined SDM as a set of assumptions, beliefs, and recourses. Thus, they identified the rationality concept in SDM as being the key challenge for successful SD. In this dissertation, the view of rationality taken from Klein and Hirschheim (1991) has been used in the theorizing of SDM CC. In combination with the CC metaphor, SAT was used to capture different types of social actions i.e., what the SD teams are doing and the reasons behind these actions.

Existing literature provides frameworks capturing SDM use; one example is Fitzgerald et al. (2002) framework method-in-action. They put forth a framework explaining the use of SDMs and that the actions carried out are contextual. This means that the formalized SDM, roles of the SDM, the development context, and the software developers will affect the “method-in-action”. The SDMCCT emphasizes the organization’s own reference point, to account for the intended goal and use of SDMs as contextualized. Still, the SDMCCT complements the method-in-action framework by taking the analysis further in regard to why i.e., the reasons behind the actions, and acknowledging that not all executed actions might be in line with the goals and values of the SDM. SDMCCT contributes with a framework: the SDM CC framework (see Table 15 in section 6.4), to be used as an analytical framework for identifying, analyzing, and explaining the occurrences of SDM deviations.

The SDM CC framework has been refined and validated throughout four iterations. It contains theoretical stereotypes that contribute to the understanding of why SDM deviations can occur from a CC perspective, such as why anti-patterns (Eloranta et al., 2015) and similar SDM deviations (Ahmadzai & Bakhsh, 2022) occur. Previous research has studied SDM deviations in SDM adoption and use for several years. In Havstorm and
Karlsson (2023) it was clear that the underlying reasoning behind SD teams’ use and adoption of SDMs has been given limited attention. The SDMCCCT can provide an understanding of how SDMs being used and reasons behind the software developers’ impact in method-in-action.

Method
This dissertation further contributes with a detailed longitudinal empirical study, focused on reasoning behind SD teams’ use of ASDM. A similar empirical study can be found in existing research; for example, Mortada et al. (2020) carried out a case study consisting of online questions, four days up to three weeks of observations, and complementary interviews. The research method from this dissertation can be used by researchers who want to carry out similar studies of SDM use using the SDMCCCT and its framework. The research method provides guidance on how to collect data and execute the analysis to identify if an SD team faces SDM CC situations. In particular, the research method provides analytical steps for how to use the SDM CC framework (see section 5.2.3.2).

The procedure used in this study is rather time consuming, complex, and requires access to rich data. Thus, these three aspects could become challenging, in the sense that the data collection and analysis need to be well-integrated. The reason for having an integrated data collection and analysis is that the identified deviation categories should serve as the basis for more focused interviews, or data collection focusing on the reasons for the pursued and recurrent actions.

Further, the SDMCCCT draws on WMT to capture the reasons behind what the SD teams are doing i.e., to understand why actions take place. Gagné and Deci (2005) put forth the use of Ryan and Deci (2000) SDT in a work environment setting such as organizations where the study unit would be a collective and not individuals. This dissertation contributes to the use of Gagné and Deci (2005) WMT by applying the theory in a longitudinal empirical study.

Identified SDM CC
This thesis also contributes to the literature on SDM adoption and use by providing empirical results on software developers’ reasoning behind their use of SDM. This dissertation identified 36 CC situations across four ASDM practices (DSMs, CI, SPD, VN). The prominent patterns of the SDM deviations and SDM CC situations are found in Table 14 in Chapter 7. Most of the identified CC-situations are fragmental and conventional. This shows that the SD teams in Delta lacked understanding of the employed practices, and that the SD teams stuck to old practices.

Furthermore, in this study, no CC situations were classified as conception. This non-existence is prominent, making it relevant to ask why that is
the case. First, to be able to identify social actions in relation to values, the intended values from the reference point need to be in focus. Also important is having close contact with the SD team members to capture such actions by receiving answers in the moment on why they act in a certain way. This means that the conception stereotype is still unknown and yet unexplored as we have not reached an understanding of these types of situations. Second, this study did not focus on decision making. Therefore, this is one possible explanation for why the analyzed actions did not capture the conception stereotype. If Delta’s management meetings were analyzed, it might be possible to identify value-based actions that could have the character of the CC classification of conception.

This dissertation has studied four ASDM practices. Of course, existing research has studied SDM deviations by focusing on the ASDM practices, such as DSM (Stray et al., 2016) and sprint demo (Mortada et al., 2020). Stray et al. (2016) studied how 12 SD teams conducted DSMs, and they discovered that SD teams deviated from scrum in DSMs. They identified guidelines for future DSMs, such as: focus on the future, avoid the facilitator allocating turns, strive to find the least disruptive time, and keep the meeting brief. These results showed that the DSMs tended to become too long due to what was found in this dissertation as “problem solving”. However, Stray et al. (2016) suggested SD teams not discuss what was done before, but instead focus on the future. As was identified in both studies, problem-solving discussions tended to take over the meeting time. It, could, as Stray et al. (2016) argued, be better to keep complex problem-solving discussions until after the DSM taken place. This dissertation complements with six other identified deviations in relation to DSMs. Stray et al. (2016) did not investigate the reasoning behind their identified deviations, although they focused on the SD teams’ attitudes towards the DSMs.

A similar study by Mortada et al. (2020) investigated why SD teams deviated from scrum. As a result, they identified 13 deviations from scrum practices: DSM, sprint planning, and SPD. This dissertation confirms previous deviations; for example, the CC situation of “not participating in the demo” is consistent with the finding “demonstration to the wrong audience” in Mortada et al. (2020). Another deviation found in SPDs by Mortada et al. (2020) was “sprint does not end with a demonstration”, and this can be related to the CC situation “sprint demo on irregular basis”. This dissertation complements with four identified deviations in the use of SPD. Mortada et al. (2020) did not investigate the reasoning behind these deviations from a social action perspective. However, they did identify reasons for the deviations, categorized as: human factors, organizational structure, and work complexity. For example, one of the human factors were classified as “habit and acceptance”.

102 TANJA ELINA HAVSTORM Cargo Cult in Agile Software Development
Notably, the VN and CI practices addressed in this dissertation have received less attention in previous qualitative studies. Thus, the list of SDM deviations provided in this dissertation can be seen as an extended catalogue of SDM deviations and can be used as a reference in future studies.

**Identified non-CC**

In addition to the identified CC situations, this dissertation contributes with the found non-CC situations i.e., studies on rational and successful SDM use, complementing existing studies (Heikkilä et al., 2013; Häggmark & Ågerfalk, 2006). This shows that the framework can be used in identifying effective use of SDMs while studying the reasons for SDM deviations. This dissertation provides insights to better tailor SDMs and it raises awareness of the SD teams’ use of SDMs. In the studies on SDM deviations, to achieve effective or non-effective use, it has been argued that SD organizations would benefit from tailoring their use of the chosen SDM (Conboy & Fitzgerald, 2010). By tailoring their use of SDMs, it is more likely that they would meet their goals and, further, not end up in CC situations. To this end, the SDMCCT and its framework can be used when identifying CC phenomenon both before and after tailoring.

**Applicability of CC phenomenon outside SDM**

Finally, this study took place in the context of SD where SDMs are used. However, rational, irrational, and non-rational use of methods or similar artefacts do not only take place in SD. Methods are used in other domain practices. Consequently, it is reasonable to expect the CC phenomena exist in other domain practices as well. The SDMCCT and, in particular, the SDM CC framework might therefore be applicable in other contexts outside the SDM field where collective behavior and the reasons behind actions might be of interest to study.

**9.2. Implications for SD domain community**

This dissertation has several implications for the SD domain community i.e., practice. As discussed at the beginning of this dissertation, SD is negatively affected by not meeting the objectives of the contextualized SDM, which can lead to failed sprints and, in the worst cases, even failed SD projects (Gregory et al., 2016) as the SD teams have failed to learn the SDM (McAvoy & Butler, 2009). This dissertation provides a possible explanation for why such failure occurs, and how an SD organization could use the results from this dissertation to get an understanding of SD actions that can lead to misinterpretations and misconceptions.

The SDMCCT and its proposed definition of SDM CC behavior and framework do not only provide research implications. The SDMCCT contributes as an in-depth characterization and description of what the IT
industry, up until now, has described using the buzzword “cargo cult” (Shore, 2008; Soria, 2012). This dissertation provides detailed insight into the CC phenomenon in the use of SDMs. Practitioners can use the definition of SDM CC behavior when referring to imitations of other or non-functioning SD processes.

This dissertation presents and contributes with a catalogue of 36 SDM CC situations that can serve as background knowledge when an organization studies itself, aiming to find keys to improve its use of ASDMs. The patterns found in Table 16 show that the two most evident CC situations an organization should look out for are the fragmental and conventional. Consequently, to identify such situations, organizations should search for lack of understanding of their claimed practices and for situations when software developers stick with old practice but claim to adhere to new ASDM practices.

The stereotypes in the SDM CC framework can be used by practitioners to identify if their organization faces SDM CC phenomenon. The framework could be used when SD teams reflect upon an inefficient SD process, when software developers express frustration about not achieving the intended goals, when IT projects fails, or as a structured part of closing SD projects. The stereotypes can also be used by, for example, agile coaches or business developers who are working with improvements of SD processes to identify where the improvements are needed.

An SDM CC analysis could be used as an SDM use maturity assessment. It contains information about in which practices an SD team experiences irrational or non-rational reasoning in its SDM use. It implies that such results can help to steer modifications towards desired outcomes. For example, if an SD team faces CC situations in its DSMs, by having the CC analysis at hand, an agile coach can see what deviations are made, and the reasons why. This analysis will help in prioritizing what to change, and after changes are made it will be possible to analyze if the changes had a positive effect, i.e., SD team does not face CC situations any longer. An SDM CC analysis on several SDM practices could also provide insight into how mature the SD teams are in their SDM use. Possibly such analysis would show that the SD teams face less CC situations in some SDM practices than others. With such results, it would be possible to see if SD teams act based on rational understanding of the SDMs.
Chapter 10. Conclusions and Final Remarks

The aim of this dissertation was “to develop a theory, called SDM CC Theory (SDMCCT), to identify, analyze, and describe the presence of the CC phenomenon in ASDM use”. To achieve this end, this dissertation uses the “cargo cult metaphor as a starting point to explain the phenomenon of flawed ASDM use. This aim was accompanied with the overall research question: “How can the cargo cult metaphor be used to identify, analyze, and describe challenges with SDM use?”, and followed by three sub-questions:

- SubRQ1: To what extent has the reasoning behind the adoption and use of SDMs been investigated in prior research?
- SubRQ2: How can cargo cult be theorized to understand challenges in SDM use?
- SubRQ3: What analytical steps should be undertaken to uncover cargo cult in SDM use?

In this chapter, I revisit the overall research question and the sub-questions in the sub-sections. I will shed light on how these questions have been addressed by the different parts of the dissertation and what the results are.

10.1. How can the cargo cult metaphor be used to identify, analyze, and describe challenges with SDM use?

The metaphor of the CC phenomenon can be used as a starting point to identify, analyze, and describe situations, showing why SDM deviations and non-deviations occur. Using the CC metaphor as a starting point, I have adopted and combined social action theory and work motivation theory to develop a theory called SDM CC theory (SDMCCT). The theory consists of a definition of SDM CC behavior that can be used to describe challenges with SDM use, and a framework that can be used to identify and analyze challenges with SDM use. Thus, the CC metaphor provides a ground to problematize flawed SDM use by introducing SDM CC behavior to the SDM field as a new perspective.

The prominent patterns of the empirical findings show that among the 36 identified CC situations the ones appearing most were the conventional and fragmental stereotypes. These findings indicate that future research and practice shall be aware of the high potential of identifying SDM use situations where old traditions and habits are being retained after SDM
adoption, and where the lack of understanding of the SDMs results in mal-practices. In the latter CC situations, it was found that SD teams deviated from their SDM due to limited understanding of the chosen SDMs and a lack of underlying reasoning behind their action.

10.1.1. To what extent has reasoning behind adoption and use of SDMs been investigated in prior research?
To place my research on SDM CC in a larger context, this dissertation presented a structured literature review. The goal of this was to gain an understanding of the ongoing discussion on SDM challenges in the SDM field and to what extent the existing research has paid attention to software developers’ reasoning behind SDM adoption and use.

The surprising finding shows that even though SDM research has a long history, and that the choice to adopt and use SDMs is known to be based on software developers’ situational knowledge of SDMs, existing research on adoption and use has predominantly focused on rational activities. The results of the literature review, Publication I in this dissertation, show that existing empirical research on this topic of SDM adoption and use provide limited knowledge on how to identify irrational and non-rational adoption and use activities.

10.1.2. How can cargo cult be theorized to understand challenges in SDM use?
The CC phenomenon can help to understand how and why an SD team fails or succeed in its SDM use. The SD teams carry out activities in a certain way, and these SDMs include goals and values that SD teams set out to achieve and adhere to. The CC phenomenon can provide insight into SD teams’ reasoning behind their activities, which means that if the SD teams encounter challenges in their SDM use, the SDMCCCT can help to identify irrational reasoning or lack of reasoning behind their actions. The SDMCCCT will also provide understanding of what type of motivation the SD teams had and that led them to the faced challenges.

In this dissertation, the CC phenomenon has been defined and placed in the context of SD and in particular the field of SDM. The definition is presented at the end of Chapter 6 (see section 6.4.1). The phenomenon was theorized and resulted in a theory, SDMCCCT, for identifying, analyzing, and explaining why challenges, such as deviations in SDM use, occur.

As well as a definition, the SDMCCCT provides a framework consisting of a set of eight stereotypes describing three non-CC situations, and five CC situations. The stereotypes are possible to identify in empirical cases of SDM use. The SDM CC framework can be found in publications II-V as well as in Table 15 in Chapter 6.
10.1.3. What analytical steps should be undertaken to uncover cargo cult in SDM use?

To uncover CC in SDM use, this dissertation presents a set of four analytical steps. The presented analytical steps guide future users of the SDM CC framework on how to analyze collected data. The analytical steps can be found in Publication V as well as in Chapter 5 (see section 5.2.3.2). Such analysis requires a rich dataset consisting of data: a) about actions carried out by a collective, and b) about the reasons behind these actions. This data is needed to have the possibility to grasp what is going on in an SD team, and the reasons why they carry out recurrent actions in a certain way, i.e., to grasp the motive behind the recurrent actions.

Furthermore, to understand how the actions are carried out, as well as why, a good understanding of the studied organization is needed, including its previous way of working, and its intended goals with the SDMs. The reason why this knowledge is required is that it is only by interpretations that it is possible to relate the stereotypes to the identified recurrent actions. Furthermore, in order to make interpretations related to the conventional stereotype, an understanding of previous SDMs used in the organization is needed.

10.2. Limitations

First, there is a validation limitation with the SDMCCT. The SDMCCT was validated in four iterations with additional sets of empirical data on ASDM use. However, each iteration in the theorization ended with lessons learned that were used and applied for the subsequent iterations for refinements of the SDMCCT.

The SDMCCT is based on a conceptualization of SDMs that includes both ASDMs and traditional SDM. Thus, the SDMCCT should be possible to use for describing, analyzing, and identifying SDM CC behavior with other SDMs than ASDMs. However, this has yet to be validated in future research.

Second, SDMCCT was theorized based on empirical data from one organization; thus, I do not claim that the results are transferable beyond this context. Still, a detailed description of the study has been provided with many details to make it possible for the reader to assess the transferability to other contexts.

Third, that said, studying and uncovering such CC phenomena using an ethnographic approach was essential. Furthermore, the ethnographic approach gave the possibility to observe the activities carried out daily by the SD teams and to understand the reasons why they occurred.
As said in Chapter 5, the data collection was conducted by one researcher onsite for three years. To avoid observation bias during data collection, both observation protocols and interview guides were used to record everything that happened and to keep the semi-structured interviews focused. Avoiding observational bias also applied to informal meetings with team members from both Delta and Beta, where the focus was to keep understanding the SD teams actions.

Fourth, the SDMCCT is the result of an abductive research that included theory searching. In this study, this search resulted in using SAT and WMT as the foundation for the SDMCCT; other way of searching for theories could result in other theories that could be combined to capture the CC phenomena, which might result in a different set of stereotypes. Thus, a different set of stereotypes would also result in another catalogue of SDM CC situations, or at least different characterization.

Fifth, as said in Chapter 4, 25 ASDM practices were identified at Delta. However, four of these ASDM practices were analyzed using the SDM CC framework. This implies that SD actions carried out by Delta’s SD teams when using their ASDM practices have yet to be analyzed to uncover additional CC situations.

10.3. Future research
Moving beyond the limitations of this dissertation opens possible ways forward and suggestions for future research.

As mentioned in the limitations, SDMCCT could be validated further. In future research, there is a possibility to apply the SDMCCT to another context. Such validation would make it possible to see to what extent the theory helps improve SDM use and evaluate the long-term effects of SD organizations using the results of the SDM CC framework for improving their SDM use. This would evaluate the true value of the framework, as the intention is to improve SD. To reach such conclusions, a longitudinal study is required using the SDM CC framework to identify CC situations, and provide feedback i.e., results back to the SD teams, and follow up the SD teams’ SDM use after the feedback.

As mentioned in the limitations, the SDMCCT was validated with a case of software developers’ use of ASDM practices. The theory would benefit from validation, both in additional SDM settings and other types of organizations. An interesting setting would be to validate the framework on traditional SDMs, but also to apply and test the framework in other areas than
SD, such as business organizations where business agility has become widespread, or in other fields where new methods are being adopted and used. Other application areas could be use of information security risk management methods.

There is also a possibility to delve deeper into WMT and its psychological aspects when studying SDM use. As this study focused on a higher level of motivation i.e., amotivation and external regulated (extrinsic) motivation in work situations, a deeper analysis of how the specific psychological aspects affect a collective behavior in CC situations would be interesting. Having said that, the SDMCCCT and the current version of the SDM CC framework can be used as a starting point for developing the theory further.
References

https://ec.europa.eu/research/participants/data/ref/h2020/other/hi/et-
hics-guide-ethnog-anthrop_en.pdf
https://www.agilebusiness.org/static/e3b17daa-4776-4666-
909f2c297c8469ce/State-of-Agile-Culture-Report-2021-22.pdf
Method Usage: Issues and Challenges in Afghanistan. In AI and IoT 
for Sustainable Development in Emerging Countries: Challenges and 
Opportunities (pp. 255-264). Springer International Publishing.
systems development methodology for different situations International 
Conference on Enterprise Information Systems, Funchal, Madeira, 
Portugal.
methodologies for information systems development. Journal of 
Information Technology, 7, 127-140.
methodologies? Association for Computing Machinery. 
Communications of the ACM, 46(1), 78.
Bajc, V. (2012). Abductive Ethnography of Practice in Highly Uncertain 
Conditions. The ANNALS of the American Academy of Political and 
Social Science, 642(1), 72-85. 
https://doi.org/10.1177/0002716212438197
Addison-Wesley.
Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., 
Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., 
Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & 
Boehm, B. W. (1988). A spiral model of software development and 
Braa, K., & Vidgen, R. (1999). Interpretation, intervention, and reduction 
in the organizational laboratory: a framework for in-context 
information system research. Accounting, Management and 
Information Technologies, 9(1), 25-47.


TANJA ELINA HAVSTORM Cargo Cult in Agile Software Development 113


Hazzan, O., & Dubinsky, Y. (2014). The Agile Manifesto. In O. Hazzan & Y. Dubinsky (Eds.), *Agile Anywhere: Essays on Agile Projects and Beyond* (pp. 9-14). Springer International Publishing. [https://doi.org/10.1007/978-3-319-10157-6_3](https://doi.org/10.1007/978-3-319-10157-6_3)


Häggmark, M., & Ågerfalk, P. J. (2006). Why software engineers do not keep to the principle of separating business logic from display: A method rationale analysis. 18th International Conference, CAISE 2006,


PART VI – ADDITIONAL MATERIAL
Appendix A – Consent form for data collection

Consent for data collection regarding research project on Cargo cult behavior in agile systems development

Due to the Swedish regulations, the General Data Protection Regulation (GDPR), Örebro University is required to inform you about the research project that takes place at your work site. It is also required that the university collect consent from you as an employee of [insert company name] about your participation in the research project. By collecting data on different elements of information systems development methods and related tools that are used in the daily work, and the reasons behind this use, the university will conduct research with the aim to improve the understanding of challenges in agile systems development method adoption. This research focuses on challenges in systems development method use on a group and organizational level. Previous research has shown challenges related to this type of adoption and an increase understanding is therefore highly important.

The research project takes place during 2018-2022 and is conducted by the researcher Tanja Mäki-Russas. The researchers will conduct a so-called ethnographical study, consisting of observations, interviews and document studies. The collected data will be processed during 2018-2022 after which the data will be archived. The data will be stored on the university’s storage area to ensure the data is processed according to the university’s information security policy.

You can find out what has been registered about you or have feedback on the processing or data collected by contacting Tanja Mäki-Russas (tanja.maki-russas@oru.se), Fredrik Karlsson (fredrik.karlsson@oru.se) or the university’s Data Protection Officer at dataskydd@oru.se. Complaints that cannot be resolved with Örebro University may be submitted to The Swedish Data Protection Authority.

I have read the information provided above and consent to the research project’s data collection.

I have read the information provided above and do not consent to the research project’s data collection.

Firstname/Surname: ________________________________

Role (project leader/forum master/team member): ________________________________

Date ______ / ______ 20________ Signature: ________________________________

Project responsible: Fredrik Karlsson, fredrik.karlsson@oru.se, +46(0) 19 30 39 94
Researcher: Tanja Mäki-Russas, tanja.maki-russas@oru.se, +46(0) 73 56 46 891
Appendix B – Interview Guide

This appendix contains the interview guide, where the interview questions can be found as well as an explanation of why the questions were asked. This interview guide was used in the semi-structured interviews.

<table>
<thead>
<tr>
<th>2020-10-26 Interview guide</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>Interview questions</td>
<td>Explanation of why the questions were asked</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ 1 ] Introduction/Background - Introductory Questions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Could you start by briefly describing your role as a team member (developer/tester) and your area of focus in the development projects?</td>
<td>To get a picture of the respondent’s background and membership in the team.</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>Do you have any specific tasks inside or outside the team?</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>How long have you worked here at Alpha and on this team?</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>Have you been part of any other team before?</td>
</tr>
<tr>
<td>1.</td>
<td>What development methods (and practices) would you say you use in your team?</td>
<td>Find out the awareness of the use of methods; does the respondent have insight into the methods and practices used?</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>How long have these been used (by the team and by you)?</td>
</tr>
<tr>
<td>2.</td>
<td>Have you received any introduction/training on these methods? (Yes or no)</td>
<td>Captures whether the respondent received/not received their methodological understanding from the training provided.</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>If yes</td>
</tr>
<tr>
<td></td>
<td>A1</td>
<td>When and where did it take place?</td>
</tr>
</tbody>
</table>
## A2
Describe it briefly, e.g., how long it was, the form of the course, whether it was theoretical/practical, etc.

**To find out more information about the training**

### B
If no

Would you find it useful? What would you like to learn more about?

**To find out if the respondent sees a need for increased methodological understanding.**

### B1

<table>
<thead>
<tr>
<th>2 ] SDM Practices</th>
</tr>
</thead>
</table>

**Questions directly linked to the observed practices used by the teams**

### [2.1] Daily scrum meetings

<table>
<thead>
<tr>
<th>What purpose would you say your team has for the daily scrum meetings?</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do you feel the team is approaching achieving that purpose?</td>
</tr>
<tr>
<td>A</td>
</tr>
</tbody>
</table>

**Is the team far from or actively working close to these goals (reference point)?**

**The respondent should start from themselves when looking at their team and reflect on whether the team lives up to the determined reference point.**

<table>
<thead>
<tr>
<th>Do you feel that the team is living up to the goal(s) (yes or no)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1 If yes, how do you do it? To see if there are concrete actions that lead the team to the achievement of the goals.</td>
</tr>
<tr>
<td>B2 If no, what are the reasons why you are not achieving the objectives? To see if there are concrete actions that prevent the team from reaching the objectives.</td>
</tr>
</tbody>
</table>

1. **Would you say this is important? (yes or no)**

A If yes, why?

B If no, why not?

2. **What motivates you to achieve the goal?**

**Investigates whether there is something concrete that motivates the respondent to achieve the goal (own goals or values/external goals or values/opportunity to develop, etc.).**

<table>
<thead>
<tr>
<th>Describe the level of motivation you experience with having daily scrum meetings using a scale of 1-10. (1. Not so motivating vs. 10. Very motivating)</th>
</tr>
</thead>
</table>

3. **For whom do these daily scrum meetings have value?**

**Is it something that is only for the team member, manager, scrum master, organization, project, or no one?**

A Why?
<table>
<thead>
<tr>
<th>2.2 Sprints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
</tr>
<tr>
<td>2.</td>
</tr>
<tr>
<td>3.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>A1</td>
</tr>
<tr>
<td>A2</td>
</tr>
<tr>
<td>A3</td>
</tr>
<tr>
<td>A3a</td>
</tr>
<tr>
<td>A3b</td>
</tr>
<tr>
<td>A4</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>B1</td>
</tr>
<tr>
<td>B2</td>
</tr>
<tr>
<td>B3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>5.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>6.</td>
</tr>
</tbody>
</table>
### A
If you **undertake** a task, how often would you say this happens?

To know to what extent the respondent is given the opportunity to undertake tasks.

### B
If you are **assigned**, who assigns this to you?

To know where the external control comes from, the team/scrum master/manager/organization.

### B1
Do you have the possibility to switch tasks?

### B2
How often do you have the opportunity to change tasks?

### How do you feel that new requirements affect you?

Find out if there is any malpractice during the sprints.
Find out how often new requirements come in during a sprint. If

**How often do new requirements come in during a sprint?**

they are to follow the scrum method and new requirements come in during an ongoing sprint, this is a sign of malpractice of the method. On the other hand, if they are working with the kanban method, it is common for new requirements to come in during an iteration and this is not a malpractice.

### How could new incoming requirements arriving midway through a sprint affect the ongoing work?

Find out if there is any malpractice during the sprints.

### 8. A
How often do new requirements come in during a sprint?

### 9. A
Why would that be the case?

### 10. A
Why is that important?

### When the team is approaching a new release to the customer, do you find that you follow the same way of working as before the release is approaching, or does the way of working change slightly? Describe briefly.

Linking observed events to the perceived degree of control

**During these times, do you feel that you have more or less control over what should be done? If so, why/why not?**

To get an indication of the degree (more or less) of governance experienced by the respondent in such situations.

### 11. B
What do you think this might be due to?

To find out why the structure is this way

---

**[2.3] Sprint planning meeting**

Does your team conduct sprint planning meetings? (yes or no)

Identify if the team uses this practice or a similar one that it refers to by another name.

**A**
If yes, what is the goal of your team with these meetings?

Identify if the respondent is aware of the goals (reference point) the
<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 Why do you conduct these meetings?</td>
<td>Identify if this is something that is routine, old habit, or if this is for the purpose of fulfilling specific values or goals.</td>
</tr>
<tr>
<td>A2 Is this important? (yes or no)</td>
<td>Reasons.</td>
</tr>
<tr>
<td>A2a If yes, why and for whom?</td>
<td>Identify the reason why this is perceived as unimportant.</td>
</tr>
<tr>
<td>A2b If no, why not?</td>
<td></td>
</tr>
<tr>
<td>A3 How often do you hold these meetings?</td>
<td></td>
</tr>
<tr>
<td>A4 What motivates you to contribute during sprint planning meetings?</td>
<td></td>
</tr>
<tr>
<td>A5 For whom do these sprint planning meetings have a value?</td>
<td></td>
</tr>
<tr>
<td>A5a Why?</td>
<td></td>
</tr>
<tr>
<td>A6 What value would you say these meetings add?</td>
<td></td>
</tr>
<tr>
<td>B If no, why don’t you have sprint planning meetings?</td>
<td>Find out the underlying factor of non-use of this practice.</td>
</tr>
<tr>
<td>B1 What value do you think this type of meeting before a new sprint would add?</td>
<td></td>
</tr>
<tr>
<td>B2 How would these meetings motivate you for an upcoming sprint?</td>
<td></td>
</tr>
<tr>
<td>Describe the level of motivation you experience with having sprint planning meetings on a scale of 1-10. (1. Not so motivating vs. 10. Very motivating)</td>
<td></td>
</tr>
</tbody>
</table>

[2.4] Sprint retrospective meeting

<table>
<thead>
<tr>
<th>Question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does your team conduct sprint retrospective meetings? (Yes or No)</td>
<td>Identify if the team uses this practice or a similar one that it refers to in some other way.</td>
</tr>
<tr>
<td>A If yes, what is the goal of your team with these meetings?</td>
<td>Identify if the respondent is aware of the goals (reference point) the team/organization has regarding the sprint retrospective meetings.</td>
</tr>
<tr>
<td>A1 Why do you conduct these meetings?</td>
<td>Identify if this is something that is routine, habitual, or if it is for the purpose of fulfilling a specific value or goal.</td>
</tr>
<tr>
<td>A2 Is this important? (Yes or No)</td>
<td></td>
</tr>
<tr>
<td>A2a If yes, why and for whom?</td>
<td></td>
</tr>
<tr>
<td>A2b If no, why not?</td>
<td></td>
</tr>
<tr>
<td>A3 How often do you conduct these meetings?</td>
<td></td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>A4 What motivates you to contribute during sprint retrospective meetings?</td>
<td></td>
</tr>
<tr>
<td>A5 For whom do these sprint retrospective meetings have value?</td>
<td></td>
</tr>
<tr>
<td>A5a Why?</td>
<td></td>
</tr>
<tr>
<td>A6 What value would you say these meetings add?</td>
<td></td>
</tr>
<tr>
<td>B If no, why don’t you have sprint retrospective meetings?</td>
<td>Find out the underlying factor of non-use of this practice.</td>
</tr>
<tr>
<td>B1 What value do you think this type of meeting after a completed sprint would add?</td>
<td></td>
</tr>
<tr>
<td>B2 How would these meetings motivate you for a future sprint?</td>
<td></td>
</tr>
<tr>
<td><strong>2. Describe the level of motivation you experience with having sprint retrospective meetings on a scale of 1-10. (1. Not so motivating vs. 10. Very motivating)</strong></td>
<td></td>
</tr>
</tbody>
</table>

### 2.5 Sprint demo/reviews

Each team has a demo show to the management team and other US stakeholders at the end of each iteration.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Who is responsible for producing a sprint demo of the implemented solutions?</td>
<td></td>
</tr>
<tr>
<td>2. Who presents this demo?</td>
<td></td>
</tr>
<tr>
<td>3. What is the purpose of these demos?</td>
<td></td>
</tr>
<tr>
<td>4. For whom does this provide value?</td>
<td></td>
</tr>
<tr>
<td>A Why?</td>
<td></td>
</tr>
<tr>
<td>5. Do you and your team get feedback on these demos?</td>
<td></td>
</tr>
<tr>
<td>A How often does this happen?</td>
<td></td>
</tr>
<tr>
<td>6. Do you find sprint demos motivating? (yes or no)</td>
<td></td>
</tr>
<tr>
<td>A If yes, describe how this is motivating.</td>
<td></td>
</tr>
<tr>
<td>B If no, describe why this is not motivating.</td>
<td></td>
</tr>
<tr>
<td>7. Describe the level of motivation you experience with having sprint demos on a scale of 1-10. (1. Not so motivating vs. 10. Very motivating)</td>
<td></td>
</tr>
</tbody>
</table>
## [2.6] Estimation

### Of artifacts tasks and sprints

**Product backlog**

1. Who is responsible for keeping the project's product backlog up to date?
2. How is it visible to you and your team members?
3. Who is responsible for estimating each item in the backlog?
4. What is the goal of having a product backlog?
5. For whom is it important to have an estimated product backlog?

6. Do you find an estimated product backlog motivating? (yes or no)
   - A) If yes, describe how this is motivating.
   - B) If no, describe why this is not motivating.

7. Describe the level of motivation you experience with having an estimated product backlog on a scale of 1-10. *(1. Not so motivating vs. 10. Very motivating)*

**Sprint backlog**

8. Who is responsible for keeping the team’s sprint backlog?
9. How is this visible to you and your team members?
10. Who is responsible for estimating each task in the sprint?
11. What is the goal of having a sprint backlog?
12. For whom is it important to have an estimated sprint?

13. Do you find an estimated sprint backlog motivating? (yes or no)
   - A) If yes, describe how this is motivating.
   - B) If no, describe why this is not motivating.

14. Describe the level of motivation you experience with having an estimated sprint backlog on a scale of 1-10. *(1. Not so motivating vs. 10. Highly motivating)*
### Tasks

| 15. | Who is responsible for ensuring that each task is estimated? |
| 16. | Is the estimate visible to you and your team members? |
| 17. | What is the purpose of estimating a task? |
| 18. | Describe how the estimation of tasks could facilitate the achievement of the sprint goals. |
| 19. | For whom is it important to have estimated tasks? |
| 20. | Do you perceive estimated tasks as motivating? (yes or no) |
|   | A If yes, describe how this is motivating. |
|   | B If no, describe why this is not motivating. |
| 21. | Describe the degree of motivation you experience with having estimated tasks using a scale of 1-10. (1. Not so motivating vs. 10. Highly motivating) |

### Continuous Integration

**Use of FogBugz then Jira (their digital SDM tool)**

<p>| 1. | When you finish the task that you have committed/been assigned in Jira, what do you do? |
|    | Describe how you go about integrating the code. |
|    | A Describe how you go about integrating the code. |
|    | B Are there any clear guidelines on how and when to integrate the code? Please describe this briefly. |
|    | C How do you think these procedures are followed in your team? |
| 2. | How often do you integrate your solutions/code? |
| 3. | What is the purpose of integrating the code continuously? |
| 4. | For whom does this provide value? |
| 5. | Do you find continuous integration of program code motivating? (yes or no) |
|    | A If yes, describe how this is motivating. |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>If no, describe why this is not motivating.</td>
</tr>
<tr>
<td>6.</td>
<td><strong>Describe the level of motivation you experience in continuously integrating codes using a scale of 1-10. (1. Not so motivating vs. 10. Very motivating)</strong></td>
</tr>
<tr>
<td>7.</td>
<td>Who is responsible for integrating the program code according to your procedures?</td>
</tr>
</tbody>
</table>

### [2.8] Method tool

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jira is where the project's product backlog and sprint backlog are located.</strong></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>How and when would you say you use the Jira tool in your work? Describe.</td>
</tr>
<tr>
<td>2.</td>
<td>Describe why and how often you use Jira in your work.</td>
</tr>
</tbody>
</table>
| 3. | Do you feel that the tool supports or hinders you in your work? (yes or no)  
   A | If yes, describe how this supports you in your work.  
   B | If no, describe how it does not support you in your work. |
| 4. | When do you think you make the most use of the tool and why? |
| 5. | When do you think you make the least use of the tool and why? |
| 6. | Do you feel that you and your team follow any specific method when using the tool? Please describe which method you think the tool has a significant role in. |
| 7. | A | For whom is this significant? |
| 8. | For whom does this provide value? |
| 9. | Do you find the use of Jira motivating? (yes or no)  
   A | If yes, describe how this is motivating.  
   B | If no, describe why this is not motivating. |
| 10. | Describe the level of motivation you experience in using the Jira methodology tool on a scale of 1-10. (1. Not so motivating vs. 10. Very motivating) |

### [3] The team

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
### Identify individual experience in terms of self-determination and external control in the development team and its tasks.

<table>
<thead>
<tr>
<th></th>
<th>To what extent (a lot or a little/high or low) do you feel that you decide and determine how your daily work is carried out?</th>
<th>Finds out whether the person experiences intrinsic motivation or control when performing their work.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>A To what extent do you feel you can influence this?</td>
<td>To find out to what extent there is the possibility for the respondent to influence their work, and how much external control there is from the team/manager/organization.</td>
</tr>
<tr>
<td></td>
<td>Are there any situation(s) that you can relate to where you feel you have more control over the tasks you must perform?</td>
<td>To derive the perceived motivation/control to specific rationality types/social actions.</td>
</tr>
<tr>
<td></td>
<td>A Which ones?</td>
<td>To be able to relate these to specific SDM practices.</td>
</tr>
<tr>
<td></td>
<td>B How often do you feel you have such an opportunity?</td>
<td>Find out to what extent the respondent feels that they have self-determination.</td>
</tr>
<tr>
<td></td>
<td>C What makes you feel that way about [the situation]?</td>
<td>Find out which factor is behind the situation.</td>
</tr>
<tr>
<td>2.</td>
<td>Are there any situations you can relate to where you feel you have less or no control over the tasks you must perform?</td>
<td>Find out if there are specific situations where there is external control governing social actions.</td>
</tr>
<tr>
<td></td>
<td>A Which ones?</td>
<td>In order to trace these to specific SDM practices.</td>
</tr>
<tr>
<td></td>
<td>B How often do you feel that you do not have such an opportunity?</td>
<td>Find out to what extent the respondent experiences external control.</td>
</tr>
<tr>
<td></td>
<td>C What makes you feel that way about [the situation]?</td>
<td>Find out the underlying factor.</td>
</tr>
<tr>
<td>3.</td>
<td>To what extent (a lot or a little/high or low) do you feel that someone other than yourself (inside or outside the team) decides and determines how your daily work is</td>
<td>Finds out whether the person experiences intrinsic motivation or control when performing their work.</td>
</tr>
<tr>
<td></td>
<td>A To what extent do you feel you can influence this?</td>
<td>To find out to what extent there is the possibility for the respondent to influence their work, and how much external control there is from the team/manager/organization.</td>
</tr>
<tr>
<td></td>
<td>B How often would you say you can influence this?</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Are there any situation(s) you can relate to where you feel that someone else (inside or outside the team) decides/decides on the tasks you have to perform?</td>
<td>Find out if there are specific situations where there is external control governing social actions.</td>
</tr>
<tr>
<td></td>
<td>A Which ones?</td>
<td>In order to trace these to specific ISDM practices.</td>
</tr>
<tr>
<td></td>
<td>B How often do you feel this way?</td>
<td>Find out to what extent the respondent experiences external control.</td>
</tr>
<tr>
<td>C</td>
<td>What is it that makes you feel this way about [the situation]?</td>
<td>Find out the underlying factor</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
</tbody>
</table>

[ 4 ] Final open question

*Identify any procedures and structures that may underpin the actions taken.*

1. Is there anything you would like to add, or a question that wasn't asked, or do you have any questions?
Appendix C – Observation Protocol

This appendix provides an overview of the observation protocol. It contains time and date, type of activity e.g., daily scrum meeting, and if the activity was considered as formal or informal. Information such as who, what, why, and additional comments were captured and noted in this protocol.

### Observations 181001 – Monday

<table>
<thead>
<tr>
<th>Activity</th>
<th>Daily scrum meeting – Team A</th>
<th>Formal/Informal</th>
<th>F</th>
<th>Other “free text”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>SM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team member</td>
<td>What</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Goes through the new priorities that came up at Friday’s Scrum/Agile meeting with the U.S. office. Indicates problems with testing, should compare with another team member to see if they get the same or different outcomes. Can possibly depend on the versions.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Why</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tells about what has been done and what is intended to be implemented during the day, as well as who may be able to help solve the problem.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity</th>
<th>Daily scrum meeting – Team B</th>
<th>Formal/Informal</th>
<th>F</th>
<th>Other “free text”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who</td>
<td>SM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>What</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Begins the meeting and begins to go through what he did on Friday, i.e. Scrum/Agile meeting with the U.S. office. To talk about the direction the team needs to take. They should have a release within the next few days, probably Tuesday / Wednesday.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Why</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>New requirements from customers mean that the team needs to rethink current priorities and increase the time in the lab for the version to go out flawlessly.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*SM = Scrum master*
Appendix D – Organizational Documents

A summary of the business documents I had access to during this research. The table consists of what type of document, when it was provided, and how the document helped the research.

<table>
<thead>
<tr>
<th>Type of document</th>
<th>When</th>
<th>How the document helped the research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method documents</td>
<td>2018</td>
<td>These documents helped to grasp an overview of Beta and Delta’s presentation of their SDMs. To identify which software development methods they had adopted.</td>
</tr>
<tr>
<td>Current organizational structure - organizational chart</td>
<td>2018</td>
<td>This document helped me to understand the organization and the roles, to place Delta in a broader picture, and to understand the decision chain.</td>
</tr>
<tr>
<td>Strategy continuous integration - toolchain</td>
<td>2018</td>
<td>This document helped me to understand Beta’s intended goals and strategy with use of continuous integration and how their various tools, such as Jira and Git, would fit together.</td>
</tr>
<tr>
<td>Intended method description</td>
<td>2019</td>
<td>This document helped to identify and understand Delta’s reference point, i.e., its intended goals with the ASDMs.</td>
</tr>
<tr>
<td>Current organizational structure - organizational chart</td>
<td>2019</td>
<td>This document helped me to understand the overall organizational structure, to place Delta in a broader picture, and to understand the decision chain. Most importantly, this version helped me to understand the changes made in each SD team, such as new team members or team members who left.</td>
</tr>
<tr>
<td>Allocated seating at Delta</td>
<td>2019</td>
<td>This document helped me to understand where each SD team member was placed and to understand how they communicated with each other within and across teams. I was able to compare this with my observations.</td>
</tr>
<tr>
<td>Agile education agenda</td>
<td>2018</td>
<td>This document helped me to first understand what kind of knowledge and information Beta was asking the agile coach for. Also, to recall later on what knowledge from this education the SD teams at Delta actually used.</td>
</tr>
</tbody>
</table>
## Appendix E – Ethnographic Documents

A summary of the ethnographic documents created during this research. The table consists of what type of document, when it was created and used, and how the document helped the research.

<table>
<thead>
<tr>
<th>Type of document</th>
<th>When</th>
<th>How the document helped the research</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD team setup</td>
<td>2018-2020</td>
<td>I created a document consisting of all the research study participants to help memorize the context, which was updated during the study.</td>
</tr>
<tr>
<td>Observation protocol</td>
<td>2018-2020</td>
<td>The observation protocol helped me to note down the essential activities carried out by the SD teams. This documentation is the core document containing what, when, by whom, and why SD team members acted as they did.</td>
</tr>
<tr>
<td>Interview guide</td>
<td>2020</td>
<td>The interview guide helped to keep the semi-structured interviews focused on understanding the reasoning behind the SD teams’ actions. The guide was developed and built on SAT and WMT, to gain information on what, and especially why, certain ASDM practices were used in a certain way, and the motivation behind these actions.</td>
</tr>
<tr>
<td>Field notes</td>
<td>2018-2020</td>
<td>In addition to the observation protocol, activities carried out during the observations were noted down. This documentation helped in analyzing the context in Delta.</td>
</tr>
<tr>
<td>Note recordings</td>
<td>2018</td>
<td>To help myself to remember what I observed, I audio recorded myself besides taking field notes.</td>
</tr>
<tr>
<td>Transcriptions</td>
<td>2020-2021</td>
<td>The transcription of the semi-structured interviews helped in conducting data analysis, using the SDM CC framework.</td>
</tr>
</tbody>
</table>
Appendix F – Theorizing Process in Detail

This appendix includes a detailed description of the theorizing process for each of the four versions and iterations of the SDM CC theory and how the framework evolved throughout the entire process.

Initial version of the SDM CC theory
As mentioned above, the IT industry has been, for some time, reflecting on unsuccessful cases of SDM use, especially ASDMs. These unsuccessful cases have been mainly about situations where practitioners have identified trend followers and used CC as a buzzword for failed SDM use.

As the ASDMs have been the most used SDMs in the industry, it has also been viewed as the trend. However, challenges have arisen in succeeding with ASDM adoption and use. The “cargo cult” buzzword has been used to describe a phenomenon of “doing but not being agile” as a problem. In (Havstorm & Karlsson, 2023) we found a lack of empirical research focusing on the reasoning behind SDM adoption and use, and the lack of, or flawed, reasoning in relation to SDM closely relates to why SDM CC situations occur.

The research community also stresses that few empirical studies focus on challenges with ASDM adoption and use (Gregory et al., 2016). As a result, this dissertation set out to provide an in-depth understanding of the SDM CC phenomenon, where the end is a CC definition situated in the SDM field and an analytical framework. The framework is intended to be used for identifying CC situations in SD organizations.

My work with the SDMCCT started with the original definition of CC by Worsley (1957), presented in Chapter 2. This definition stressed the repeatable collective behavior of others where certain activities were carried out to reach the same success as others. When working with the definition, my search for theories first led me to SAT (Weber, 1947). SAT consists of stereotypes describing social actions, which, in this dissertation, represent activities from original definition of CC. SAT provided possibilities, through its stereotypes, to capture four different types of social actions based on the context. Secondly, I wanted to understand why the SDM activities were carried out, i.e., the motivation behind these actions. To study motivation behind actions I combined SAT with SDT (Ryan & Deci, 2000), to help in defining SDM CC behavior.

Further, a combination of the two theories was carried out. Four distinct CC situations were described, using the types of rationality from SAT and types of motivation from SDT. These four CC situations were also the first version of the CC stereotypes. The stereotypes were represented in the
framework as being situated along two dimensions: type of rationality and type of motivation.

To connect these situations to SDM, I used buzzwords from the IT industry that captured situations of challenges in ASDM use. For example, one buzz phrase was “jump on the bandwagon”, which represents people joining a hype without considering any reasoning for why. See Paper II for details.

In 2018, the paper was presented to the research community at an international research conference, Information Systems Development (ISD2018). Lessons learned were used as input for further elaboration on the SDMCCT. Based on the feedback I received, these were the major lessons brought into the next phase of my theorizing process: (1) the research community saw a need to add stereotypes that capture successful SDM use situations, and (2) the research community asked for empirical validation of the framework.

Second version of the SDM CC theory

To further theorize SDM CC, the framework and its stereotypes were elaborated further based on the lessons learned from the initial definition and first version of the framework.

As mentioned above, the first version of the framework did not provide possibilities to capture successful SDM use situations. Thus, extending the framework could be useful to grasp the entire picture of SD teams’ use of SDMs, although it should be stated that this extension is not driven by the SDM CC definition. With regard to the definition presented in Paper II, it did not receive much attention during the conference where it was presented, and thus remained as is. The elaboration of the framework led to adding stereotypes that represent successful SDM use, i.e., rational social actions. Furthermore, I removed the dimensions in the framework as I instead put forth six distinct stereotypes.

When elaborating on the theory and the framework, I compared the framework with empirical data to find out if it was possible to analyze the empirical data using the framework. This is an example of the constant comparison between theory and empirical data, which is central the abductive approach according to Czarniawska (2007). The empirical material (observations, interviews, and business documents) was analyzed using the elaborated framework (the details are described in section 5.5 Data analysis). Publication III put forth the first version of the analytical steps for how to use the framework. The empirical material used in this phase consisted of data on one SD team’s use of the ASDM practice of daily scrum meeting (DSM).

In 2019, the definition and the second version of the framework were presented at an international research conference, Americas Conference in
Information Systems (AMCIS2019). The paper used the framework with empirical data to uncover situations of CC behavior from one SDM practice that one SD team used. Based on the feedback I received, these were the major lessons to bring into the third phase of elaboration on the theory: (1) the empirical result in Paper III only indicated on extrinsic motivation, and (2) no affectual or value-based social actions could be detected.

**Third version of the SDM CC theory**

Based on the lessons learned from the second version of the theory, further elaboration was made on the stereotypes in the theory. As the lessons learned from the second phase stressed, intrinsic motivation is rather complicated to identify.

This lesson learned made it clear that in order to make the analysis work, the crucial applicability level when applying SDT was, i.e., what type of motivation could be captured. Additional reading on the SDT led me to understand that SDT focuses on individuals. Since I studied a collective behavior, I had to find out if it was possible to use SDT for that purpose and, in that case, how. In this third phase when I went back to the theorizing in combination with additional empirical data, I came across work motivation theory (WMT) by Gagné and Deci (2005). WMT is applicable for organizational studies on work motivation, and it builds on SDT with a purpose to open possibilities to analyze and study groups and not individuals.

Consequently, in the third phase, SDT was replaced by WMT, providing two different types of extrinsic motivation: external regulation and internalized regulation. The internalized regulated motivation is the most autonomous, and less controlled external regulation. Amotivation was also added to the SDM CC definition as non-rational social actions lack reasoning, i.e., these concepts very much align. Regarding the type of motivation, it is possible to view the replaced intrinsic motivation by internalized regulated motivation. The non-rational social actions that previously were stated as extrinsic motivation were replaced by amotivation, which is based on WMT (Gagné & Deci, 2005).

In summary, I created six stereotypes and further developed the analytical steps. The framework was also transformed into a new layout, a table. The table consists of the six stereotypes, followed by type of rationale, and type of motivation.

The third phase included empirical data focused on the ASDM practice DSM. This time data from three SD teams was used. The empirical material was analyzed with the new version of the SDM CC framework. In 2023, the results were published and presented at an international research conference, Hawaiian Conference on Systems Science (HICSS56).

Based on the feedback I received, the major lesson to bring into the fourth and final version of the theory and the framework, as well as to the refined
definition of SDM CC behavior, was that there is a need for clarifications of the stereotypes as they now appear to be difficult/complicated to grasp.

**Fourth and final version of the SDM CC theory**

Based on the lessons learned from the third version of the theory, further elaboration was made. As the lessons learned pointed out in the third phase, the stereotypes were difficult to grasp as some of them contained several aspects, which made the empirical findings on the stereotypes complicated to present in a clear way.

Based on the lessons learned about unclarities, in combination with additional reading of literature and empirical analysis, the following elaborations were made to the theory. Havstorm et al. (2023) two stereotypes (fragmental and achievement) were divided into four, as the two existing stereotypes contained multiple motivations. This change resulted in two revised stereotypes and two added stereotypes. The theory now contains eight ideal stereotypes.

The fourth phase included empirical data focusing on the ASDM practice CI, SPD, and use of VN, from three SD teams. The empirical data was analyzed using the new version of the SDM CC framework, and its eight SDM CC stereotypes.


The Swedish Research School of Management and Information Technology MIT

The Swedish Research School of Management and Information Technology (MIT) is one of the 16 national research schools supported by the Swedish Government. MIT is jointly operated by the following institutions: Blekinge Institute of Technology, Chalmers University of Technology, University of Gothenburg, Jönköping International Business School, Karlstad University, Linköping University, Linnaeus University Växjö, Lund University, Mälardalen University, Stockholm University, Umeå University, Örebro University, and Uppsala University, host to the research school. At the Swedish Research School of Management and Information Technology (MIT), research is conducted, and doctoral education provided, in three fields: management information systems, business administration, and informatics.

DISSERTATIONS FROM THE SWEDISH RESEARCH SCHOOL OF MANAGEMENT AND INFORMATION TECHNOLOGY

Doctoral theses (2003- )


44. Röndell, Jimmie (2012), *From Marketing to, to Marketing with Consumers*. Department of Business Studies, Uppsala University, Doctoral Thesis.

45. Lippert, Marcus (2013), *Communities in the Digital Age: Towards a Theoretical Model of Communities of Practice and Information Technology*. Department of Business Studies, Uppsala University, Doctoral Thesis.


129. Lindeberg, Fredrik (2021), *Coordinating the Internet: Thought styles, technology and coordination*, Linköping University, Doctoral Thesis.


140. Mählér, Viktor (2022), *Succeeding Implementation: The Internet of Things as a Digitally Transformative Technology*. Department of Informatics, Umeå University, Doctoral Thesis.


143. Sun, Yunchen (2022), *Designing Routines for Industrial Digitalization*. Department of Business Studies, Uppsala University, Doctoral Thesis.


149. Rostami, Elham (2023), Tailoring information security policies - a computerized tool and a design theory. Department of Business Studies, Örebro University School of Business, Örebro University, Doctoral Thesis.

150. Hornbach, Jessica Janina (2023), Accounting for Accountability: Theoretical and Empirical Explorations of a Multifaceted Concept. Department of Business Studies, Uppsala University, Doctoral Thesis.


156. Williamsson, Ia (2023), Establishment Patterns of Model Diffusion in Academia and Practice. Department of Informatics, Linnaeus University, Doctoral Thesis.
