Agile Ambidexterity

Multiple Case Study of Finnish Software Development Organizations

Emma Castrén
Malin Gylling

Supervisor: Jonas Söderlund
English title: Agile Ambidexterity – Multiple Case Study of Finnish Software Development Organizations

Authors: Emma Castrén and Malin Gylling

Advisor: Jonas Söderlund

Publication type: Master’s thesis in Business Administration
Strategy and Management in International Organizations
Advanced level, 30 credits
Spring semester 2016
ISRN Number: LIU-IEI-FIL-A--16/02282--SE

Linköping University
Department of Management and Engineering (IEI)
www.liu.se
ABSTRACT

Title: Agile Ambidexterity - Multiple case study of Finnish software development organizations

Authors: Emma Castrén and Malin Gylling

Date: 31st May 2016

Background: Exploring the ambidexterity literature in the context of agile software development organizations from the perspective of how the organizational characteristics that result from the application of agile methods affect the achievement of ambidexterity

Aim: To gain insight into how agile software development organizations achieve ambidexterity.

Methodology: How agile software development organizations achieve ambidexterity was studied through a multiple case study where the total of four case projects in two different organizations were examined.

Findings: This study indicated how the characteristics of agile software development organizations have an essential role in how ambidexterity is achieved in these organizations.

Keywords: ambidexterity, exploitation, exploration, agile software development organizations
ACKNOWLEDGEMENTS

This thesis presents the finish line of the past two years of studying in the SMIO programme. This thesis journey was anything but easy; it included great amounts of confusion, frustration and sometimes even desperation. However, we would be lying if we said we didn’t enjoy every bit of it since “with great challenges come great rewards” and this indeed is the essence of our spirit throughout this journey. With the following words we want to show our sincerest gratitude for everyone who have been a part of our journey.

Firstly, we would like to give our deepest thanks to our thesis supervisor Jonas Söderlund. His comments and guidance during the process were extremely valuable and his feedback made us develop our thinking and always aim higher.

We would also like to thank our seminar group: Edgaras Rakevicius, Louis Auzias, Annika Aarnio and Ellen Kimber. They all provided excellent and valuable feedback throughout the process and questioned our ideas and thoughts and that way helped to clarify our sometimes confused minds.

We would like to give many thanks to the case organizations, Fraktio and Druid, and especially the interviewees who took the time to sit down with us and share their thoughts. Without their help and the information gained through the interviews, this thesis would not have been possible.

In addition, our deepest gratitude goes to our fellow SMIO’s. The past two years have given us so much and that is greatly because of the people we have been extremely fortunate to have around us and share this unforgettable time with. Especially during the thesis process they have provided endless support, encouragement and advise.

Last, but most definitely not least, we would like to thank each other for all the moments, varying from complete bafflement to great joys and everything in between.

Emma Castrén

Malin Gylling

“Creativity is paradoxical. To create, a person must have knowledge but forget the knowledge, must see unexpected connections in things but not have a mental disorder, must work hard but spend time doing nothing as information incubates, must create many ideas yet most of them are useless, must look at the same thing as everyone else, yet see something different, must desire success but embrace failure, must be persistent but not stubborn, and must listen to experts but know how to disregard them.”

(Michael Michalko, 2011)
Table of contents

1. Introduction .................................................................................................................................................. 1
  1.1 Background .............................................................................................................................................. 3
  1.2 The issue .................................................................................................................................................... 5
  1.3 The purpose and objective ....................................................................................................................... 8
  1.4 Contribution and target groups ............................................................................................................... 10
  1.5 Delimitations .......................................................................................................................................... 11

2. Theoretical framework ................................................................................................................................. 13
  2.1 Ambidexterity - balancing competing demands ..................................................................................... 13
    2.1.1 Exploitation and exploration ............................................................................................................ 15
  2.2 Antecedents for ambidexterity ................................................................................................................ 17
  2.3 Importance of ambidexterity .................................................................................................................. 19
  2.4 Different approaches to ambidexterity .................................................................................................... 21
  2.5 Different types of ambidexterity - Point and distributed ................................................................. 24

3. Methodology .................................................................................................................................................. 28
  3.1 Research topic identification .................................................................................................................. 28
  3.2 Research design ...................................................................................................................................... 29
  3.2.1 Multiple case study ............................................................................................................................ 30
  3.3 Research context ..................................................................................................................................... 31
    3.3.1 Agile overview .................................................................................................................................... 32
    3.3.2 Common agile methods .................................................................................................................... 34
    3.3.3 Case organizations ........................................................................................................................... 35
  3.4 Data collection ......................................................................................................................................... 36
    3.4.1 Interviews .......................................................................................................................................... 37
  3.5 Data analysis ............................................................................................................................................ 41
  3.6 Research quality ....................................................................................................................................... 42
    3.6.1 Credibility .......................................................................................................................................... 44
    3.6.2 Transferability .................................................................................................................................... 44
    3.6.3 Dependability .................................................................................................................................... 45
    3.6.4 Conformability .................................................................................................................................... 45

4. Empirical findings ......................................................................................................................................... 47
  4.1 Fraktio ...................................................................................................................................................... 47
    4.1.1 Being a successful software development organization ................................................................. 49
    4.1.2 Client relationship ............................................................................................................................. 50
    4.1.3 Organizational environment ............................................................................................................ 51
    4.1.4 Communication and information flows ............................................................................................ 53
    4.1.5 Keeping it simple ............................................................................................................................... 54
  4.2 Druid ......................................................................................................................................................... 56
    4.2.1 Being a successful software development organization ................................................................. 58
    4.2.2 Client relationship ............................................................................................................................. 59
    4.2.3 Organizational environment ............................................................................................................ 60
    4.2.4 Communication and information flows ............................................................................................ 64
    4.2.5 Keeping it simple ............................................................................................................................... 65

5. Analysis ......................................................................................................................................................... 68
  5.1 Exploitation and exploration ................................................................................................................... 68
    5.1.1 Exploitation ....................................................................................................................................... 69
    5.1.2 Exploration ........................................................................................................................................ 73
  5.2 Agile ambidexterity ................................................................................................................................... 76
    5.2.1 Organizational structure and culture ................................................................................................. 76
    5.2.2 Contextual approach to ambidexterity ............................................................................................. 78
    5.2.3 Distributed type of ambidexterity ..................................................................................................... 82
  5.3 Bringing it all together - Ambidexterity in agile software development organizations ................... 86
5.4 Agile software development organizations as ambidextrous communities ........................................... 90

6. Conclusion ............................................................................................................................................. 92
   6.1 Back to the beginning ..................................................................................................................... 92
   6.2 Providing the answers .................................................................................................................... 93
   6.3 Implications .................................................................................................................................... 97
   6.4 Limitations ....................................................................................................................................... 97
   6.5 Areas for future research ............................................................................................................... 99

References .................................................................................................................................................. 102

Appendix .................................................................................................................................................... 108
   Appendix 1. Agile Manifesto and Principles ......................................................................................... 108
   Appendix 2 – Interview guide ............................................................................................................... 109

List of figures
   Figure 1. Histogram of all papers on ambidexterity between 1996 and 2012 ....................................... 5
   Figure 2. Outline of thesis .................................................................................................................... 12
   Figure 3. Point and distributed ambidexterity ..................................................................................... 25
   Figure 4. Fraktio interview details ....................................................................................................... 39
   Figure 5. Druid interview details ......................................................................................................... 39
   Figure 6. Methodology summary ......................................................................................................... 46
   Figure 7. Summary of the empirical findings ....................................................................................... 66
   Figure 8. Exploitative and exploratory activities in the case projects ................................................. 69
   Figure 9. Summarizing framework of the analysis ............................................................................... 87
1. Introduction

“And the trouble is, if you don’t risk anything, you risk even more.”
(Erica Jong, 1973)

Organizations today continuously meet challenges with adapting to the changing business environments through new solutions and innovations, and at the same time make use of and improve existing capabilities and operations. These two modes, referred to as exploitation and exploration in this thesis, often require different attention and actions from organizations and, therefore, create competing tensions, which organizations need to manage. In addition, not only do the dynamic and complex business environments create the competing demands but also intensify them (Gaim & Wåhlin, 2015). Yet, acknowledging the existence of these demands is crucial, since according to Birkinshaw and Gupta (2013), managing existing tensions between competing objectives is a central part of what organizations do and balancing them is an essential part of ensuring successful performance. However, they posit a dilemma that organizations often find themselves struggling with.

Indeed, the importance of balancing competing demands for organizational performance has been noted in the literature. According to Gaim and Wåhlin (2015), existing evidence shows that engaging in paradoxes leads to short-term peak performance, which in turn reinforces long-term success, however, how this can be done in practice still remains somewhat unknown. The inability to reach balance between the two activities can have drastic effects. Cao, Gedajlovic and Zhang (2009) argue that failing to find balance between competing demands, i.e. exploitation and exploration, can leave a firm exposed to either risk of obsolesce or the risk of failure to appropriate. Extensive focus on exploitation will enable improvements on existing capabilities, however, it can drive an organization to a vicious success trap where the organization develops solutions that are obsolete. Conversely, organizations focusing primarily on exploration might face difficulties realizing the benefits of new innovations (March, 1991) and searching innovations after another can lock organizations in a failure trap (Levinthal & March, 1993). History provides
us with some famous examples of the inability to pursue both activities and find a balance between them. Motorola and Kodak faced dramatic technological changes in their respective industries and even though both companies recognized the changing market requirements and took initial actions to adapt to them by creating new assets, they still failed to create coherence and support between exploitative and exploratory activities.

Instead of having to choose one demand over other, organizations can strive to resolve the dilemma and pursue to become ambidextrous and balance both demands. Being ambidextrous, according to Merriam-Webster’s dictionary, means being “able to use both hands equally well”. In its simplicity, this definition reveals the essence of ambidexterity, however, in business context definitions of the concepts are broader. According to Eriksson (2013:334), ambidexterity “involves the capability to both exploit existing knowledge and technologies for short-term profits and also explore new knowledge and technologies to enhance long-term development”. Indeed, ambidexterity is viewed as providing a solution for managing and resolving the dilemma between exploitation and exploration. Furthermore, literature on ambidexterity emphasizes its positive impact on firm performance and many authors have concluded that ambidexterity leads to higher performance (Birkinshaw & Gibson, 2004) and firm survival (O'Reilly & Tushman, 2013), and the likelihood to gain more success (Vinekar, Slinkman & Nerur, 2006). In fact, Smith (2014) argues that long-term performance depends on engaging in both exploitation and exploration, i.e. being ambidextrous.

An industry where contradictory strategic choices are increasingly present is the software industry as companies need to develop customized and packaged solutions for the market and clients (Napier, Mathiassen & Robey, 2011). Wang and Rafig (2014) add that especially high-tech companies that operate in a dynamic market environment are often forced to consolidate existing businesses while simultaneously finding new opportunities. Geraldi, Kutsch and Turner (2011) found that in IT projects exploitation arises in the form of individual projects applying elements of standardised technology solutions with well-defined operational processes and structures. Conversely, exploration arises from the unique set of challenges that each customer requirement pose (ibid). Nowadays, many software organizations follow agile methods in their business operations, which can be
seen creating a new dimension to the ways exploitation and exploration are pursued and balanced due to the specific organizational characteristics that result from the application of agile methods.

Advocating for alternative value propositions, agile software development arose against the backdrop of the traditional approach to software development (Boehm, 2002). The traditional software development approach advocates extensive planning, codified processes and predictable activity in order to strive for perfection (ibid). Conversely, in agile software development, change is embraced, individuals and interactions between them are of high importance and the software is developed in short iterative cycles with close customer collaboration (Moran, 2015:1). A recent survey conducted in Finland revealed that in the top 5 of most renowned companies were software companies that advocate loose hierarchical structures, support and facilitation rather than strong leadership and individual responsibility and freedom (Saarinen, 2016), characteristics that greatly resemble agile. Indeed, the popularity of agile has increased as a vast amount of software companies nowadays carry out business operations through agile methods and Dingsøyr, Dybå and Moe (2010) stress that agile software development has a major influence how software development is conducted. Regarding Geraldi et al.’s (2011) notions on exploitation and exploration, in agile software development these issues become highly relevant due to fast pace, which on the one hand requires the use of well proven technology solutions, and the close client relationships that each, on the other hand, impose unique solutions.

1.1 Background

Business management literature has for long engaged in extensive discussions on how organizations can ensure the short-term and long-term viability. In the quest to do so, organizations are forced to react and adapt to changes as well as engage in innovation to ensure sustained successful performance. This leads into a situation where organizations are required to exploit existing capabilities, resources and processes while simultaneously exploring new opportunities and innovations. However, this posits a dilemma as exploitation and exploration demand different resources and capabilities from
organizations and therefore they are considered to be competing against each other. Furthermore, March (1991) asserts that exploration of new alternatives decreases the speed with which existing skills can be improved. Conversely, improving existing procedures can result in experimenting other procedures less attractive (ibid). The existing literature has presented and discussed a solution for organizations to resolve this dilemma, namely being ambidextrous, and pursue exploitation and exploration simultaneously.

Birkinshaw and Gupta (2013:290) call ambidexterity “an academic construction”. This phrase exemplifies the theoretical nature of the concept and its ambiguous meaning in practice. Yet, ambidexterity itself is not a new area of research as the concept was initially introduced by Duncan in 1976. Thereafter, ambidexterity has been researched in various contexts and concepts with reference to for example exploration and exploitation, efficiency and flexibility, and alignment and adaptability (Birkinshaw & Gupta, 2013). Since the concept was first introduced, ambidexterity has received increased attention and has now reached a rather trendy position in the research field (ibid). Exemplifying this, figure 1 shows how the concept of ambidexterity in research papers has increased during the last decade, and studies on ambidexterity are burgeoning. However, this increased interest towards ambidexterity does not come without drawbacks as the research field on ambidexterity has become fragmented. On the one hand, it is good that the literature field around ambidexterity becomes stronger and receives more and more attention through refinement and extension of the concept (Raisch, Birkinshaw & Tushman, 2009). On other hand, ambiguity around the concept is strongly present and the prior research done in various contexts has resulted in decreased clarity around its meaning and measurement (Birkinshaw & Gupta, 2013). Yet, Birkinshaw and Gupta (2013) argue that ambidexterity is a useful way to frame the challenges organizations face when managing two competing objectives simultaneously.
1.2 The issue

The competing demands balanced in ambidexterity have been conceptualized as dualities in various ways. Studies on competing demands have included concepts such as certainty and flexibility (Thompson, 1967 in O’Reilly & Tushman, 2013), exploitation and exploration (March, 1991) and adaptability and alignment (Gibson & Birkinshaw, 2004). Since the organizations being exemplified in this thesis operate in the software industry, where organizations are constantly confronted by dual demands of bringing new products and processes to the market and making use of the products and processes they have at hand (Chandrasekaran, Linderman & Schroeder, 2012), the concepts of exploitation and exploration are chosen to best describe and exemplify the tensions. In order for organizations to flourish and survive in the software industry, pursuing exploitation and exploration simultaneously is highly relevant and the reason for why this thesis acknowledges the concepts exploitation and exploration over other concepts also conceptualized to ambidexterity.

Corbett, Cornelissen, Delios and Harley (2014) note that even though the understanding of ambidexterity is not complete, its importance to organizations for innovation, organizational performance, adaptation and survival is clear. O’Reilly and Tushman (2013) argue that the essence of ambidexterity is in the organization’s ability to leverage existing assets and capabilities from the mature side of business in order to gain competitive advantage in new areas. This capability has proven to be of great importance in today’s dynamic and complex business environment. Ambidexterity becomes highly relevant and interesting for
the case organizations since the industry has been argued unable to temporally separate exploitation and exploration in order to remain competitive in the fast changing environment (Chandrasekaran et al., 2012). Indeed, both business and technology worlds have become more uncertain and turbulent (Cockburn & Highsmith, 2001), demanding organizations to in an effective way respond to this change with simultaneous pursuit of exploitation and exploration (Cao et al., 2009). On the same note, Chang (2015) argues that ambidexterity is especially important in software organizations since they need to find innovative ways to flexible satisfy client needs and deepen existing client base to support set targets.

A great amount of research on ambidexterity has evolved around the management of tensions at the higher organizational levels (Simon & Tellier, 2015). On the same note, Birkinshaw and Gupta (2013) emphasize that the dilemma between exploitation and exploration and the challenge to achieve ambidexterity can be resolved at the organizational level, however, this creates a new dilemma as we move down the organizational hierarchy all the way down to the individual worker. Interestingly, Turner and Lee-Kelley (2013) state that research on ambidexterity has focused at the organizational level and describing the “What” of ambidexterity, yet, there is a lack of understanding of the “How”, namely the underlying mechanisms, architectures and dynamics through which exploitation and exploration can be balanced. As noted, the current literature emphasizes managerial decisions and actions in achieving and managing ambidexterity, whereas organizations that apply agile methods are more based on self-organizing teams with less hierarchical structures and high degree of collaboration and individuals being responsive and equal. Therefore, it is of high importance to examine how organizations characterized by these issues achieve ambidexterity, as the mentioned issues can be seen creating different circumstances for organizations in their attempt to achieve ambidexterity, and therefore the agile setup provides an interesting context in which to study ambidexterity.

Even though the ambidexterity literature has explored several contexts, to the best of our knowledge, prior research on ambidexterity conducted with agile software development organizations is somewhat scarce. Existing studies can be found in context of project-
based organizations and on team level, however, they might not cover the specific aspects resulting from application of agile methods. In the context of software development organizations, prior literature has studied how these organizations achieve ambidexterity through different subunits with one being agile and the other traditional (Vinekar et al., 2006), thus proposing structural ambidexterity. However, as Napier et al., (2011) note, even though this structural strategy might be effective, organizations with fewer resources might face difficulties in realizing this strategy. Instead, they proposed contextual ambidexterity being the approach in software development teams, and based on their study they posit that developing a high performance context enhances the organization’s ability to align and adapt (ibid; Napier et al., 2008). Similar findings were provided by Ramesh, Mohan and Cao (2012) who studied conflicts between alignment and adaptation in agile distributed development, where project stakeholders are dispersed, and found that these organizations develop contextual ambidexterity. Even though the before-mentioned studies provide valuable insight into ambidexterity in software development organizations, we argue that there is room for further research with a stronger focus on the agile methods as more insight is required to understand agile methods effect on the organizational context in relation to ambidexterity.

Indicating the need to examine ambidexterity in the lower organizational level, Napier et al. (2011) argue that project-level considerations are especially important in software organizations. Therefore, it becomes of interest to seek additional insight into and understanding of the lower organizational levels. The level of analysis will therefore be on the case level, however, the findings from the cases will be reflected to the organizational level. Thus, the main aim of this study is to gain further insight into how agile software development organizations achieve ambidexterity. As will be displayed later in this thesis, organizations that follow agile methods are likely to develop specific characteristics that can affect the way these organizations balance exploitation and exploration, i.e. achieve ambidexterity. As noted, a great amount of existing research and literature have emphasized ambidexterity being managed at the senior management level whereas agile organizations rarely embody strong hierarchical structures and thus might lack managerial positions. Therefore, understanding ambidexterity in this specific context becomes of great interest since if ambidexterity is indeed managed from the higher organizational levels, this
can be considered to create different scenarios in how agile organizations manage and achieve ambidexterity.

The above-mentioned issues serve as a strong steppingstone for this thesis. Even though studies on ambidexterity at the organizational level are common, its presence and effects on lower levels in an organization, e.g. team and individual level, is still an area where further research is needed. Previous literature also describes that achieving a balance between exploitation and exploration is heavily affected by formal and contractual aspects such as hierarchical structures and control mechanisms (Eriksson, 2013). In this thesis we will stretch and challenge these aspects by the use of two agile software development organizations and bring the aspect of ambidexterity down from organizational level to team level where formal and hierarchical aspects of the organizations are less apparent through an agile way of working.

1.3 The purpose and objective

The aim of this thesis is to gain further understanding and develop a contribution to the existing stream of literature regarding ambidexterity, and more specifically provide an insight into ambidexterity in agile software development organizations. As pointed out earlier, ambidexterity research is continuously evolving through studies done in various contexts and perspectives, resulting in vague conceptions and various meanings and therefore, the authors feel that there is a need to further shed light into the ambiguous area of research. Moreover, the software development industry becomes of interest as more and more organizations are delivering software as a service and applying agile methods in doing so, which further emphasize that agile software development organizations provides a current and intriguing context in which to examine ambidexterity.

In order to map these issues, it is important to review and distinguish the exploitative and exploratory activities pursued in the case projects and how the application of agile methods influences these activities. Here, exploitative activities reflect reliance on existing technology, knowledge and processes while exploratory activities entail experimentation and testing of new technology, knowledge and processes. Furthermore, by reviewing the
existing literature, specific organizational characteristics can be recognized which have been highlighted in the discussion concerning agile software development. These characteristics will be taken into consideration when further examining and analyzing ambidexterity in agile software development organizations.

Thus, the following research question is posed:

**RQ:** How is ambidexterity achieved in agile software development organizations?

In order gain more insight into the ambidextrous activities and answer the research question, the following sub question is proposed:

**SQ1:** How do agile methods affect how these organizations exploit and explore?

Detailed look into the practices that are taken, who are involved and how the practices are implemented would provide valuable insight and generate this thesis and area of research with important knowledge regarding the concept of ambidexterity (Birkinshaw & Gupta 2013). As noted earlier, in order to be able to gain insight into how ambidexterity is achieved in agile software development organizations, it is essential to map how do these organizations exploit and explore. Moreover, the common approaches to realizing ambidexterity suggested in the current literature will be reviewed in order to identify whether the organizations follow one or a combination of the different approaches, and whether the findings support the existing research.

Simon and Tellier (2015) conclude that balance between exploitation and exploration can be achieved at the project level through particular network configuration between the individuals in the project. The level of analysis is on the case level and therefore the purpose is to gain insight into the activities taken within the case team and then further apply and review the findings on an organizational level to increase the understanding concerning ambidexterity in agile software development organizations. Rather than having a broader organizational focus from the beginning, the case level focus enabled the interviewees to better frame their responses into examples. In addition, examining two
different cases in each organization provided opportunities to gain various aspects and opinions and in that way also identify potential variation.

In order to gain insight into the activities concerning ambidexterity in an agile organization, two Finnish software development organizations, Fraktio and Druid are studied. Fraktio and Druid can be described as agile software development organizations through their agile work methods, and the organizational cultures that reflect the core concepts of the agile methods. The applied research approach is a combination of the deductive and inductive approaches as the thesis aim to test existing theories and also propose an additional framework based on the findings. With studying four cases, two from each organization, a multiple case study was chosen as a research design as it enables the thesis to replicate and extend the findings and theoretical framework among individual cases (Eisenhardt, 1991). Furthermore, multiple case study enables the researchers to draw a more complete theoretical picture and its comparative logic supports reviewing the similarities and differences between the different cases (ibid). Indeed, examining multiple cases enables comparisons through which the researchers can clarify whether the emergent findings are simply idiosyncratic to a single case or consistent in several cases (Eisenhardt & Graebner, 2007). The multiple case study design aids the comparison on various levels. Firstly, the two cases within one organization can be compared to gain a deeper understanding of each organization. Secondly, as the findings are reflected to the organizational level, comparison can be made between the two case organizations as well. Therefore, examining two cases within each organization enables the researchers to get a broader view and validate the findings. Thus, despite the main focus being on the case level, studying two cases from each organization enables the researchers to further reflect the findings to the organizational level.

1.4 Contribution and target groups

The main contribution of this thesis will be the added knowledge of and insight into how ambidexterity is achieved in agile software development organizations. More specifically, this study will review ambidexterity in terms of the specific characteristics and context of agile software development organizations. Testing and extending the current literature
presented in this thesis can serve as an additional source of reference for further research in this area in order to more profoundly understand ambidexterity in organizations relying on agile methods. This is of importance since great amount of the prior research and literature on ambidexterity has revolved around more traditional organizational structures, examining the concept in agile software development organizations can unveil some intriguing aspects for the academia.

Thus, as noted, this thesis aims at support and deepen the existing literature and provide further knowledge on how ambidexterity is achieved in agile software development organizations. The target groups are both academic and business audiences and the findings of this thesis are aimed to give organizations additional view on ambidexterity in the specific context of agile. Indeed, this is proposed by Lee, Delone and Espinosa (2006), who argue that understanding how to achieve ambidexterity in software development is imperative for global managers and developers.

1.5 Delimitations

This study is conducted with companies that operate in Finland, thus the findings can be considered to reflect this specific national context. Furthermore, the case organizations are rather small in terms of employees (both having 20-30 employees). Therefore, the scope of the thesis will be narrowed to reflect small Finnish software organizations that rely on agile methods in the business operations. The decision for narrowing to this specific context was mainly based on the access that the researches had to the case organizations.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1: Introduction</td>
<td>This first chapter presents the overall thesis topic to the reader through displaying the issue at hand as well as the purpose and objective of the thesis. In addition, the relevance for and contribution to the existing research field and literature will be discussed.</td>
</tr>
<tr>
<td>Chapter 2: Theoretical framework</td>
<td>The theoretical framework is based on the literature concerning the main concepts reviewed for this thesis. These main concepts consist of ambidexterity in organizations, the importance of it and the different approaches through which organizations can realize ambidexterity.</td>
</tr>
<tr>
<td>Chapter 3: Methodology</td>
<td>This chapter will present the key issues regarding the research process, such as research design, data collection and analysis methods. Furthermore, the chapter will present the research context, namely agile, in more detail as well as the case organizations, how and why they were chosen for this thesis.</td>
</tr>
<tr>
<td>Chapter 4: Empirical findings</td>
<td>This chapter will focus on presenting the empirical findings that were obtained from the interviews. In addition, this chapter will provide a short summary of each of the four cases.</td>
</tr>
<tr>
<td>Chapter 5: Analysis</td>
<td>The fifth chapter will go on analyzing the obtained findings in relation to the main issues and themes presented in the theoretical framework. Additionally, a model of the key points from the analysis is presented.</td>
</tr>
<tr>
<td>Chapter 6: Conclusion</td>
<td>The last chapter will provide a conclusion based on the analysis and discuss some managerial implications and issues concerning future research as well as limitations with the thesis.</td>
</tr>
</tbody>
</table>

**Figure 2.** Outline of thesis
2. Theoretical framework

The aim of this chapter is to set and discuss the theoretical framework for this thesis. The importance of theoretical framework is noted by Van De Ven (2007:19) who states that theory is influential in directing a research study and that selecting and building a theory is perhaps the most strategic choice taken when doing a study. Thus, this chapter is carefully constructed of concepts and issues that are essential for this study.

The theoretical framework outlines the relevant theories for this thesis, including definitions and discussions around them. The chapter is constructed with focus on one main concept. The main concept is ambidexterity, a concept that is a rather theoretical and rarely described as such in everyday language. The term simply implies balancing between two competing demands, and throughout this thesis the competing demands will be explained through activities of exploitation and exploration. Moreover, organizational antecedent for ambidexterity and the common approaches to ambidexterity suggested in the current literature are discussed. Lastly, a novel framework of different types of ambidexterity is reviewed.

2.1 Ambidexterity - balancing competing demands

Grant (2013:189) argues “everything is in a state of constant change - the business environment especially”. This creates pressure for organizations and in fact a crucial aspect for organizational survival is to be able to reach a balance between long-term success and short-term profit. Therefore, organizations need to have a strategy that is not only “competing for today” but also “competing for tomorrow” (Grant, 2013:18) as in today’s dynamic business environment organizations need to ensure continuous development of existing operations and capabilities as well as grasp novel opportunities and solutions. This, however, has proven to be troublesome for organizations since the ways these two activities are pursued and the resources and actions they require can be
somewhat conflicting. Indeed, Smith and Tushman (2005) state that the exploitation and exploration are associated with different architectures and processes.

As pointed out earlier, research on ambidexterity has received increased attention in strategy and management studies (O’Reilly & Tushman, 2013). Furthermore, its importance for firm performance is widely acknowledged (Birkinshaw & Gupta 2013; O’Reilly & Tushman, 2013), however, the way ambidexterity is defined makes the concept overly flexible as it can be studied and applied to nearly any issue in organizational research. According to Merriam-Webster dictionary, being ambidextrous means being able to use both hands equally well. Even though this definition is good in its simplicity, it is not suitable for business context. In business context, in its initial introduction by Duncan in 1976, organizational ambidexterity was regarded being achieved through dual structures where exploitation and exploration are divided into different organizational units (Papachroni, Heracleus and Paraoutis, 2014). Other definitions do not go that deeply into structural issues when explaining ambidexterity as Andriopoulos & Lewis, 2009 simply describe ambidexterity as an organization’s ability to manage the balance and tension between two competing demands. Indeed, Turner, Maylor and Swart (2015) posit that the state of ambidexterity acknowledges the co-existence of both exploitation and exploration and that they can occur simultaneously rather than being at opposite ends of a continuum. Thus, by becoming ambidextrous, an organization can balance both exploitation and exploration (ibid). Birkinshaw and Gupta (2013:291) define ambidexterity as “an organization’s capacity to address two organizationally incompatible objectives equally well”, and similarly, He and Wong (2004:484) suggest “a firm is regarded as ambidextrous if it has relatively equal emphasis on both dimensions”. Also, Raisch et al., (2009:685) address ambidexterity as “organizations that are capable of simultaneously exploiting existing competencies and exploring new opportunities”.

With reference to knowledge, Turner, Maylor and Swart (2011:2) define ambidexterity as “the ability to both use and refine existing domain knowledge (exploitation) whilst also creating new knowledge to overcome knowledge deficiencies or absences identified within the execution of the work (exploration)”. However, they, as well, conclude that the way the concept is defined is extremely flexible (ibid). On the same note, continuing with the
knowledge aspect, Eriksson (2013:334) states “ambidexterity involves the capability to both exploit existing knowledge and technologies for short-term profits and also explore new knowledge and technologies to enhance long-term development”. With a stronger reference to innovation, Tushman and O’Reilly (1996:24) describe ambidexterity as, “the ability to simultaneously pursue both incremental and discontinuous innovation [...] from hosting multiple contradictory structures and processes within the same firm”. Yet, O’Reilly and Tushman (2011:330) argue there is more to ambidexterity than just the ability for an organization to pursue efficiency and innovation, and they argue that ambidexterity is about “developing the capabilities necessary to compete in new markets and technologies that enable the firm to survive in the face of changed market conditions”.

As noted, multiple definitions of the term ambidexterity have been proposed (Birkinshaw & Gupta, 2013; He & Wong, 2004; Raisch et al, 2009; Tushman & O’Reilly, 1996; Turner et al., 2015; Turner et al., 2011 and Eriksson, 2013) and this thesis will build upon the concepts from Eriksson (2013) who considers the knowledge and technology aspect of ambidexterity, which best suit the essence of this thesis. Evidence on the effects from ambidexterity has been consistent across industries but as shown, the meaning of the term ambidexterity differs widely. In order to prevent creating confusion or conflicting findings in the empirical chapter, being precise in the definition is highly vital (O’Reilly & Tushman, 2013), as to why the definition of Eriksson (2013) is chosen.

2.1.1 Exploitation and exploration

Various competing demands have been examined and discussed in the existing literature regarding ambidexterity, and it has been highlighted in the previous section that the main competing concepts for this thesis will be exploitation and exploration. Exploitation and exploration is seen to best describe the tension and relevance for organizations operating in the software industry, which is being examined in this thesis.

In his publication, James March (1991:71) connected exploitation with processes such as “refinement, choice, production, efficiency, selection, implementation and execution” whereas exploration entails processes such as “search, variation, risk taking,
experimentation, play, flexibility, discovery and innovation”. Thus, the essence of exploitation lies in “refinement and extension of existing competencies, technologies and paradigms” whilst the essence of exploration fixates in “experimentation with new alternatives” (March, 1991:85). While providing a good basis, these definitions give room to broad and various interpretations. Pellegrinelli, Murray-Webster and Turner (2014) describe exploitation as deploying existing knowledge and capabilities while exploration refers to experimentation, searching for novelty and encouraging creativity, extending boundaries of prevailing practices and conceptions and accepting failure. On the same note, Lavie, Stettner and Tushman (2010) posit that exploration refers to non-routine problem solving and searching for new knowledge. In their article, Gilsing and Nooteboom (2006) present some key characteristics of exploration and exploitation. The former includes radical innovations, high frequency of interaction, tacit knowledge, experimentation with novel combinations and short duration. The latter consists of incremental innovation, experimentation in organization, codified knowledge, low frequency of interaction and long duration (ibid). Furthermore, in prior work, exploitation and exploration have also been discussed in terms learning through local and distant search. Katila and Ahuja (2002) distinguish between local (exploitative) and distant (exploratory) search where in the former, problems are addressed with pre-existing knowledge bases and the latter involves conscious efforts to drift away from current routines and knowledge bases. Indeed, exploitation and exploration are learning-related activities (Gupta, Smith & Shalley, 2006), where exploitative learning refers to increasing the depth of knowledge while explorative learning increases the breadth (scope) of knowledge (Kauppila & Tempelaar, 2016).

Lavie et al. (2010) posit that even though the framework of exploitation and exploration is straight-forward, it has generated debates regarding the definitions of both modes, their measurement, antecedents and consequences. In his work, March (1991) described exploitation and exploration requiring and competing for scarce resources and therefore organizations are required to make explicit and implicit choices between the two, thus the two activities are viewed as two ends of a continuum (Gupta et al., 2006). However, this kind of trade-off relationship between exploitation and exploration have shifted into a paradoxical thinking (Eriksson, 2013). Paradox entails a challenging tension that has
interrelated elements existing simultaneously, yet they are contradictory (Lewis, Andriopoulos & Smith, 2014). Indeed, Lewis et al. (2014) state that viewing exploration and exploitation as paradoxical highlights their interdependent nature. Therefore, exploitation and exploration are viewed as orthogonal, that is as simultaneously achievable (Gupta et al., 2006). Moreover, the notion of balance between exploitation and exploration need not to be equal 50/50, as Andriopoulos and Lewis (2009) explain the balance does not denote a mediocre split between the two, but rather the ability to excel in both exploration and exploitation. Regarding the appropriate level of balance, most scholars dispute whether the appropriate level of exploitation and exploration is contingent with the organization’s mission, dominant logic or industry conditions (Lavie et al., 2010).

Indeed, Atuahene-Gima (2009) notes that the balance could imply that high exploitation may need to be coupled with low exploration or vice versa in order to enhance the firm performance. Too much exploration can lead to the firm moving away from one new idea to the next without exploiting prior learning and experience. Additionally, new products might be underdeveloped and their fit to customer needs might be indistinct. Thus, a small amount of exploitation can moderate potential excesses in exploration as companies are able to evaluate and assimilate new ideas more effectively. Conversely, introducing small amount of exploration will aid firms in overcoming the costs connected to exploitation. (ibid) Likewise, He and Wong (2004) note that exploitation of existing capabilities is usually required to explore new capabilities and vice versa. On the same note, from a knowledge perspective, Lavie et al. (2010) argue that distinguishing exploitation from exploration is challenging due to the multidimensionality of knowledge.

Due to the various amount of existing definitions of exploitation and exploration, it is important to emphasize one. Throughout this thesis exploitation will be described as the short-term focus on efficiency, based on existing knowledge and technologies, while exploration highlights the long-term focus on innovation, based on new knowledge and technologies (Eriksson, 2013).
2.2 Antecedents for ambidexterity

Contextual factors that are specific for ambidextrous organizations have been identified in the existing literature. Ramesh et al. (2012) suggest that these factors consist of structural context, culture and climate. Structural context entails the set of administrative mechanisms that foster particular types of behaviour within the employees. Secondly, organizational culture consists of the implicit values, beliefs and principles that are supported by the organizational practices. Lastly, organizational climate refers to the organizational and environmental characteristics that shape employees' behaviour. (ibid)

Indeed, Atuahene-Gima (2009) states that exploration and exploitation thrive under different organizational conditions, which can result in difficulties in combining them, i.e. to achieve ambidexterity.

Jansen, Van Den Bosch and Volberda (2006) discuss the organizational antecedents that support explorative and exploitative innovation. In fact, the authors hypothesized that formal hierarchical structure in terms of centralization and formalization affect explorative and exploitative innovation (ibid). Firstly, they argued that centralized decision-making decreases the likelihood that unit members suggest innovation and new solutions as it narrows communication channels and reduces quality and quantity of ideas and knowledge used for problem solving (ibid; Lavie et al., 2010). Following this notion, Adler, Goldoftas & Levine (1999) note that decentralized structure supports the function of headquarters being a facilitator rather than a controller. Yet, centralization of decision authority has opposite effect on exploitative innovation as it increases information-processing efficiency and thus facilitates exploitative innovation. Furthermore, formalization is likely to reduce variance through incremental improvements and therefore it is more supportive for exploitative innovation. Conversely, strong reliance of formalization of rules, procedures and instructions weaken and restrain experimentation as individuals are less likely to deviate from the structured behaviour. (Jansen et al., 2006)

Furthermore, having a common, underlying layer of strong culture and vision are essential for ambidexterity. Indeed, Jansen et al. (2006) discuss the informal social relations within an organizational unit and posit that connectedness increases opportunities for informal conversations and therefore further aids combining the knowledge of different individuals.
Moreover, they state that social relationships enable adoption of exploratory innovation (ibid). Lin and McDonough (2011) assert that a knowledge sharing culture that fosters the values of uncertainty tolerance, openness to challenge, and trust may help to enhance the exploitation of existing knowledge and the exploration for new capabilities. Furthermore, greater behavioural integration enables coping with contradicting knowledge processes, i.e. exploration and exploitation (ibid). Likewise, Simon and Tellier (2015) argue that in order to achieve ambidexterity, teams of individuals rely on different network structures and types of ties. However, too dense networks can have a hindering effect as well. Strong norms and established shared behavioural expectations, that are common in highly-density networks, might limit access to divergent views and alternative ways of working, thus, constraining departure from existing knowledge and pursuit for exploratory innovation. Conversely, in terms of exploitative innovation, connectedness enables trust building and cooperation that further support the development of deep understanding from which the units can refine and improve existing products, processes and markets. (ibid)

Another important aspect to ambidexterity is pointed out by Cao et al. (2009), and is argued to be the organization size. A larger organization is less exposed to the risks of failure and obsolescence due to their larger resource base, which provides them with a better buffer. On the contrary, smaller organizations are more vulnerable for these risks due to their fewer resources to handle the risks (Cao et al. 2009). Furthermore, they propose resources to be another contingency with suggesting that success at pursuing combined dimension depends on the extent to which sufficient resources are accessible and allocated to support high level of engagement in both activities (ibid).

2.3 Importance of ambidexterity

It can be stated that the general notion of ambidexterity regarding business operations is mainly positive. Lavie et al. (2010) argue that March’s premise from 1991, stating that organizations, which balance exploitation and exploration are to gain superior performance, is made explicit in ambidexterity research. Indeed, ambidexterity has often been addressed in relation to firm performance and O’Reilly and Tushman (2013) note that many studies positively associate ambidexterity with firm performance in terms of e.g.
sales growth, innovation and firm survival. Similarly, Liu and Leitner (2012) argue that ambidexterity leads to sustained competitive advantage. Cao et al. (2009) agree this as they point out that achieving ambidexterity enables a firm to enhance its competitiveness. Same kind of positive notion was also noted by Junni, Sarala, Taras and Tarba (2013) through their meta-analysis of prior studies on ambidexterity, as they found that organizational ambidexterity was positively and significantly associated with performance. More specifically, they found that a number of studies on organizational ambidexterity argue that firms pursuing both exploitation and exploration are more likely to achieve superior performance compared to firms that pursue one dimension over the other (ibid).

Therefore, due to the positive consequences ambidexterity has on business performance and success, it becomes of even greater importance to strive to achieve ambidexterity. Andriopoulos and Lewis (2009) argue that achieving ambidexterity enables success and therefore, inability to reach balance between exploitation and exploration can have drastic effects of firm performance. A balance between the two activities is vital to achieve since strong focus on exploitation can result in what Levinthal and March (1993) call a “success trap” where returns are positive, proximate and predictable (March, 1991) resulting in increased emphasis on exploitation activities. Indeed, Grant (2013:210) argues that established capabilities and their embeddedness within organizational structure and culture create great barriers to building new capabilities, and therefore, the more developed the existing capabilities are, the greater barrier they create. On the other hand, strong focus on exploration can lead to a “failure trap” with failures leading to new searches and again to new failures (Levinthal & March, 1993), where returns are uncertain, distant and often negative (March, 1991). Therefore, as Levinthal and March (1993) argue, organizations engaging only on exploitation will commonly suffer from obsolescence and organizations focusing on exploration rarely gain returns on its knowledge. On the same note, March (1991) states that exploration of new alternatives reduces the speed of improving existing skills and conversely, improving competence in existing procedures results in experimentation in others becoming less attractive. Regardless of the noted effects of extensive exploitation, companies tend to emphasize short-term planning with exploitation of current resources and capabilities over long-term planning through exploration of new opportunities Grant (2013:204).
Even so, Levinthal and March (1993) argue that firm survival requires a balance between exploitation and exploration activities, however, the precise mix of the two that is optimal is difficult to specify. The common problem organizations face is to pursue exploitation sufficiently in order to secure current viability and simultaneously engage in exploration activities to ensure future viability (ibid). This emphasizes the importance of ambidexterity and strengthens the conception that this is an area that should be taken into close consideration and managed in order to reach optimal performance. It is argued that the positive effects of ambidexterity on firm performance are stronger for technology organizations and when the conditions are more uncertain (O’Reilly & Tushman, 2013). Indeed, Lavie et al. (2010) state that organizations need to adjust the corresponding levels of exploitation and exploration to changes in the environmental conditions.

2.4 Different approaches to ambidexterity

Ambidexterity can be achieved through exploitation and exploration in different ways and one universal way of how it occurs does not exist as the way ambidexterity is implemented and achieved often reflects the characteristics of the specific organization. Eriksson (2013) states that the balance how exploitation and exploration are balanced in various organizational levels is greatly affected by formal organizational and contractual aspects (e.g. hierarchical structures and control mechanisms) and informal social aspects (e.g. culture, cooperation and shared vision). While considering the way of achieving ambidexterity, existing literature provides three common approaches, namely structural, sequential and contextual ambidexterity.

Traditionally, the literature on ambidexterity has suggested a structural division regarding ambidexterity. The notion of focusing on exploitation and exploration separately through different units stems from Duncan’s work from 1976 (in Papachroni, Heracleous & Paroutis, 2014) in which the concept of ambidexterity was initially introduced. In structural ambidexterity, exploitation and exploration are separated in different units with the different units focusing on one or the other (Turner, Maylor & Swart 2015). Here, the top management team is responsible for the coordination between the two units (Papachroni et al., 2014). The suitability of this approach is noted by Adler et al. (1999) who point out
that organizations should make distinction between structures that facilitate exploitation and exploration. They refer this approach as partitioning and suggest that it offers opportunities to deepen skills through specialization and thus allows sustained and focused efforts to both routine and non-routine activities (ibid). The unit focusing on exploration can be exemplified through a R&D unit that embraces uncertainty related to innovation while the exploitative one can be related to a manufacturing unit where maximizing efficiency is of importance (Huang, Baptista & Newell, 2015). Further supporting the suitability of structural approach, Chen and Kannan-Narasimhan (2014) argue that separating exploratory unit from the parent organization aids insulating the existing inertia. Yet, structural division of exploitation and exploration activities can be more suitable for more established and larger organizations as Eriksson (2013) posits that structural ambidexterity can be commonly seen in large organizations where the two activities can be more easily separated due to greater resources and established operations. It is perhaps because of these of issues why Birkimshaw and Gibson (2004) argue that structural ambidexterity is the standard approach to ambidexterity. However, Chen & Kannan-Narasimhan (2014) state that structural ambidexterity is achieved at the business unit or corporate levels, thus applying it to team or individual levels is unfavourable. Furthermore, they argue that separated units are unable to share knowledge and resources with the mainstream units and therefore, even if structural ambidexterity is pursued it is essential to organizational processes to integrate the different units together to realize the full benefits (ibid).

Sequential ambidexterity, also often referred to as temporal ambidexterity, is pursued through shifts in structures over time (Eriksson, 2013). The organization moves from one domain to the other, which means that exploitation and exploration are separated in time (Turner et al. 2015). Lavie et al. (2010) posit that sequential ambidexterity, which they refer as temporal separation, is rooted in the notion of punctuated equilibrium. Here, people switch sequentially between the two types of tasks (Adler et al., 1999). Sequential ambidexterity can be exemplified through an exploratory phase of new product development, followed by an exploitative phase when the new product is moved to the market (Turner, Maylor, Lee-Kelley, Brady, Kutsch & Carver, 2014). Thus, the two activities are temporally separated and focus is first put on one type of activity and then the
other (Eriksson, 2013) rather than pursuing them simultaneously. Considering the advantages of sequential ambidexterity, Adler et al. (1999) argue that sequential switching between exploitation and exploration allows greater focus and reduces the risk of confusion. On the same note, Eriksson (2013) states that sequential separation of exploitation and exploration is more suitable in stable environments. However, Chen and Kannan-Narasimhan (2014) assert that switching between exploitation and exploration requires the development of process mechanisms and interpersonal relationships. Indeed, Lavie et al. (2010) state that the conflicting pressures for exploitation and exploration are still present at the time of transition.

Differing from the somewhat traditional, top-down approaches (Liu, Wang & Sheng, 2013) to ambidexterity described above, contextual ambidexterity refers to simultaneously pursuing conflicting demands on a business-unit level. The concept of contextual ambidexterity was presented by Gibson and Birkinshaw (2004), which brings the tension between exploitation and exploration to the individual and business unit levels through an organizational context that encourages individuals to make own judgements regarding the use of their time between the conflicting demands (O’Reilly and Tushman, 2013). This involves a single organizational unit and the same members within these units who pursue both exploitation and exploration (Grant, 2013:205). Here, according to Napier et al. (2008), ambidexterity is perceived as a shared responsibility shaped and reflected by the day-to-day activities of individuals. Contextual ambidexterity is enabled by a set of processes and systems together with a dynamic and flexible context that support individuals to manage competing demands (Gibson & Birkinshaw, 2004). This system is comprised of four performance management and social context related attributes that nurture an appropriate organizational context (Lavie et al., 2010), namely stretch, discipline, support and trust (Gibson & Birkinshaw, 2004). Stretch refers to provoking individuals to voluntarily pursue more ambitious objectives. More specifically, actions such as developing a collective identity and building a shared ambition contribute to the establishment of stretch. Secondly, discipline stimulates individuals to voluntarily aim to meet all expectations that are generated by their commitments. Here, organizations can establish standards for performance and behaviour, systems for open and rapid feedback, for example. Support attribute emphasizes assistance and encouragement that individuals
give to each other’s and it is enabled by freedom to take initiatives at lower levels and highlighting providing guidance and help over strong authority. Lastly, trust induces individuals to count on the commitments of each other. Here, common and equal decision-making power and filling positions with individuals who possess the required capabilities affect the establishment of trust. (ibid) Therefore, Gibson and Birkinshaw (2004) posit that when a supportive organizational context, including the before-mentioned attributes, is created individuals will pursue both exploitation and exploration-oriented actions. Indeed, they further note that the dynamic and flexible context in a contextually ambidextrous unit allows individuals to apply their judgement (ibid).

2.5 Different types of ambidexterity - Point and distributed

The existing literature on ambidexterity provides little explanation of how ambidextrous activities are divided between individuals in organizations. The different approaches of ambidexterity give meaning for how ambidexterity is implemented within an organization, yet, they do not fully take into account the individual level and whether or not individuals’ ambidextrous activities are to be considered as stand-alone or group efforts. A relatively new perspective to ambidexterity was developed by Turner, Swart and Antonacopoulou (2016). They distinguish two different types of ambidexterity from the identification of the varying types of exploitation and exploration actions at the individual and team level within projects. The concepts, namely point ambidexterity and distributed ambidexterity, provide valuable additional insight into the aspects of whether ambidexterity results from a single individual’s or a group of individual’s actions. Through their study the authors present five managerial actions that enable ambidexterity. (ibid)
Turner et al. (2016:12) define point ambidexterity when “an individual who is significant actor in creating group- or project-based ambidexterity, performing or coordinating both exploitative and exploratory actions that are not replicated by other individuals”. That is, point ambidexterity is achieved when a single individual in a project plays the key role and performs or coordinates both exploitative and explorative actions in the project. This was exemplified as an individual undergoing exploratory work with the client and thereafter determining solutions and making sure they are implemented by others in the team, which indicates that the explorative work lay the foundation for the exploitative work as the subsequent tasks are better defined. Here, the single individual is central in making sure that exploitation and exploration pursued in order to fulfill the project’s benefits. Therefore, project-based ambidexterity can be attributed to a single point, which is the single individual. Similar notion was proposed by Tushman, Smith and Binns (2011) in their hub-and-spoke model, where the responsibility for management of the tension between innovation and core products was on the CEO who then coordinates and negotiates the trade-offs with business unit and innovation leaders.

Based on their study, Turner et al. (2016:12) provide an alternative type of ambidexterity and define distributed ambidexterity as “the pattern of exploitative and exploratory actions among a group of individuals, the sum of which provides exploitation and exploration at the
level of the group, organization, project or work unit”. Therefore, on the contrary to point, in distributed ambidexterity, ambidexterity is not achieved by one single individual, rather when exploratory and exploitative actions are distributed among different participants in the team. The members in the team are more focused on one part of the project and undertake different but complementary work, which demonstrates aspects of exploitation and exploration. Therefore, distributed ambidexterity is more a group function. (ibid) Here as well, Tushman et al. (2011) propose a supportive aspect. They propose a ring-team model where business leaders are brought together and collective decisions take place regarding allocation of resources and trade-offs. The ring-team model entails a common seek for resolutions that fulfill overall need, both short-term and long-term, of the business (ibid).

Furthermore, the authors proposed that self-adjustment activities were involved since how individuals performed tasks that were not only connected to their personal ability and responsibility but also consistent with the ones of other project members. (Turner et al., 2016) Indeed, the authors propose that point and distributed ambidexterity are linked with self-adjustment and state that the relationship could entail significant dynamic effects since one type of ambidexterity might evolve towards the other in some situations (ibid).

Looking more closely to the actual activities, Turner et al. (2016) introduce five managerial actions that support and enable ambidexterity in project-based organizations. Through buffering, the manager acted as a barrier to prevent undesirable distractions. This can result in coordinated and efficient knowledge sharing within the team and reduce the likelihood that team members would be overwhelmed by task requests from the customer, some which might be conflicting the project plan. Another action is gap-filling, where managers deliberately beat deficiencies with performing tasks that were necessary but perhaps not being performed. As the role of project management is to ensure that all requirements are delivered, the gap-filling action fall on the project manager. Thirdly, integration is a central managerial action as it actively brings the knowledge within the project together. Important activities enabling integration are meetings and regular communication through which stakeholders, project staff and overall objectives can be aligned. Hence, integration can be seen supporting the completion of exploitative and
explorative tasks as they contribute to the overall achievement of desired benefits and it calls for sufficient understanding of all aspect of the work. The fourth action is role-expansion where the manager is required to stretch and do more than normally as a response to critical events that involved increased exploitation and exploration. Lastly, the fifth managerial action is tone-setting where the manager sets exploitative or exploratory ethos. Turner et al. (2016) connect this tone-setting to the concept of contextual ambidexterity with creating an organizational setting that encourages individuals to rely on their judgement to divide their time between exploitative and explorative tasks. (ibid) Even though all five managerial actions are essential in enabling ambidexterity, through their study, Turner et al. (2016) noted that one is the most significant for generating project-based ambidexterity, namely integration, due to its interwoven nature with the remaining actions.
3. Methodology

The following chapter will provide information about the methodological issues and aspects concerning this thesis. The chapter presents the actions taken during the research and sheds light into the research process as a whole. Structure of the chapter will follow as such; first, the research topic identification is discussed. Thereafter, a description of the research design and the chosen research strategy will take place, following with introduction of the research context and the studied case organizations, Fraktio and Druid. Furthermore, this part will display issues concerning data collection and analysis. Lastly, the issues regarding the quality of the research are discussed.

3.1 Research topic identification

The idea behind this thesis was built around the author's' interest in one of the main concepts of this study, namely ambidexterity. Furthermore, shortly after reviewing the current literature on ambidexterity the authors found that the concept had been and currently is a popular topic within the field of research as it has been studied in various contexts and from different perspectives, which increased the authors’ interest towards the concept. With the aforementioned main concepts in mind, the authors began to dig deeper into the potential aspects to explore.

Additionally, the interest to study agile organizations was another issue motivating this study. The authors were especially interested in the case organizations and their specific characteristics. Fraktio and Druid operate in a way, which emphasizes the employees as individuals and gives high priority to the social aspects of work. Indeed, it was the organizational culture and agile way of working that intrigued the authors to study the case organizations. It can be said that the authors considered Van De Ven’s (2007:7) concept of engaged scholarship where one central issue is that an engaged scholar views organizations and clients as learning place rather than data collection sites and funding sources. This kind of learning place will serve as an idea factory where practitioners and
scholars co-produce knowledge concerning important questions and issues through testing alternative ideas and views of common problems (ibid).

3.2 Research design

The research design provides the thesis with a framework for how collection and analysis of data will proceed (Bryman & Bell, 2003). The choice of research design reflects decisions about the priority being given to a range of dimensions of the research process (ibid) and will provide a general plan for how to answer the research question (Saunders, Lewis & Thornhill, 2012:159). A qualitative research methodology was applied in order to make sense of the issue being studied. The qualitative approach was chosen since the authors will have the opportunity to, for a short time, take place in the natural setting of the organizations being studied, and since the aim is to gather and develop an in-depth understanding (Saunders et al., 2012:163).

Bryman and Bell (2003) suggest that qualitative research is suitable for viewing how patterns and events unfold in terms of processes. As the aim of this study is to gain insight into the organizational behaviour, and more specifically how is ambidexterity achieved in agile software development organizations, Van De Ven (2007:22;194) further proposes a process model, which is usually connected to “How” questions, in order to examine change and development under an observed sequence of events. The processes and activities describing how ambidexterity is achieved in organizations still remain not fully covered in the academia, therefore this study will follow the main ideas of a process model in order to uncover events and activities and their origins and effects in the four specific cases. The findings obtained through following a process model is described as a post hoc knowledge by Van De Ven (2007:208), which helps interpreting the unfolded events and creating a narrative of the process. However, a limitation that the retrospective study has is that it can result in creating a bias since a tendency exists to filter out events that do not fit or make the story less coherent (ibid).

Regarding the research approach, characteristics of both deductive and inductive were applied. As this thesis will examine and test existing theories in the chosen research
context, a deductive approach is followed, even though no strict hypotheses were distinguished (Bryman & Bell, 2011). Additionally, as the research context provides new insight into the existing theory, an inductive approach arises. The in-depth understanding of the phenomenon being studied aims to develop and contribute to the existing literature, which strengthens the inductive approach (Saunders et al., 2012:146).

Moreover, this thesis will serve as an exploratory study as it gives the opportunity to discover what is happening and gain insight about the topic of the thesis (Saunders et al., 2012:171). The authors wish to clarify an understanding of the “problem” and gain insight about the topic of interest are reasons for why an exploratory study is valid in this case (ibid).

3.2.1 Multiple case study

Research strategy is the researchers plan of action to achieve the goal of the thesis, which typically includes a plan of how the researchers will go about in order to find the answers the research question (Saunder et al., 2012:173). The applied research strategy for this study is a multiple case study. This strategy was chosen since conducting a case study can provide the research with powerful examples (Siggelkow, 2007) and it is a favourable way to understand the dynamics within a single setting (Eisenhardt, 1989), namely agile software development organizations.

Advantages and disadvantages of both single and multiple case studies have been argued in the research literature. Dyer and Wilkins (1991) argue that a single case study is superior to a multiple case study when the goal is to create a high quality theory. They further argue that the quantity of data emerging from multiple cases is too large and the descriptions become rather thin with focus on the surface data and information instead of deeper social dynamics (ibid). Eisenhardt (1991) on the other hand argues that multiple cases can help researchers to develop a more elaborate theory and through multiple cases piece together individual patterns and complete a more accurate and theoretical picture. While Eisenhardt (1991) argues for a more complete picture by using multiple cases, Dyer and Wilkins (1991) see the problem with too much focus on the constructs
being developed and what is being measured that the researchers will miss the context and the rich backgrounds of the case. However, the aim of distinguishing patterns and gaining insight can instead by conducting a single case, the researchers can see new theoretical relationships and question old ones (Dyer & Wilkins, 1991).

As noted, a multiple case study strategy will be followed through examining four case projects in two case organizations. Examining two case projects in each organization enables the researchers to gain a broader view into the various activities taken within the case projects and through this view the researchers are able to draw conclusions that represent the organizational aspects as well. In addition, multiple case study provides a basis for comparison. The comparability is supported by Bryman and Bell (2003), who suggest that the logic of comparison enables the researchers to comprehend a phenomenon better when compared in relation to two or more contrasting cases or situations. Moreover, this strategy will further enable the comparison to be done on different levels. Not only can the two case projects be compared on a case level but also on the organizational level. Therefore, the different cases were chosen based on the possibility for the authors to discover similar, and also different, results from each one and by that amplify the point the researchers wish to express (Saunders et al., 2012:179).

3.3 Research context

A non-probability sampling was conducted where the researchers use a purposive sampling to select cases, which enabled the researchers to answer the research question and to meet the objectives of the thesis. According to Saunders et al. (2012:281, 287), this is an appropriate method in a smaller sample such as a case study research where researchers select cases based on what information they can provide. This sampling method was applied when searching for the case organizations and Fraktio and Druid were considered to represent possibilities to study the key concepts of this thesis, thus the sampling was purposive due to the case organizations suitability. In February 2016 when one of the authors of this thesis had the first meeting with the CEO of Fraktio, the thesis topic and the main issues were discussed, thus, providing a good basis for selecting cases that would represent these issues. The initial plan was to only include one case
organization, namely Fraktio, however the second case organization was discovered during the data collecting period as an opportunity arose to interview another, yet similar organization operating in the same industry and providing similar solutions. Regarding the second case organization, purposive sampling could not be applied as early as with the first case organization, however, the researchers were able to discuss the thesis topic and the main issues with the CEO of Druid a few days prior to the interviews. Therefore, as noted, the main reason to apply purposive sampling method was to ensure the selection of representative cases, which in this study meant selecting case projects, where ambidexterity could be gained insight.

3.3.1 Agile overview

Explaining to a great extent the characteristics of the case organizations and the actions they take is the agile method they follow in their business operations. Therefore, it is essential to look more closely into the key aspects of agile method and this part will be done in the following section.

The traditional approaches to software development are often based on the premise that the software development process is accurate and controllable. Traditional approaches can be described as plan-driven that emphasize discipline through documentation of milestones, requirement and designs (Napier et al., 2008). Abrahamsson, Salo, Ronkainen and Warsta (2002) note that traditional process-oriented methods rely on the requirements that the software projects are completely locked and set before the actual design and software development takes place. However, this approach might not be suitable when flexibility, adaptability and agile methods are needed (ibid). Here, Dingsøyr, Dybå and Moe (2010) emphasize the agile methods as being a reaction to the traditional plan-based methods, which focus on rationalized, engineering-based approach with comprehensive planning and codified processes. McHugh, Conboy and Lang (2011:503) define agile software development as “a group of agile methodologies that focus on developing software in short time periods”. Khoshroo and Rashidi (2009) conclude that as agile values and principles are based on iterative, learning, interactive and human-centred approaches, they require different kind of project management than traditional and plan-driven
approaches. Indeed, agile software development takes an alternative approach as Moran (2015:12) stresses that an agile team must balance the demand for adaptation against the potential pressure to standardize. Therefore, the agile methods can be considered to provide suitable solutions for fast changing business environments, such as the software industry. Indeed, the popularity of agile methods had increased as Dingsøyr et al. (2010) argue that agile software development has had a significant influence on the way software development is conducted.

The core of agile methods lies in the agile manifesto that consists of central beliefs of the concept (Appendix 1). These beliefs emphasize the individuals and the interaction between them rather than having strict processes and tools on how to guide a team. (Moran, 2015:1) Close team relationships and working environment exemplifies the importance of interpersonal connections (Abrahamsson et al., 2002). In addition, another key objective is to continuously release tested software (ibid) as focus is put on creating working solutions over producing comprehensive documentation (Moran, 2015:1). Customer centricity is essential in agile manifesto as close collaboration is emphasized over extensive contract negotiations (ibid). Indeed, Ramesh et al. (2012) note that agile development relies strongly on constant communication among the team members and the client. Here, the importance of face-to-face interaction increases. Lastly, flexibility is central since responding to change is of high importance rather than following a detailed plan (Moran, 2015:1). Flexibility is of great importance since the teams need to be able to respond to possible adjustment needs that arise during the development process (Abrahamsson et al., 2002). However, even though the manifesto emphasizes the superiority of described aspects, it does not claim that following a more traditional approach are invaluable, rather that more value can be found in the agile approach. (Moran, 2015:235).

The basis for the manifesto can be found in the twelve agile principles (Appendix 1). The principles provide detailed description of the core activities of agile methodology and serve as guidance for action (Moran, 2015:2). The principles represent some key beliefs behind agile organizations as organizations follow these agile principles often develop specific characteristics in their business operations. Common characteristics for agile
organizations are strong emphasis on human aspects through self-organizing teams, intensive communication and learning happening between individuals in these teams. Furthermore, continuous development is realized through short, iterative cycles with close cooperation with the client. This enables the teams to quickly learn and get feedback for their solutions. (Moran, 2015:190)

3.3.2 Common agile methods

Moran (2015:14) points out that there are several well established agile methodologies used by organizations that range from putting focus on product development to project management. Due to the scope limitations of this thesis, two specific agile methods, namely Scrum and Extreme Programming (XP), are shortly presented as they are, according to Hoda and Noble and Marshall (2012), considered to be the most widely adopted agile methods. As the aim of this thesis is not to look deeper into different agile methods rather to shed light into the contextual aspects they entail, the two agile methods are briefly discussed.

Moran (2015:19) describes Scrum as a product development methodology that focuses on the management of software requirements and development. Scrum does not extend to other activities such as business change management, systems development or data migration as it adapts to the organization’s existing practices. Practices within the Scrum framework are established as sprint and product backlogs where increment is produced and released in an iterative way. Scrum entails three different roles; developers, scrum masters and product owners. (ibid) Developers are represented in the project team while the product owner is a representative of the client. Scrum Master acts as a facilitator and oversees the development process.

While Scrum and XP share some common terminology and practices, they differ in terms of structure and philosophy (Moran, 2015:19). Similar to Scrum, XP focuses on project development aspect and consists of interwoven set of techniques the reinforce each other (Moran, 2015:14,16). Here, the adherence to standards and commitment to unit testing, refactoring and integration are emphasized. XP practices include for example daily stand-
up and pair programming. In addition, XP entails roles such as the customer, analyst, programmer, tester, coach and manager. (ibid) In XP, the coaches are responsible for setting direction, aligning people and motivating the team (Hoda et al., 2010).

3.3.3 Case organizations

The first case organization is a Finnish company called Fraktio, which provides comprehensive web service and software development solutions. More specifically, Fraktio offers services for business development, user experience design, software development, managed services and training. Being founded in 2012, the company is rather young, however, it has already established a firm client base. The company employs 20 people, most of which are developers and designers. The importance of client-centric vision is widely acknowledged within the company and the cooperation resided in developing understanding on the client’s business operations and needs. In its business activities, Fraktio emphasizes agile methods, team working, common decision-making and courage to step out of one’s comfort zone. Furthermore, the company has created multiple platforms for knowledge sharing and learning, such as trainings, blog and “Friday presentations”, where both Fraktio employees as well as people outside the company can hold a presentation on a specific topic. (Fraktio)

The second case organization is as well a Finnish web and software development company called Druid. The company was founded in 2012 by five founding partners and their services are implemented on Drupal, which is an open source content management system that is built and developed by the Drupal community (Drupal.org). Druid’s web services include software development solutions and their maintenance. Furthermore, the company provides services within business development, concept building, graphic solutions and training. In addition to client services, Druid has been an active actor in the development of the Drupal software. The company has presence in Finland and the Netherlands and employs nearly 30 people. With strong compliance to the agile methods the company vision highlights trust and freedom and transparent decision-making. (Druid)
3.4 Data collection

The primary data collection for this thesis was conducted through interviews since it is considered to be the primary and most efficient way to collect the information necessary for this sort of research question (Saunders et al., 2012:372). The interviews were conducted in a semi-structured approach with members of four different projects from the two organizations, so total of four interviews were undertaken. Two of the interviews, namely the ones conducted with Fraktio, were done as group interviews with two respondents and the remaining two interviews with Druid were both done with one interviewee.

Semi-structured interviews are an appropriate approach because the authors had an idea of what they wanted to find out and they also allow the conversation during the interviews to be more free and the discussions to develop during the interviews (Miles & Gilbert, 2005). This differs from a structured interview where the questions are determined and are the same throughout every interview (ibid). The freedom to develop the interviews in different ways supports the process to better find out how ambidexterity is achieved in agile software development organizations. Another issue affecting the decision to conduct semi-structured interviews was that the researchers wanted to understand the reasons behind decisions and actions taken by the respondents, and at the same time have the possibility to ask the respondents to elaborate or further explain or build on the answers they provided. The qualitative approach makes it possible to understand the experience the interviewees have in a more profound way, and from that be able to make connections (Miles & Gilbert, 2005).

In addition to conducting interviews, some documents were used as a source of secondary data. These documents included for example company documents, websites and blog posts. Bryman and Bell (2003) suggest that these kind of organizational documents can provide the researchers with valuable background information and in this thesis, organizational documents were used to support gaining a coherent understanding of the case organizations. This suitability of this action is supported by Bryman and Bell (2003)
who propose that in case studies organizational documents can be studied to create a description of the organization and its history.

### 3.4.1 Interviews

The researchers had one interview guide with key questions they wanted to cover in the interviews, although the use of them varied between the four interviews. The interview questions focused and were built around the main themes of this thesis, ambidexterity, agile methods and organizational characteristics of agile teams. The interview guide was roughly divided into two sections, background questions and case project related questions. The former section contained questions concerning the respondent’s position, employment duration in the organization and their position in the case project. The latter section focused more specifically to map the different aspects and the various activities made in case projects. (Appendix 2) In addition, the interview guide contained two finishing questions inquiring the respondents’ views on the software development industry and aspects on what it takes to be a successful software company. As noted previously, the interview guide served as a basis for the discussion in the interviews and not all questions were asked in every interview as some points would arise in responses to other questions.

As pointed out by Saunders et al. (2012:374), the order of the questions can also differ depending on the flow of the conversation. This type of interview allowed an open discussion in order to explore the subjects more thoroughly (Miles & Gilbert, 2005). Indeed, Bryman and Bell (2003) suggest that semi-structured interviews might not follow the intended schedule and alternative or new questions can be included as the interviewer picks up on things said by the interviewee. In this sense the semi-structured nature of qualitative enquiry enabled a degree of flexibility in the interview process as the interviewer could ask additional questions when an interesting topic arose (ibid).

From the total of four interviews two were conducted with one respondent (Druid) and two as group interviews with two respondents in each (Fraktio). Saunders et al. (2012:400) describe interviews that are conducted with two or more people as group interviews. The decision to conduct two types of interviews was made mostly due to the time restrictions. However, conducting group interviews in addition to single interviews gave the advantage
of enabling a breadth points of view to emerge and the whole group to be able to respond to these views (Saunders et al., 2012:402). Furthermore, Saunders et al. (2012:403) state that group interviews might provide an efficient way to interview a larger number of individuals than would possibly be done through one-to-one interviews. As mentioned in the Data collection part (3.3), the participants were chosen with a specific purpose in mind and for many group interviews the purpose originates from the notion that the interviewers feel they can learn a great amount from the chosen participants (Saunders et al., 2012:401). Therefore, from both case organizations the participants were chosen based on the involvement in the projects, which were investigated for this thesis purpose.

Turner (2010) points out three important issues concerning conducting qualitative interviews, namely preparation for the interview, constructing effective research questions and the actual implementation of the interview. Firstly, it is essential to inform the respondents about the purpose of the interview as well as discussing the terms of confidentiality (ibid). Here, regarding the actual conduction of the interviews, the question forms were sent to the interviewees in advance in order to give the respondents an opportunity to get familiar with the main themes of the interview and get a sense of the future interview. However, the interview guides were only sent to Fraktio prior the interviews as the second case organization, Druid’s participation in this study was confirmed close to the actual interviews. The participants from Druid were given a verbal introduction to the topic and questions prior the interviews. Secondly, documentation during the interview is crucial and the information should be, if possible, recorded and written into notes (Turner, 2010). In order to ensure that all important information will be captured, audio recording was used during all four interviews. Saunders et al. (2012:402) state that this tactic is good even if the interview is being recorded as it will allow each interviewer to focus on their tasks. Since there were two interviewers present in two of the interviews conducted with Fraktio, one had the main responsibility in facilitating the discussion while the other was making notes. For the two interviews conducted with Druid, only one interviewer was present, which limited the ability to take notes. Lastly, the process of interpreting the received results is important as researchers need to “sense” what has been uncovered (Turner, 2010) and make conclusions based on the results. This process is described more detailed in the data analysis part (3.4).
As pointed out, the data was gathered through four interviews, which length varied from approximately 50 minutes to 80 minutes. The interviews were conducted with employees that could be considered central for the case teams in terms of having great amount of information and knowledge of different phases, activities and other members in case teams. Each interview took place in the case organizations office premises. The interviews conducted with Fraktio were done in English and the remaining two with Druid were done in Finnish. The initial idea was conduct all four interviews in English, however, the decision to conduct the interviews with Druid in Finnish was based on the preference of the interviewees and interviews conducted with Druid were translated into Finnish later on. All four interviews were conducted around the same time, more precisely between March 21\textsuperscript{st} and 24\textsuperscript{th}, 2016. The first two interviews were conducted with Fraktio, both during the same day. The third and the fourth interview were done with the second case organization, Druid, on separate days. Having the interviews on different days gave the authors the possibility to reflect the obtained data and verify that the applied interview guide and their content were suitable and therefore no alterations were made to the interview guide between the interviews.

<table>
<thead>
<tr>
<th>Fraktio</th>
<th>Case details</th>
<th>Date</th>
<th>Time</th>
<th>Recorded</th>
</tr>
</thead>
<tbody>
<tr>
<td>J.P</td>
<td>Case Auction</td>
<td>21.3.2016, 1h 21min</td>
<td>Recorded</td>
<td></td>
</tr>
<tr>
<td>P.L</td>
<td>Case Auction</td>
<td>21.3.2016, 1h 21min</td>
<td>Recorded</td>
<td></td>
</tr>
<tr>
<td>V.K</td>
<td>Case Research</td>
<td>21.3.2016, 52min</td>
<td>Recorded</td>
<td></td>
</tr>
<tr>
<td>V.M</td>
<td>Case Research</td>
<td>21.3.2016, 52min</td>
<td>Recorded</td>
<td></td>
</tr>
</tbody>
</table>

\textbf{Figure 4. Fraktio interview details}

<table>
<thead>
<tr>
<th>Druid</th>
<th>Case details</th>
<th>Date</th>
<th>Time</th>
<th>Recorded</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.I</td>
<td>Case University</td>
<td>23.3.2016, 51min</td>
<td>Recorded</td>
<td></td>
</tr>
<tr>
<td>M.S</td>
<td>Case Beauty Salon</td>
<td>23.3.2016, 1h 23min</td>
<td>Recorded</td>
<td></td>
</tr>
</tbody>
</table>

\textbf{Figure 5. Druid interview details}

While many advantages of semi-structured interviews have been argued for, some disadvantages exist. Indeed, Saunders et al. (2012:381) emphasize the importance of being aware of the limitations and drawbacks with conducting semi-structured interviews.
The concern about reliability is related to interviewer bias, where the tone or nonverbal behaviour of the interviewer can create bias in the way the respondent respond to the questions being asked (ibid). In order to tackle this issue, “prior planning prevents poor performance” is, according to Saunders et al. (2012:383), the key to a successful interview. Visiting one of the case organizations before conducting the interviews, getting a good understanding from the websites and blogs of both the case organizations, while also having regular communication via email are some actions the researchers took in order to reduce the risk of being biased when conducting the interviews. As mentioned earlier, even though initially contact and communication was established only with Fraktio, the fact that the two organizations are rather similar can be seen decreasing the limitation of Druid not being involved from the beginning.

Moreover, another issue regarding semi-structured interviews arises from difficulties to ensure the coverage of all essential topics. This issue could be seen arising while going through and analyzing the transcriptions of some of the interviews as some topics in the obtained data were at times not relevant for the analysis of this study. However, during the interviews the interviewers did not want to restrict the interviewees in order to ensure that no essential information would be left unsaid. Moreover, at the end of each interview the interviewees were asked if they had some essential issues to bring up that were not discussed earlier in the interviews. Furthermore, especially in group interviews, group effects need to be regarded as a limitation (Bryman & Bell, 2003). On the same note, Saunders et al. (2012:402) point out that in a group interview some participants might try to dominate the interview leaving the other interviewees inhibited and resulting in that they might agree with the others’ views even if they in reality would disagree. This was tackled through testing the emergent views by encouraging all group members to take part in the discussion (ibid) as opinions of both interviewees were clearly asked for. Relating to this issue, the interviews had two participants, which limited the problem of domination from one participants since the interviewer could easily manage and navigate between the two interviewees without feeling overwhelmed by the task.
3.5 Data analysis

After conducting the interviews, the first step in the analysing process was to transcribe all the collected material from the interviews into a Word-document. Also, two of the interviews were further translated into English since they were initially conducted in Finnish. The transcription was done close to the actual interviews in order to avoid a build-up of audio recordings and the associated transcription work (Saunders et al., 2012:550). The transcribing was done to make the following process of analysing the data easier, as Bryman and Bell (2003) exemplifies, it is easier to have the information in a document instead of listening to the audio recordings over and over again.

The data collected in the interviews are not straightforward to analyse due to the volume of data and unstructured material (Bryman & Bell, 2003). Therefore, in order to identify initial patterns and recurring themes from the data, the main findings of each interview were collected in an excel sheet, and this file also further served as a rough summary of all the data. Already here, recurring themes could be discovered.

Through identifying recurring themes from the excel sheet, different categories could be identified as a first step in the analysing process. By identifying categories, the original data could be rearranged in order to provide an emergent structure in which to analyse the data further (Saunders et al. 2012:557). In order to find the appropriate categories for the data analysis, the researchers followed the propositions and objectives of the thesis, namely, ambidexterity, agile methods, exploitation and exploration. The propositions guided the theoretical orientation and in that sense shaped the data collection plan and therefore yielded the analytic priorities on which categories could be develop and distinguish further. (Yin, 2014:136) From this process irrelevant data could also be taken away and the structure of the data could be made more comprehensive (Saunders et al. 2012:557). The next step was to attach all the relevant information from the interviews to the different categories (Saunders et al. 2012:558). Here, both the transcriptions and the excel sheet facilitated and made the process easier. According to Saunders et al. (2012:560), when further searching for key patterns and relationships in the data, new categories or subcategories often emerge in order to refine the focus. Some of the initial
categories were divided into subcategories due to the volume of data and in order to make the process of distinguishing patterns easier.

The stepping-stones for the analysis of the data gained through the interviews were within-case and cross-case analysis. Firstly, the within-case analyses were pursued in order to gain a richer understanding on each individual case (Eisenhardt, 1989). Within-case analyses were pursued first in order to aid the authors to make sense of the volume of data from each case and to gain insight into the initial patterns arising from each case (ibid). Eisenhardt (1989) argues that within-case analysis serves as an accelerator for cross-case comparison, and here the four cases were first compared based on similarities and differences between the main findings and recurring themes. This process further enabled the researchers to make a comparison on a higher level, namely between the case organizations since not only did the interviews provide valuable information about the specific cases, but also about the organizations in general. This is understandable as the vast majority of actions and processes taken in the cases are rooted in the organizational characteristics and thus it was natural that the interviewees reflected back to these issues while discussing the cases.

As the empirical section began to take its final form, it was sent to the interviewees from both case organizations in order to gain further comments and ensure that the displayed information was understood the right way. Based on the comments, some minor changes were made. Furthermore, in order to ensure that no vital information was missed, the transcriptions were examined once more. In addition, this gave the possibility to verify that all information was referred to and explained the right way. Lastly, closer to the final submission, the whole thesis was sent to the interviewees including the corrected empirical section so the interviewees got a last opportunity to provide comments.

### 3.6 Research quality

Van De Ven (2007:12) describes that problem formulation, theory building, research design and problem solving need to be evaluated in terms of relevance, validity, truth, impact and coherence. Firstly, the problem should be grounded in reality relevant to the
considered research audience. To begin with, the discussion on exploitation and exploration has for long been relevant in business literature and studies. In addition, ambidexterity has become rather popular topic in current literature and research. Furthermore, the relevance of the case organizations is important as more and more services are developed as software, by this kind of organizations. Thus, based on the presented issues the relevance of this study is argued for. Secondly, the theoretical model should be clearly expressed and consist of logically valid arguments. This was assured through the construction of the theoretical framework in a way that it begins from core of ambidexterity, namely the competing demands of exploitation and exploration. Thereafter, the framework builds on through presenting and discussing additional aspects to ambidexterity. Thirdly, the research design and the conduct of the research should follow the standards and methods that the scientific community believes to produce a truthful solution. The various methodological possibilities and issues were considered prior conducting the study. Conducting a qualitative research supports the aim to uncover various processes in their natural social setting. Moreover, since the sample was four case projects in two organizations, multiple case study was chosen as a research design, which is often suitable in qualitative research. (Bryman & Bell, 2003) Lastly, the findings should have an impact in advancing science and enlightening the practice in a profession. (ibid) This study advances the ambidexterity literature and research through providing insight into how ambidexterity is achieved in the context of agile software development organizations. As later on in this study will be presented, some new and additional insight is provided to the current literature and research based on the findings of this study.

Furthermore, the research quality can be viewed in terms of trustworthiness. Using the concepts of credibility, transferability, dependability and conformability is a way to ensure the trustworthiness of this thesis and assessing the quality of a qualitative research (Bryman & Bell, 2003). These concepts are argued to better fit a qualitative research in comparison to reliability and validity, which are more aimed to ensure the quality of a quantitative research (ibid) and therefore the four concepts are relied on in this study.
3.6.1 Credibility

Credibility addresses the issue of how believable and congruent the findings from a research are compared with reality, while it also regards how reliable the results are (Bryman & Bell, 2003). According to Bryman and Bell (2003), this can be ensured by looking at how the research was undertaken, by the “good practice” and also by submitting the findings to the initial information provider for confirmation. In this thesis we followed the good practice approach by reviewing and considering previous theory in this field. In addition, in order to increase the credibility, the findings from the interviews were sent to the case organizations for approval and confirmation. Here, the findings presented in the empirics were sent to the case organizations in order to get the approval, avoid misunderstandings and to make sure the information were presented in the right way without the researchers’ interpretations. In addition, before the actual interviews, one of the case organization (Fraktio) was visited and both of them studied through information provided in their websites, which is a way to further increase the credibility (Shenton 2004).

3.6.2 Transferability

Since qualitative research is often deep rather than broad, the question whether the findings can be applied to other contexts than the one being researched arises (Bryman & Bell, 2003). In order to ensure transferability of this study, descriptions of the case organizations were provided. In addition, the study presented the method, namely agile, both organizations apply in their business operations. By providing the readers with this type of information they have the possibility to make judgements regarding how the findings from the research can be transferable or generalized to a broader arena. The readers can also more easily compare both the findings and the concepts to other contexts. (Shenton, 2004) According to Shenton (2004), it is the researchers’ responsibility to make sure enough information is provided so that even though the case is unique, it can be an example within a broader group. Therefore, as mentioned, the context of the case organizations are being presented, the number of organizations taking part in the thesis, where the organizations are located as well as their size. Similarly, background information
about the actual case projects is provided. Furthermore, how many employees were participating in the interviews, which kind of data collection methods were being used and the time period for when the data was collected is also presented. However, still worth mentioning, the results from this qualitative study are best understood in the context in which the case organizations operate (Shenton, 2004).

3.6.3 Dependability

The degree of dependability regards the likelihood of the findings to apply and be the same at other times, i.e. if the research would be repeated (Shenton, 2004). For this thesis, all the information from the interviews, the initial questions, the audio recordings together with the notes, interview transcriptions, e-mail conversations with contact persons at the case organizations, and all empirical material have been saved in order to ensure the dependability of this research. With the help from that material, a similar research can be conducted. To further ensure the dependability of the study, the research design and how the study is being implemented is carefully described as suggested by Shenton (2004).

3.6.4 Conformability

Conformability depends on the level that the authors have allowed their personal values or opinions to sway the outcome of the research (Bryman & Bell, 2003). The importance is to ensure that the results and ideas in the research come from the case organization and the people being interviewed, rather than from the preferences of the researchers (Shenton, 2004). In order to minimize the conformability, the decisions and methods being used in this thesis are argued for. It should also be noted that a discussion concerning limitations of the choice of research, methodology and the findings will show the reader that the researchers have a critical mind and an awareness of the limitations.
## Methodology summary

<table>
<thead>
<tr>
<th>Research Topic</th>
<th>Ambidexterity in agile software development organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research Design</td>
<td>Exploratory study with a qualitative method</td>
</tr>
<tr>
<td>Research Approach</td>
<td>Combination of deductive and inductive approach</td>
</tr>
<tr>
<td>Research Strategy</td>
<td>Multiple case study</td>
</tr>
<tr>
<td>Research Context</td>
<td>Purposive sampling: Fraktio and Druid</td>
</tr>
<tr>
<td>Data Collection</td>
<td>Semi-structured interviews and group interviews, conducted face-to-face</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>Transcription, within-case and cross-case analysis</td>
</tr>
<tr>
<td>Research Quality</td>
<td>Trustworthiness: credibility, transferability, dependability and conformability</td>
</tr>
</tbody>
</table>

*Figure 6. Methodology summary*
4. Empirical findings

This chapter provides descriptions of the empirical findings that were gained through the interviews with case organizations. Total of four interviews were conducted with key individuals from four different cases, two from each case organization. The conducted interviews were done using the same interview guide in order to gain an understanding on the main issues and processes in each case and to provide a basis for analyzing similarities and differences between the cases. As mentioned earlier, the level of analysis will be on the case level, however, as much of the data reflect and stem from aspects from the organizational level, the findings are further reviewed on the organizational level as well.

This chapter presents each case through a brief introduction followed by more extensive reviews. These reviews follow the main themes that arose during the interviews. In addition, since a great amount of the data reflected the agile principles, they were used as a guiding structure in the presentation of the findings (Appendix 1). Agile methods and organizations as the research context and the case organizations were presented more extensively in the previous chapter and the reader is encouraged to go back part 3.3 if information about the case organizations is required. In order to follow the required anonymity, all cases are referred by altered names. Moreover, limited information of the clients with whom the case organizations worked in the cases is provided to protect confidentiality, however, sufficiently enough to describe the context in which the cases were conducted. In addition to changing the name, in Case Beauty Salon the industry has also been altered in order to ensure additional anonymity.

4.1 Fraktio

The following sections provide short descriptions of and information about the case projects that were examined in this study. The displayed information presents the purpose of the case projects, duration and team structure.
Case Auction

Case Auction involved the development of an Internet based auction service platform. On the platform, different actors, such as companies and consumers, are brought together to make purchases. Often the companies represent the selling party and consumers the buying party, however, companies can act as the buying party as well. The items that are sold in the auction platform vary from cars and apartments to smaller items such as office interior equipment.

Case Auction was initiated at the end of summer 2013 and during the following autumn a team from Fraktio took over the project. The case begun with an introduction phase where the two parties, Fraktio and the client, got to know each other and as the team took over the project, they were given the full responsibility over the whole platform; its maintenance, programming and development.

The team structure has evolved during the path of the project. In the beginning of the project, the interviewees, J.P and P.L, were the only ones working with the project from Fraktio. Currently, three employees from Fraktio, mainly software developers, are working with the projects, however, some other employees, such as designers, have participated in the project in some tasks. In addition, from the client side, two people, namely the owners of the project, have been actively participating in the project work and as the project proceeded the client recruited their own software developer and in general have more people working with the case from their side. Also, as the project has evolved, the client has established a customer service team and sales representatives.

Case Research

Case Research consisted of a pre specification project where the team conducted a screening for the requirements of a future project. The client was a governmental research organization that wanted to integrate all test data created in one of their laboratories into one place. Earlier, all the test data was stored in multiple places but in order to reduce errors, the data were to be stored in one location. Here, the team's task was to identify all
the essential factors, which had to be considered when developing a solution for integrating the data and define what kind of system was needed to solve the problem. The task was special in the sense that the laboratory, the tests conducted in the laboratory, as well as the data were under many strict regulations that had to be taken into consideration when developing a suitable system, such as the requirement that the data had to be stored for at least 50 years.

The project began at the end of 2015 and lasted for approximately two months. Case Research was somewhat special in the sense that the company had not done this kind of specification projects previously and that its scope was rather large. Furthermore, unlike most projects, Case Research did not include any actual software development as the work served more as a basis for the future software and system development.

The team consisted of two employees from Fraktio, namely two user experience designers. The amount of people who were involved from the client side changed during the course of the project and the people represented different parts of the organization, such as IT and the actual laboratory employees.

### 4.1.1 Being a successful software development organization

Interviewees from Fraktio explain that they differentiate themselves from their competitors in the way that they are currently focuses on web solutions as some competitors offer wider solutions such as embedded systems. Furthermore, J.P emphasized Fraktio’s values, structure and ways of working being different from the corporate mindset that some competitors possess. Regarding being agile, J.P pointed out that many of their competitors’ state that they operate in an agile way with their clients, yet they are not agile internally. On the same note, V.M noted that some competitors frame themselves as being agile, however, that does not necessarily reflect what happens in everyday work.

Reviewing employees as a valuable resource also arose during the discussion. J.P specified that organizations need to hire people with “… growth mindset, this constant improvement mindset…”. Likewise, V.M mentioned that one needs to have good people,
treat them well and be able to retain them within the organization. In addition, it is important that these people enjoy the journey of growing and improving. Overall, J.P and P.L highlighted the importance of a good work atmosphere both within the organization and with the clients. On further notice, V.K noted that “staying ahead of your time” is important as the industry is rapidly changing and thus requiring that “… you just have to be learning stuff all the time.”. “Making happy customers” was also mentioned by J.P as a crucial aspect to success. P.L added that it is of great importance that the client needs are prioritized and that no unnecessary activities or software are developed. Moreover, constant development was emphasized as being mandatory in the industry. When asked more specifically how the interviewees saw this constant development being realized, they described one essential issue as seeing projects as independent and not trying to apply the same tools for all projects.

### 4.1.2 Client relationship

The importance of the relationship with the client was highlighted in Case Auction as P.L expressed that “… trust is mandatory, otherwise we couldn’t get anything done […] that’s always the default…” (P.L). The extent to which creativity could be pursued was said to depend on the client and in the Case Auction, the client was highly flexible and open for experiments. The high flexibility from the client came from “…a kind of trust system” (P.L) which further enabled the team to act in a flexible manner. Yet, the client relationship for Case Research differed from Case Auction. In Case Research the client relationship was described between the interviewees and the end users, who the team interviewed for the specification, not that much between the team and the owner of project. Also, the fact that the client was a rather traditional governmental organization created some difficulties as explained by V.M, “…the biggest problem in this project was to kind of like force them to be agile but then again questioning ourselves that can we be agile in this project?”. Further explaining the specific characteristics of the project and the relationship V.K noted that “…you need to have trust if you’re going to code and use Slack with a client…”. V.K further added that this was a new client and in this context it did not make sense to first establish the needed level of trust and then use Slack for this short period of time. Moreover, since Slack is often utilized when actual software is developed, which was not done in Case
Research, establishing the Slack with the client was not of importance. However, both V.M and V.K agreed that the client provided a flexible way of working and V.K further explained that it was easy to arrange meetings and that everything with the client went “…quite nicely”.

4.1.3 Organizational environment

The Case Research was described as unique by the interviewees, mostly due to its scale and the field the client was operating in. The field especially posed some specific requirements for customization due to the vast amount of regulations concerning the client’s business. When asked about the support from the organization, the interviewees felt free to challenge existing ways of working and experimenting and both interviewees felt encouraged to both undertake experimentation. V.M emphasized that “…felt okay to try something new.”. Similarly, V.K explained that “I think we had free hands to do basically anything but it was kind of like common sense to not push it too much.” The use of common sense guided the team to keep the experimentation to the needs of the project and client’s business, but the team was somewhat restricted to experiment in terms of technological solutions as the suggested solution needed to be reliable and proven to work. For the interviewees, it was the first time they worked together in a team, and they emphasized complete trust for each other, however, the characteristics of tasks and the overwhelming amount of information they had to go through and analyse also created some frustration.

Somewhat similar situation regarding support occurred in Case Auction, as P.L explained “…if it’s good for the project, for the customer and so on, we can basically do what’s best so yeah, we can quite freely choose how we work…”. When further asking about creativity in the project, J.P expressed that “And if you mean creativity equals improvisation or that kind of thing, we believe in creativity and improvisation more than like strict planning, let’s say, quite heavily and then do more in like “plan little, do little, plan little, do little” kind of approach”.
Even though the organizational support towards experimentation was mentioned to be high, when asked about the degree of creativity the team members in Case Research could apply in their work, V.M described that “… not so much because it was like mostly documenting facts.”. In addition, the client’s business environment posed some obstacles for being creative as V.K explained, “Because it was specification so it was kind of like completely …] like totally new field and like everything was like “wow oh it works like this” so I can’t really come up with any new stuff how to do that instead.” However, when the actors on the client side could not come to a conclusion or had competing views, the team served as a unifying actor. The fact that the team was acting as an objective, unifying actor V.M described as, “Then I feel a little bit kind of creative, in a different way”. Here, according to the interviewees, creativity was realized as “… problem-solving and how to communicate things clear…”.

The environment at Fraktio also encourage to consult other members in the organization, this due to their expertise or to merely get a “fresh pair of eyes” and feedback. This was explained by J.P as, “… you might even do consulting on some tool or library…”. P.L also added that “… we can ask feedback from people who aren’t directly working in this project team.” The importance of this kind of “consulting” was expressed by P.L as “… there are new technologies coming every week and so on, and we don’t have really formal […] workshops or something, we don’t buy education from outside, but then you learn when you are doing the work and some project team might use some new technology and then we can utilize it somewhere else so we have these connections that team have, we learn by doing...” and “… we want to like create a structure where people are together…”. Common decision-making reflects the cohesiveness of the team as P.L exemplified that “… usually when we make a decision we go it through (with) the team first…”. Common decision-making was found especially important if the decision was to have an effect on everyone else. However, the team also had some “… high-level guidelines, for instance how the code should look like…”, so that the team members could make individual decisions as well.

In Case Auction, how the team was constructed was described to be driven by the interest the employees have towards the upcoming case work. In addition, the availability of the
employees plays a role in who is assigned to work in the case project. Reflecting the same aspect, in Case Research, how the members ended up working with the case was described to result from expertise.

4.1.4 Communication and information flows

At the beginning of both case projects, an introduction phase was put in place. In Case Auction, during the introduction phase, the team got to know the client and vice versa. Similarly, in Case Research, a kick-off workshop with the client was done. The idea behind the kick-off workshop was described by V.M as “...so we could get the whole picture [...] goals [...] and make them think about risks in the project [...] and what they want from the system...”.

In Case Auction, even though P.L described that “... we don’t have that many processes that usually are the same for every project...”, there were few items or processes that are used in every project, one is Slack, a real-time online communication chat, and one is co-reviews. Slack channels were established both for only team members from Fraktio and also for the whole case team including members from the client organization. The idea behind the co-reviews is to support and increase collaboration and that no one is the sole person to see the code and put it to production. Conversely, Slack or co-reviews were not used in Case Research. As V.M pointed out, Slack is often used when a project is ongoing with continuous development. Instead, in Case Research, communication with the client was mostly done by face-to-face meetings as the team interviewed different actors who were involved from the client side.

The team in Case Research gathered a lot of information from different actors in the client organization, and they faced the challenge of deciding which could be the most relevant and essential for the final solution, V.M explains “... just deciding what was relevant and not [...] what am I learning here...” and V.K further adds “...we can get all the information together but what is actually meaningful...”. Both interviewees felt that creating a relaxed environment where everyone could share their thoughts was important in order to get a coherent understanding of all aspects of the project. The team members described
tackling these challenges with long conversations with the different actors. Furthermore, as both team members made own notes and assumptions during these discussions, they could compare and use those as basis for further conversations. Going through the documentation and cross checking notes and observations was an important step for the interviewees, V.K explained that the team members had observed some issues differently and thus this kind of cross checking activity played an important role in assuring key aspects and learning from each other. The case team could also take advantage and gain knowledge and information from other members at Fraktio when consulting coders for their expertise.

The importance of communication was further emphasized in Case Auction since it was seen as not only a part of the collaboration, or a result of it, but also as a precondition. As P.L explained, “... sometimes the knowledge you need to do, that you need in order to get the work done might be really broad so you just might not have all the information you need so you must be willing to also get help if you need.”. J.P agreed to this by stating that “... what we are doing is that we are constantly talking a lot, like we said, but encouraging people to take these steps that when you need help.” P.L explains that when decisions being made, they are being discussed within the team, and the same if the decision is affecting another employee, it is natural to discuss the options internal first and later present the final options to the client.

4.1.5 Keeping it simple

P.L pointed out during the interview that if there is existing code that can be used for something else, there is no point in develop a new. The interviewees in Case Auction described the project as a “brownfield” project, meaning that there was already existing code and established systems and platforms when the team took over the casework. Differing from a “greenfield” project, where the team would create a new platform and site, however still using existing code storages, in a brownfield project you develop the existing platform and site. In the beginning of the project the team working with Case Auction had to rely on existing material, e.g. systems and platforms, as the client had bought the service from someone else. The interviewees stated that problems which arose during the
project were not of completely unique nature as P.L explained “... these are something we can see beforehand if we have knowledge of these...”. More specifically, problems with hardware and server related issues were somewhat common and were such that the team “… knew how to go with those when they arrive…” (P.L). Overall, reactions to solving the problem depended on the criticalness of the problem as P.L described that “... if it’s something really critical [...] drop all the work we are doing instantly and start resolving these problems…” and “… if it’s critical usually the whole team shifts the focus on the problem.”.

The purpose of Case Research was the initial specification, which could be used as a basis for the actual future project, where the actual system would be developed. In order for the technology and data to last the required time period (50 years), the team concluded they should not look for new experiments, rather apply existing technology that is already known to be working. Overall, the tools that were applied in Case Research were described to be “… quite traditional…” (V.M). Also, doing user testing, i.e. interviewing the future users of the system, without actually developing software at the same time was described to be “… a common procedure even though we don’t do that so much.” (V.K).

When asked more specifically about the extent to which existing capabilities or resources were applied, P.L in Case Auction described that all the existing skills and expertise that the team members possessed were of high importance. In addition the application of certain tools was described to be affected by “... certain opinions and certain best practices...” that the team had. Indeed, in the software side, existing repository management, i.e. a storage location for software packages, was applied, and the team was also able to use existing libraries, i.e. storage location for code, processes, scripts and classes. Furthermore, a common deployment tools was used in Case Auction. This was emphasized by P.L, “... we can use the libraries but that’s basically it, everything else we have to customize.” Furthermore, the team did not follow any official project management system, however, some project management related issues from previous projects were applied in Case Auction as well. P.L noted that “... we took the good parts from here and there and tried to little by little build our own project management flow”. He further explained that the team seeks to gather all the good parts from different methodologies.
without fixating on only one. This was mostly due to the fact that there are always different factors in different projects and thus one tool cannot be applied in every project.

### 4.2 Druid

The following sections provide short descriptions of and information about the case projects that were examined in this study. The displayed information presents the purpose of the case projects, duration and team structure.

**Case University**

A university organized a competitive bidding in 2013 in order to find a software provider to develop the university’s web site and platform. Case University was initiated in the end of year 2013 as a result of winning the competitive tendering, however, prototyping different software solutions began six months before the actual bidding took place. Following agile software development, Scrum was applied in the case as a system of project management. In addition to the framework Scrum provides, the team developed some additional processes and activities that were found good in supporting the common Scrum activities.

The team structure has changed as the case proceeded. Currently, Druid has four active employees working with the case. In the beginning, as the work was mostly consulting, one employee from Druid was actively working with the case. As the project proceeded, new employees were brought into the case, however, the maximum amount of employees working with the case simultaneously still being five employees. Yet, in different phases of the case approximately 8 individual employees have been circled in the case work. In addition, from the client side, employees have been closely involved in the case work. The client had their own proxy PO, product owner. Moreover, the case was appointed a steering group that included different decision makers from different organizations within the client organization and the interviewee A.I from Druid.
During the first six months of the project, basic implementation was produced through a pilot that was done with the University Magazine. Tasks in this implementation phase varied quite a lot, however, developed material was published and released quickly. Following this, the team began building the actual corporation site and after reaching a stage when the site could be published, they continued to develop homepage machine type of solution. Here, the creation of sub groups under the same site for all the different sub organizations was possible. The developed site was built on Drupal, which is an open an open source content management system that Druid applied in software development. More specifically, Case University was built with version 7 of Drupal.

**Case Beauty Salon**

Work in Case Beauty Salon is a result of a long client relationship. Case Beauty Salon has included multiple different projects, however, all of them have been connected in one way or another. The client is a Finnish company providing beauty services and the aim of the overall project was to develop their web site as well as a web appointment booking site. A part of the overall project was to develop the interface between the different sites and the user information system. Furthermore, another sub-project was the development of a web appointment-booking site for the client’s subsidiary, which provides beauty services specifically targeted for children. Lastly, Druid has also developed the client’s internal web site. Similar to Case University, the sites were built Drupal, which is an open an open source content management system, however, here the sites were built with version 8 of Drupal.

Case Beauty Salon was initiated in spring 2013, however, one of the team members, who is also the CEO of the company, has worked with the client during his previous employment. The team structure has evolved during the years and the number of employees on the team has been around 2-4 persons, on average 2,5 to 3 people. Currently three employees from Druid are continuously working with this project. Aspects making the teamwork a bit special in this project are the fact that one of the team members is working remotely from Bangkok and another one is working while travelling around the
world, while the remaining team member is working at the office. The team applied Scrum as project management system for Case Beauty Salon.

4.2.1 Being a successful software development organization

Not reflecting the case specific aspects but a higher-level view, one key aspect that arose when discussing the main issues relating to success of software organizations was understanding the client. Indeed, A.I emphasized that as the biggest issue regarding being successful. Similarly, M.S highlighted service attitude as being most important aspect. Furthermore, he explained that one aspect of success originates from successful projects and that often the factor behind successful projects might not be whether or not you are technologically qualified enough rather that you are able to fulfill the client needs. In addition, being skillful in selling was also considered essential as good selling skills compensate perhaps less adequate competence. He further exemplified that some companies might be good in selling even if they might not possess the best technological competence. On the same note, M.S emphasized the ability to convince the client and getting quickly familiar with the client’s business. Another aspect that arose was the size of the organization as especially it might be a significant factor for clients when they choose whom to cooperate with. Comparing Druid to its competitors, A.I highlighted that applying agile methods gives an advantage to the company. He further pointed out that some companies say that they work in an agile way but in reality only few does. It was considered important to follow the agile practices, such as Scrum, but also be able to build additional processes on top of the framework and this was something A.I considered Druid been successful in. In addition, being bold in their opinions and not being afraid to say no is something that A.I considered as important. This was also brought up by M.S as he explained that “... important thing is that we do not do what the client wants...”, referring to that the employees are encouraged to be open if they consider an idea or solution not suitable. Strong presence in the Drupal community and taking part in the development of Drupal are also factors that were considered to give Druid a competitive advantage. This was supported by M.S who also described the strong involvement in developing Drupal and the company’s good reputation there as key aspects.
4.2.2 Client relationship

While discussing the relationship with the client and the trust between the two parties, M.S described an event in Case Beauty Salon after the team detected potential a bug that might cause a serious gap regarding information security, they informed the client right away. Considering the issue with seriousness and in an open manner resulted in that when the client was asked if he had completely lost his trust for the team the client replied that “...no the opposite, the trust is higher than he ever could have imagined...” (M.S). M.S further explained that “... if we want to test something we don’t even need to go that much back and forth...”, implying that the team could apply and test new solutions and ideas in the case work without necessarily running it through the client. An additional aspect exemplifying the level of trust in both case projects is the simple fact that Druid implements all its software projects with Drupal. As described earlier, Drupal is an open source content management system, meaning that new solutions are published in the Drupal community, even if they were developed for a specific client. Therefore, basically anyone who is active in the community and using Drupal, including potential competitors of the client, could make use of the possibly novel, developed solutions. This side of the Drupal system was described sometimes being intimidating for the client, however, as the team explained and exemplified the different aspects to positive issues of Drupal, the client felt more assured. Therefore, the team’s ability to emphasize the advantages of Drupal and, how the interviewee described it, “... push it through the client organization...” displays how the client trusted the team and its professional judgment in issues regarding the use of Drupal. Additionally, in Case Beauty Salon, building the site on a new version of Drupal that had not been released yet exemplified the trust between the client and the case team. Similarly in Case University, trust between the client and the team was also considered as high. This came clear for example when the team suggested something that would require that the team “... went on a completely new trail...” (A.I) even though this new trail was far from the original premise that the team was set to work with. Here, the client’s trust towards the team to make the right decision resulted in a better outcome as changing courses was described to be “... completely right solution...” (A.I) in the end.
Communication and interaction with the client was frequent in both cases. Frequent communication and interaction between the team and the client, which is a characteristic of agile software development, were also considered to be important regarding the casework. The strong emphasis on openness enabled that the client relationships developed to be strong and ideas and obstacles could be freely shared.

4.2.3 Organizational environment

Case University separated itself from previous projects since they used an external Scrum coach. Furthermore, even though applying Scrum as a methodology was not by itself new, but in previous projects the organization had not developed processes on top of Scrum as strongly as in Case University. A.I explained this further as, "... we have a good foundation but then that we are able to build on that foundation something else...". A.I further elaborated that in practice, it is extremely important what you can build on top of Scrum, and that is something they have been successful in. Another new activity was described to be the application of a media management system to Drupal. The implementation of the media management system was not only new to the team but also in the whole Drupal community. Reflecting the importance of the media management system was when A.I noted that its implementation was in a way the issue that initiated the whole project. Furthermore, application of new tools arose in terms of comprehensive testing, which had not been done in this particular way before.

When discussing the degree of creativity that could be pursued in the casework, the importance of the client relationship arose. The client of Case University enabled the team to create space for creativity and experimentation. This was explained by A.I more specifically as, "... they understand really well the testing and like doing experiments or stuff like that...". A.I further explained that experimentation and creativity could be divided into technical and process sides. On the process side, the team was able to test various different solutions as they could “...make some kind of solution or a “might-work” solution and see if it flies...” (A.I). Application of these kind of “might-work” solutions was done quite often even though the final result was unsure. Similarly, on the technical side, the team was able to apply some new solutions. This was described to be due to the scope of
the project but also due to the client who gave strong support so it has, according to A.I, “...been up to our interests or skills how we want a thing to go, so we have been able to learn new stuff and also rethink the blocks we have used”.

Reviewing application of novel processes and experimentation on a broader scale, A.I described that since the case organization is still relatively young and has not found the right way or structure, the organization is building the structure on experimentation. Therefore, the emphasis that the case organization gives to experimentation is rather strong. However, even though great amount of trust towards the team members is given, it does not come without some expectations. A.I described that in Case University, if someone is to implement a new solution, trust is given that the person has done the background work and knows what would be the best. A.I further exemplified that “…if someone is given a task, it might be that there is no clear picture how it should be done so basically the work that you have to do is the detective work before hand, this should be done like this and so on”.

Overall, Case University was described to be the case organization's “favorite”, implying that the case has been significant for the whole organization in the sense that it could be used to both apply solutions, which were learned and proven good and also get the chance to test and experiment solutions that might be working. Subsequently, the solutions applied in the case could be applied somewhere else as well, as Case University gave opportunities “…if there has been something that we want to test then we can try it here in our project and then see how it works…”. (A.I)

Case Beauty Salon was described to be “… in a way similar in a way different…” compared to other projects. One reason for this was that the team used some other technologies than they would normally use, however, these technologies did not play remarkable role in the work. M.S characterized Case Beauty Salon as involving plenty of experimentation, “… if something there has been a lot of experimentation...”. One of the major experimentation was the implementation of Drupal 8, a new version of the open source content management system that Druid uses in all their solutions. At that time, Drupal 8 had not yet been published in the Drupal community, meaning that there were
some parts of the software the team could not rely on from the community rather they had to build them themselves. The decision to implement the site on the new version was a result of thorough investigation and testing in order to mitigate potential risks. Exemplifying the emphasis that the investigation and testing were given is that during one sprint the team did “... small fixed but most of the time we used to experiment...” how various elements could be solved in the new version. Implementation of Drupal 8 was important in the sense that it would strengthen the company’s position as a forerunner and also bring the release of the new version closer for the whole community as everything the team developed was put to the community for other Drupal users to apply. Furthermore, with the new version some elements of the software could be done more easily. In addition, through the application of newer technology and service, the client benefits from its longer life cycle.

In terms of experimentation, the team was able to use their creativity in Case Beauty Salon. M.S highlighted this by stating that “... in every freaking place we try things...”. This was greatly a result of the high level of trust the client had for the team as well as the organizational culture. Highlighting the strong trust is an event when the team informed the client that they were about to try out a solution, the client’s reply was as follows “... why are you even telling me this, go and try...”. However, the extent to which the team members pursued creativity was often reflected back to the case and to the actual business benefit the new solutions would bring to the client. This was explained by M.S as, “... at some point I was even surprised that a developer, they basically have free hands to do anything, but they still think about like, actually if we try this will it ever be beneficial for the clients business, “no”, technically this would be an awesome thing but okay, even if it would succeed or not, it would not change the client’s business in any way so it can’t be applied here...”. In addition, in Case Beauty Salon experimentation arose in problem-solving. This was exemplified when a potential bug in the software emerged and the bug was caused by some old software that had been left in the system and was not done by the team members, it thus arose as an unexpected and unique issue. When this kind of critical problems arose, sole focus was put on them and “... everything else is stopped...”. 
As noted, on a broader level, the interviewee states that the organizational culture supports creativity and experimentation as M.S notes that “... it’s easier to be forgiven than get a permission...”. The team members were encouraged to take initiatives and look for novel solutions as M.S expressed that the team members “... do not need to ask permission...” if someone wanted to bring up and test a new solution. When explaining about the freedom that the team members had in terms of creativity and experimentation the interviewee stated, “... when they (team members) come and ask something from me, like “here are the options” and if he presents 3 options and I ask what do you think is best, it really does not matter which one he says, then I just say yeah that one, that one I would have picked too...”. The importance of this kind of approach to experimentation was further described by M.S “... I want people to make their own decisions [...] the hardest part is that even if you know that definitely not like that and you say it out loud then the ability to make own decisions falls completely apart...”.

For Case University, trust was especially emphasized within the team and described to be one of the team's strengths as A.I explained, “...we are stitched together quite strongly [...] within the team we have not had to think about if someone does what he/she is told”. A.I further added that trust arises in the sense that the team members trust when someone implements something, they have done the background work for it and they are the right person to say if this is the right way to go or not.

In both cases the role division in the teams was described to be established quite naturally. Here, M.S highlighted that the roles formed naturally as the expertise of each member drives the exercised role, in addition to their interest in specific tasks. Similarly, A.I described that “… there are sort of these natural roles...”. An important issue was concluded by M.S who noted that “… people are at their best when they work with what they want to work with...”. Moreover, both noted that the roles in the team can be described to be somewhat interrelated. M.S described that the roles and the tasks interrelate quite a lot generally, however, mostly on a higher level as it often relates to integration of back-end and front-end aspects. A.I. stated that approximately half of the team can perform tasks within the same know-how and perhaps one outside that know-
how, however, it might not be favourable because this kind of rotation might affect the pace of the work, which is of high importance.

4.2.4 Communication and information flows

In Case Beauty Salon taking advantage of the whole organization as sources of ideas and help was found helpful. This aspect was further explained through the characteristics of agile software development as M.S stated “... agile development is built purely upon openness, nothing would ever work out without that ...” If the team faced difficulties with a task they know someone else had done in another project, they could go and discuss it with them. The interviewee explained this by stating that “… you have hit your head against the wall for a long time and then you talk to someone and then you get it done in an hour…” (M.S). Here, working at the same office environment was of great importance as M.S explained that “… if the client has a team of four and one guy sitting here then they basically have all 25 employees and their knowledge at their use...”. Indeed, the channel that was considered to be the most important was interaction and interaction in the same physical space. More specifically, A.I emphasized that the biggest channel of communication is face-to-face interaction and discussions. On the same note, M.S concluded that their “… culture would not be anything if people are not here...” and thus highlighting the importance of interaction within the organization. In both cases, chat system called Flowdoc was used for quick online communication both within the teams and with the clients.

Knowledge and experiences about new technologies were shared through phrases such as “Hey, this is like this…”, “I am testing this…” and “By the way this does not work like this…” (M.S). Through this type of notions, the employees shared valuable information and knowledge between each other and learned something new they could apply to their respective work tasks. Furthermore, in Case Beauty Salon trial and error type of learning was emphasized as M.S explained “… we have discussed that you should make mistakes because that’s how you learn [...] we aim to support that...”. The interviewee further explained that “… like if a testing has gone bad then we do not need to try it again and think [...] if this could be better if we did it like this…”, displaying that if an experimentation
of a novel solution failed, the team could quickly bounce back from it and learn that it was not a suitable solution.

Similarly, in Case University, A.I strongly emphasized that exchanging information and knowledge was a way to learn from each other. As a form of knowledge sharing and learning, A.I pointed out that “cross-used” different channels of communication quite a lot, he mentioned they have their own internal chat channel for this specific project, an official chat channel where the client is present as well. The team used a work control system called Jira, where different tasks, their status and who was working with the task were viewed. The system was in a way unofficial “steering” the work as A.I described that “... they are that kind of steps that are in a way in our back bone...”. One step in the work control system was peer review, which meant that someone else goes through and checks the code and its functionality and this was often done by sitting together by a same computer and going through the task and the code. This kind of cooperative activity can be seen as a form of learning as both members had to explain what had been done, especially if the task was complex. Furthermore, Druid has implemented a concept called “Starplayer” where one team member is appointed as the “go-to” person when you need information. Being the Starplayer does not mean full responsibility of the case work rather than having an overview what is happening. Starplayers of all on-going cases had close communication as they had a meeting once a week where they went through their cases and discussed possible problems or failures and also successful solutions.

4.2.5 Keeping it simple

In Case University, regarding the use and application of existing technologies and tools, there were quite a lot of basic implementation with the usage of Drupal, where the team could apply existing components. Reliance on existing technologies and systems was in this sense regarding as a positive issue as it enabled that when some components were needed, the team already knew which “… direction…” to take when applying them (A.I). A.I further explained that the solutions that were suggested were partly based on experience. When asked about the need for customization in Case University, A.I explained that the team was able to work with a somewhat generic methods and
components and could apply what A.I referred as, “… lego blocks…” that they already had. A.I further explained that this was found as a favorable aspect since that way they did not “… always need to “invent the bike”, sort of simple thing over and over again when you start with a new project…”. Furthermore, in Case University, like in other projects Druid did, Scrum was applied as project management system. Likewise, in Case Beauty Salon the team was able to rely on existing programs and processes. In addition, the actual web platforms the team was developing already existed when the team took over the project and therefore the team began working with established systems.

In Case University, the team applied Scrum as project management system, similarly as the organization applied it to all of its projects. Therefore, as case organization had applied Scrum in other projects, thus the framework and its application were familiar to the team. Like in Case University, in Case Beauty Salon the team followed Scrum platform and the application of Scrum can be viewed as an existing process as the organization had experience in using Scrum as an agile project management system. The Scrum framework consists of sprints were defined tasks are performed and a part of existing activities is producing weekly reviews of the sprints as a way to summarize taken actions and gain feedback. In addition, in Case University rotational work was applied in the Scrum Master role as different team members were circled in the role. The fact that the team circled team members in the Scrum Master role gave all the team members the opportunity to view the project work from a different perspective and that way learn new aspects. The role of the Scrum Master is to act as a coach or a facilitator for the Scrum team.
# Figure 7. Summary of the empirical findings

<table>
<thead>
<tr>
<th>Environment</th>
<th>Communication and Information Tools</th>
<th>Client Relationship</th>
<th>Organization Development</th>
<th>Being a Successful</th>
<th>Case</th>
<th>Research</th>
<th>University</th>
<th>Druid</th>
<th>Fraktio</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Improving the working environment in the organization</td>
<td>- Communication support and sharing</td>
<td>- Improved role and responsibility</td>
<td>- Strategy development</td>
<td>- Being a successful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- More frequent and better communication</td>
<td>- More frequent and better communication</td>
<td>- Improved role and responsibility</td>
<td>- Strategy development</td>
<td>- Being a successful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Improved communication and collaboration</td>
<td>- Improved communication and collaboration</td>
<td>- Improved role and responsibility</td>
<td>- Strategy development</td>
<td>- Being a successful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Improved communication and collaboration</td>
<td>- Improved communication and collaboration</td>
<td>- Improved role and responsibility</td>
<td>- Strategy development</td>
<td>- Being a successful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Summary of the empirical findings**

- **Environment**: Improving the working environment in the organization and more frequent and better communication.
- **Communication and Information Tools**: Communication support and sharing.
- **Client Relationship**: Improved role and responsibility.
- **Organization Development**: Strategy development.
- **Being a Successful**: Being a successful development.
- **Case**: Druid Fraktio.
- **Research**: University.
- **University**: Research.
- **Druid**: Druid.
- **Fraktio**: Fraktio.
- **Action**: Case.
5. Analysis

In this chapter the empirical data and main findings obtained from the case organizations Fraktio and Druid will be analysed. The analysis is conducted through reflecting the findings to the theoretical framework. In order to provide an answer to the research question, the main goal is to gain insight into how agile software development organizations achieve ambidexterity.

In order to grasp how ambidexterity was achieved in the case organizations, this chapter will describe the exploitative and exploratory activities taken in the case projects. Moreover, the empirical findings are applied to display what approach to and type of ambidexterity agile software development organizations pursue. Lastly, this chapter will present and discuss a model that further illustrates the main findings of the analysis and the relationships between them. In addition, the chapter will end with a discussion of a new perspective in reviewing agile software development organizations as ambidextrous communities.

5.1 Exploitation and exploration

Prior research and literature suggest that exploitation and exploration are distinctively different activities in terms of the architectures and processes they require (Smith & Tushman, 2005). However, in order to be successful and have sustained performance, organizations need to balance both activities, i.e. become ambidextrous. In order to grasp how agile software development organizations achieve ambidexterity, the empirical findings are analyzed in order to distinguish the exploitative and exploratory activities taken in each case and further in each case organization. These activities are regarded in light of Eriksson’s (2013) definitions, that is, exploitation is application of existing knowledge and technologies and exploration is based on new knowledge and technologies. In addition to identifying these activities, this section discusses how does the
application of agile methods in software development organizations affect how they exploit and explore.

The exploitative and exploratory activities are summarized in Figure 8. The exploitative and exploratory can be illustrated through the technological and process perspectives as many of the activities can be demonstrated through the perspectives. The activities in the table are further explained in the following sections.

<table>
<thead>
<tr>
<th>Fraktio</th>
<th>Druid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Auction</strong></td>
<td><strong>Research</strong></td>
</tr>
<tr>
<td><strong>Technology</strong></td>
<td>- Existing site and platform (brownfield) - Code, software, libraries, repository management tools, deployment tools</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>- Well-proven project management processes - Peer review - Slack</td>
</tr>
<tr>
<td><strong>Exploration</strong></td>
<td><strong>Technology</strong></td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>- Unique problem-solving</td>
</tr>
</tbody>
</table>

**Figure 8.** Exploitative and exploratory activities in the case projects

### 5.1.1 Exploitation

As noted, exploitation, as defined by Eriksson (2013), refers to having short-term focus on efficiency that is based on existing knowledge and technologies. In all of the four projects exploitation was prominent, both in similar and different ways. The findings indicate that all four case projects have in some aspect been relying on existing technologies, past experience and knowledge.
The findings illustrate a common pattern regarding exploitation on the process side of the case work since both organizations applied common project management systems in the case projects. In Druid, both case projects (Beauty Salon and University) applied Scrum as a project management system. This is seen as exploitation since it is a system Druid used for almost all of their projects, thus it can be viewed as a familiar method and building on previous knowledge, techniques and processes. The application of Scrum stems from the agile method as it is considered to be one of the most common and known project management system that agile organizations apply, therefore the connection between the application of Scrum and being an agile organization is rather evident. In both cases the teams worked in short sprints and had weekly reviews and feedback, as the Scrum framework proposes. This enabled the teams to better respond to changes in the case work, client requirements and the whole industry, which supports the agile method. Furthermore, in Case University, the team used some generic methods, and as mentioned in the empirical section, also referred as “lego blocks”, which entailed building on already existing processes and experience in order to find the best possible solution for the client.

Similar reliance on project management processes was observed in Fraktio. Even though the organization did not apply any official project management system such as Scrum, particularly in Case Auction the case work entailed application of “well proven” project management aspects and processes from past projects. Applying these project management systems is considered as exploitation since they embody processes that are common in the case organization as they were already applied in other projects as well. Therefore, the case teams and their members were familiar with these processes through existing knowledge and experience. Conversely, Case Research did not indicate strong reliance on project management systems. This can be explained through the specific characteristics of the case work. Since the case work did not include actual development of software rather a specification of the issues affecting the future development of software and systems, project management systems and processes that support development and release activities of software were not needed. In addition, as the case team only employed two employees of Fraktio, the need for managing actions of multiple individuals was low as the two members could easily discuss and go through different observations together.
Another process that was observed in majority of cases was peer reviews. Peer reviews were put in place in order to make sure that the developed solutions were of high quality through having an additional person checking for example the code in the solution. In Fraktio, peer reviews were referred to as co reviews and in Case Research this process was illustrated through the cross checking of observations and notes the team members had made. Similarly, in Case University conducting peer reviews was an established process. The findings from Case Beauty Salon did not explicitly show the application of the peer review process, however, this can be explained through the fact that the team members were globally dispersed. Yet, the findings did illustrate high degree of communication that the team members had with each other and with the other members of the organization. The application of this process is viewed as exploitation as the findings displayed that it was not only applied in the case projects rather it is a process that is included in software development processes in general. Ensuring the high quality of the developed solutions stems from the agile method that emphasizes technical excellence and delivering working software.

Stemming from the importance of constant communication in agile methods, the case projects established real-time communication channels both within the case teams and with the client. In Druid, the case teams in both Case Beauty Salon and University used a chat system called Flowdoc, in Fraktio Slack was used for the same purpose in Case Auction. Here, Case Research differed as the team did not establish a chat communication channel within the team or with the client. Again, since the case work was mostly investigation and specification for systems that will be developed in the future, there was not a similar need for real-time communication with the members of the client organization.

Exploitation arose in the technology side as well. Firstly, in both case organizations, one case project revolved around an existing platform or site. More specifically, in Case Beauty Salon (Druid) and Case Auction (Fraktio), the teams had to rely on existing components of the sites since they further developed and maintained an already made platform. In Case Beauty Salon, the platform was already established when they started the project, and similarly in Case Auction, the client had bought the service from another party, so already from beginning of the project the team would rely on the existing material, e.g. code and
software. In addition to exploitative aspect arising from the development of an existing platform, in Case Auction technologies such as deployment tools, libraries and repository management, that had been applied in other projects, were used.

Additionally, even though building a new site, existing technology was also relied in Case University. In Case University, as they built the site on Drupal, the team was able to apply a great amount of existing components of Drupal to the case work. There, the case work involved a great amount of basic implementation and applying of existing components, and this use of and reliance on existing technologies and systems enabled the team to already be aware of which direction to take and what components might be of best use. Unlike in three other case projects, the aspect of exploiting existing technology did not arise that strongly in Case Research. This is not surprising since the team did not develop software in the case project, thus there was no need to apply existing code, for example. However, as illustrated especially in Case Auction, and what is highlighted in agile software development organizations, simplicity is highlighted through the fact that if there is existing code that can be applied, developing new code is not favourable. Therefore, it is not surprising that reliance on existing technologies (code, software, systems) was such a prominent finding in the case projects.

Furthermore, exploitation arose in what Katila and Ahuja’s (2002) call local search where problems are addressed with pre-existing knowledge bases. Exploitation of existing knowledge became visible when the interviewees in Case Auction and in Case University discussed problem-solving. More specifically, when the team encountered problems concerning hardware or server, the interviewees in Case Auction explained that the team was able to use solutions from previous experiences and knowledge to solve the situation.

To conclude, the case to case comparison revealed consistent findings in issues regarding exploitation, especially in Case Auction, Beauty Salon and University where exploitation can be illustrated from a process perspective and technology perspective. These findings support Geraldi et al.’s (2011) notions on software organizations exploiting standardized technologies and well-defined operational processes. The fact that these issues did not arise as prominent in Case Research is explained by the fact that the case differed from
the other three since no actual software was developed. However, exploitation could still be seen in Case Research since the tools applied in the case work were still quite traditional and common.

5.1.2 Exploration

Unlike exploitation, exploration concerns having long-term focus on innovation that is achieved through application of new knowledge and technologies (Eriksson, 2013). The findings illustrate that all four case projects entailed exploratory efforts and activities to some extent.

The findings displayed that exploration emerged in terms of applying and experimenting new technologies and solutions. In Druid, both case projects had significant exploration efforts. In Case University, the case work included application of a media management system that was not only a novel component for the case organization but also for the whole Drupal community. In addition, comprehensive testing was implemented in the case work, which was something that had not been done in prior projects. Similarly, in Case Beauty Salon the team built the site with a whole new version of Drupal, namely Drupal 8, that had not been done in any prior project in the organization. Furthermore, this version was not released in the Drupal community when the team began to apply it, which implies the novelty of the version. In addition, in general Case Beauty Salon other technologies were used than normally, even though these technologies did not have a remarkable role in the case work. In Fraktio, exploration of new technologies could be found in Case Auction where some new libraries were taken into use in the case work. Similar exploratory activities regarding technologies did not appear in Case Research. Here, the explanation lies in the fact that the team did not develop any actual software in the case project. Instead, the case project entailed exploration aspect through its uniqueness. This uniqueness resulted from the scale and business environment of the client since it posed specific requirements and regulations for the proposed solutions.

Exploration on process side was less evident in the case projects. In Druid, some new processes were put in place, thus exemplifying exploration. Indeed, in Case University
new activities and processes were considered as exploratory since they had not been applied in prior projects. One process was the development of additional processes on top of the Scrum framework. Additionally, the case project had an external Scrum coach, which was also something that had not been applied previously in prior projects of the organization. Exploration in terms of processes did not arise that significantly in the remaining cases. In Case Research, one aspect of exploration in terms of process was that the team implemented user testing prior to actually developing software and systems. This was uncommon as it was often done in projects where software was developed, but in Case Research it was done to identify the various needs of the future users and demands for the system.

Furthermore, in some case projects exploration arose in terms of approaches to problem solving. Similar to what Katila and Ahuja (2002) termed as distant search, the case teams had to move away from existing knowledge bases since the problems resulted from issues not initially familiar to the teams. In Case Auction and Beauty Salon, since the case projects were “brownfield” projects with existing code and platforms, which were not developed by the case teams, some unforeseen and critical problems would arise from bugs in the code. The problems resulting from bugs were in these case projects considered to be unique and unforeseen, thus, all of them required an own approach for resolving them.

Key finding regarding exploration was the effect of support and freedom provided by the organizations’ since in all four cases the organizational support arose as a great force affecting and enabling exploratory activities. Therefore, for all the cases, the possibility to be creative resulted greatly from the organizational environments and characteristics. Moreover, this kind of organizational support stems from the agile methods since they emphasize providing a right kind of environment and support for the individuals and trusting that they produce and deliver what is expected of them. Moreover, the aspect of self-organizing teams reflects common decision-making power and organizational structures with low hierarchy, which can further be seen creating this kind of the organizational environments. In Druid, both case projects displayed strong indications of the organizational environment encouraging the case teams and their members to explore.
Even the whole organizational structure was described to be built on experimentation since the organization was still relatively young and therefore had not yet found the right work structures. Furthermore, specifically in Case Beauty Salon, the amount of freedom the team members had to explore can be considered to be an extreme as it was encouraged even over the potential risks of failure. Here, a trial and error approach was emphasized as it was considered important that individuals learn from mistakes.

Similarly, in Fraktio, the findings displayed that in general the organizational culture encouraged individuals to be creative and improvise. Even though the case work did not entail a significant amount of exploration, overall, the team members in Case Research felt that they were supported by the organization to experiment.

In the case projects that entailed actual software development, namely Case Auction, Beauty Salon and University, the client relationships were observed to have a great impact on the case teams’ ability to experiment and test, i.e. explore. This finding is not surprising since agile methods advocate for client centricity and collaboration. The flexibility and support for experimentation and being creative from the client’s side to the case team was described to be high in all these case projects. In Druid, both case projects displayed the importance of client relationship in exploration activities. In Case University, the client was flexible and open towards experimentation and therefore it was greatly up to the interest and skills of the team members to actually decide in which direction to go, what to learn or how to rethink previous solutions. In Case Beauty Salon, the high level of trust enabled the team members to explore and the client was understanding towards new solutions. Similarly, in Fraktio, close client relationship had a significant effect on exploration. The client in Case Auction was described as having a flexible attitude towards the case work and the high level of trust between the client and the team enabled the team members to explore. Therefore, the members had the opportunity and flexibility from the client to shape the project and bring new technologies and ideas forward.

Here, an interesting finding arose. Even though the findings indicate a great amount of freedom the case projects enjoyed in terms of exploration both from the client and organization, a somewhat common feature regarding exploration came forward, namely
“reflecting”. This meant that even though the members had the freedom to apply novel solutions, they still reflected them back to the case work and whether these solutions would bring any actual business benefit and value to the case work and that way to the client’s business.

5.2 Agile ambidexterity

At this stage of the analysis the exploitative and explorative activities that were taken in the case projects are identified and the analysis will now focus on how these activities were balanced. As presented in the theoretical framework, the importance of organization’s being able to pursue both exploitative and explorative activities simultaneously is widely acknowledged. In volatile business environments organizations are not only required to seek for novel opportunities but also ensure to use and gain returns on prior investments. The link between ambidexterity and higher performance is proven to exist through many studies.

5.2.1 Organizational structure and culture

The theoretical framework presented some key issues regarding the antecedents of ambidexterity. The findings indicate that both organizations can be considered to have suitable organizational structures and cultures for achieving ambidexterity. Indeed, the literature suggests that the structure of the organization have a role whether ambidexterity could be favourably achieved (Jansen, et al., 2006). Here, hierarchical and centralized organizational structures were suggested to have an unfavourable effect on exploratory activities as they could discourage team members to take initiatives and apply various knowledge in problem-solving. All four cases reflected freedom to be creative and experiment. The locus of decision-making could be found on the individual and team levels. In addition, the fact that arose in all four cases was that the team members could consult people who were not officially part of the case team. This further strengthens the notion that the organizations were structurally rather loose. Moreover, the aspect that both organizations are relatively young can be seen having an influence on the less rigid structures. This was an issue that A.I especially pointed out in Case University, that as the
company is relatively young, they still search for the right work structures through experimentation. Therefore, it can be stated that both organizations represent somewhat loose organizational structures. To conclude, this can be seen as a result of the application of agile methods as they entail self-organizing teams that are fairly decentralized and decision-making power is distributed all the way to the case levels.

However, to balance the loose structures that favour exploratory activities, the literature proposed that formal procedures are more suitable for exploitative activities as they provide efficiency and support improvements (Jansen et al., 2006). Even though agile methods promotes flexibility and high responsiveness to change, indications of the use of rather formal processes could be found in all four cases. In Druid, the application of Scrum as project managements system in the case projects offers common processes that the team members are accustomed to from previous projects. Therefore, Scrum framework can be seen as providing somewhat formal, underlying guiding structure for the case work. Even though Fraktio did not apply any formal project management system in its cases, the findings indicated that project management processes that were found well working in previous cases were applied. Moreover, both organizations applied peer reviews (referred as co-reviews in Fraktio) in the cases, which meant that everything that is developed was reviewed and checked by another person before release. Making sure that developed and released solutions are of high quality reflects the service attitude and willingness to fulfill client expectations.

Sharing the same vision and culture were proposed to provide an essential basis for ambidexterity. Here, social relationships and knowledge sharing culture were of importance in enabling exploitative and exploratory activities (Jansen et al., 2006; Lin & McDonough, 2011). The importance of organizational relationships became prominent in the findings as in both case organizations interactions and knowledge sharing between organizational members played an essential role in how exploitative and exploratory activities and solutions were distributed and balanced within the organizations. Therefore, it can be stated that there indeed were strong network structures and social ties within the organizations. However, as Simon and Tellier (2015) stated that too strong and dense network might have a diminishing effect on exploratory activities but this was not found in
the findings. Indeed, the organizations seem to have found a connectedness in order for both exploitative and exploratory activities to flourish. Thus, both organizations can be seen as having suitable structures and cultures in terms of providing a right kind of organizational structure for ambidexterity since they provided a basis for exploitative and exploratory activities. The contextual factors enabling ambidexterity in these organizations are discussed in the following section in more detail.

5.2.2 Contextual approach to ambidexterity

The literature review displayed that organizations go about ambidexterity with different approaches. The three most common approaches are structural, sequential (temporal) and contextual ambidexterity. This section will go on reviewing the findings in light of these approaches and analyze whether indications of one or a combination of approaches were applied. Overall, the findings displayed strong connection to the contextual approach in achieving ambidexterity, however, few aspects of the alternative approaches could be identified.

The somewhat traditional view to ambidexterity, namely structural ambidexterity, referred to dividing exploitation and exploration into different organizational units. Eriksson (2013) suggested that structural ambidexterity is often common in large organizations that possess greater resources and operations that enable the structural separation of exploitative and explorative activities. Here, exploitative units could be exemplified focusing in incremental innovation and efficiency whilst exploratory units are often R&D focused with tolerance for uncertainty. Not surprisingly, the findings did not have clear remarks that would suggest structural ambidexterity. Both organizations are rather small in terms of the amount of employees and have somewhat simple organizational structures and therefore, pursuing structural ambidexterity might be restrained by fewer resources. Moreover, the operations and tasks taken in the case organizations and case projects were described to be somewhat interrelated and the agile software development processes call for the ability to rapidly adjust to changing needs, therefore the case projects require both exploitative and exploratory activities to be pursued in the case projects, thus structural strategy to ambidexterity was not applicable. The findings are
consistent with what Chen and Kannan-Narasimhan (2014) suggested about structural ambidexterity being achievable on business unit or corporate levels rather than team or individual levels. Furthermore, since the case projects entailed high levels of knowledge sharing and interactions both within the case team and the whole organization, structural separation might have a hampering effect.

One aspect in Case University could be described as promoting structural ambidexterity. The case was described to be the organization’s “favourite” in the sense that the team could test new solutions and processes within the case work and thereafter apply those in other projects. Therefore, Case University can be considered to resemble a “unit” where exploratory activities were pursued and further applied and integrated. In addition, in the remaining cases the same notion arose that exploration in one case project could be further applied in other projects. However, even though worthwhile mentioning, this aspect does not imply structural ambidexterity that strongly since Case University also applied exploitative activities in the case work such as Scrum. As structural ambidexterity refers to exploitation and exploration activities being divided into different units (e.g. case projects), the findings do not indicate reference to structural ambidexterity. Due to the agile software development process, dividing exploitation and exploration activities into different units or organizational structures may not be feasible or even possible. Furthermore, the case organizations’ size might hamper structural ambidexterity.

Another approach to ambidexterity is sequential. Through sequential ambidexterity, organizations balance exploitation and exploration with separating the two activities in time and focusing on one at a time and thereafter turning focus on the other. This was exemplified by Turner et al. (2014) through an exploratory phase with new product development activities, which is then followed by an exploitative phase where the new product is released to the market. None of the four cases indicated a strong and clear pattern of sequential switches between focus on exploitation and exploration. Signs of some degree of sequential ambidexterity can be identified in Case Beauty Salon, where the team focused nearly solely on testing the new version of Drupal 8 during one sprint and therefore performing more exploratory activities and leaving the “everyday” and more exploitative activities to a smaller role. However, since the case work also entailed aspects
of exploitation, even if with less focus, strict temporal separation was not applied. In addition, as noted, there were no supporting patterns that could be identified from the remaining cases, and therefore this can be considered to be an individual incident that resulted from the need to make advancements in testing the new version.

Another aspect where signs of sequential ambidexterity can be reviewed is the problem-solving activities in two case projects (Auction and Beauty Salon). The findings indicated that the criticalness of the problem determined the actions that were taken in regards to solving the problem. If a problem was critical, everything else was “dropped” and focus was put solely to solving the problem. Here, how the problems were solved depended on whether the team had previously encountered a similar problem or whether it was a new and unique one. Hardware related problems were described to be somewhat common, thus the team could often relate to past experience and prior knowledge in solving the problem, thus reflecting exploitation. Solving unique problems on the other hand was more explorative as the problems were often unforeseen and resulted from bugs in the software that the team might not have developed. Therefore, if the problem was critical and thus the teams focused solely on that and applied either exploitation or exploration in solving it, an aspect of sequential ambidexterity can be identified. Even though these aspects do not indicate a sole focus on sequential ambidexterity and serve only as “mild” events, they are still worthwhile addressing since they reflect alternative avenues for achieving ambidexterity.

The lack of indication of sequential ambidexterity in the findings is not surprising. Sequential ambidexterity is considered to be suitable for stable environments (Eriksson, 2013), which is somewhat opposite what software development entails. Therefore, overall it can be noted that sequential ambidexterity might not be suitable for agile software development since the work is done iteratively with short planning cycles and therefore the teams need to be able to respond quickly to changing needs and different situations by applying both existing and novel solutions. In the proposed solutions the team needed to configure existing and novel knowledge and technology.
Lastly, the third approach to ambidexterity is contextual ambidexterity. Here, social context and performance management are of great importance in enabling contextual ambidexterity (Gibson & Birkinshaw, 2004). These consist of four attributes, namely stretch, discipline, support and trust. Stretch referred to encouraging and provoking the individuals to voluntarily pursue ambitious objectives and it can be identified in all four cases through the high organizational support and freedom described by interviewees. Moreover, software development in general, as many other professions, is characterized by the desire to deliver high quality services and solutions. This aspect can be seen further supporting the individuals to reach high performance and aim towards ambitious objectives. The fact that strong client satisfaction and service attitude were mentioned to be of importance in being a successful software organization strengthens this aspect. Secondly, discipline refers to individuals voluntarily meeting the expectations that result from their commitments. Even though individuals in the case teams had remarkable freedom to experiment, it became evident that they still reflected new ideas and solutions to the potential business benefit they could bring to the case work and for the client's business, therefore indicating that the individuals had a “disciplinary frame” that their affected their work. Thirdly, support reflects the assistance and encouragement individuals give to one another and it results from freedom to take initiatives and assistance among the individuals. The findings displayed high degree of cooperation and communication not only among the case teams but also among the whole organizations. Knowledge sharing was not only restricted within the case team as the team members could go and “consult” people that were not officially part of the case team. The fact that the team members did this kind of “consulting” can be seen resulting from the loose organizational structure and also the fact that the organizational members resided in the same physical environment that enabled interaction with other members of the organization. Finally, the last attribute is trust, which refers to the individual's ability to rely on the commitments of each other. The findings illustrated rather loose decision-making structures as decisions could be taken both in the individual level as well as commonly within the case team, which reflects the established trust. Here, especially noted in Case University, if a team member applied a new solution in the case work, it was expected that he or she had done the background work in order to be the right person to know whether or not it was a suitable solution.
As the findings emphasize, the organizational structure and culture played an essential part in providing the employees with and supporting them in taking exploitative and exploratory actions. Why contextual ambidexterity is found most prominent in agile software development organizations can be explained through the contextual factors that the organizational structure and culture provide. Indeed, in all four cases the organizational support and encouragement towards exploitative and exploratory activities became evident. The team members were guided by organizational discipline and stretch to deliver high quality solutions that fulfill the required standards and supported and encouraged to experiment. This finding that the case organizations had contextual approach to ambidexterity supports Napier et al.'s (2011; 2008) notions who found that software organizations develop contextual ambidexterity. Overall, even though some signs of structural and sequential ambidexterity can be identified, the findings display that contextual ambidexterity was the main approach through which the case organizations achieved ambidexterity.

5.2.3 Distributed type of ambidexterity

To gain further insight into how ambidexterity is achieved in agile software development organizations, the findings are reflected to the framework of point and distributed ambidexterity. The relatively new framework created by Turner et al. (2016) emphasizes the managerial role and activities regarding ambidexterity, which in this thesis are not completely applicable since the basic foundation of an agile organization is based more on equality and less hierarchical structures. Even so, it can still be applied and analyzed in order to review how the exploitative and exploratory actions are being addressed in the case projects, and to shed light into what type of ambidexterity is more prominent. The different actions, namely buffering, gap-filling, role-expansion, tone-setting and integration (Turner et al., 2016) can all be identified and designated in the case projects without the managerial aspect.

Buffering, which is distinguished as the manager acting as a barrier to avoid distractions affecting the performance and the central point of contact with the client, which provides an efficient approach to knowledge-sharing in the team (Turner et al., 2016). From the
case organization Druid, the concept of Starplayer can be related to the buffering aspect in that sense that the Starplayer acted as the “go-to person” with a good overview of what is happening and what should be done. The case revealed that the concept had a clear role in the knowledge sharing aspect as the Starplayer took part in meetings where valuable information and ideas were shared. The Starplayer is not managing the project, which relates to the buffering since the main aspect of buffering is to be the “voice of reason” and direct other team members (Turner et al., 2016). Further, in most case projects the contact with the client were handled by several people in the case teams through communication channels like Slack and Flowdoc, which were open for all people working in the specific cases. The knowledge sharing in the case projects strongly took place through continuous personal interactions between case members, which stems from the agile methods. As mentioned in the previous section, communication with the clients was done by different communication channels, but also by face-to-face interactions. This is not surprising since constant and on-going communication with the client is strongly emphasized in the agile methods and the knowledge sharing between case members as well.

Gap-filling is the next action where the manager ensures that all the tasks are in fact being performed, and according to Turner at al. (2016), this can be seen as an exploitative function which supports the knowledge sharing within a project. Similar to the buffering, the concept of Starplayer can be applied here. The Starplayer has an overview of the project and could in that sense have a broader view on what needs to be done in order to move on according to schedule, and make sure all assignments are being completed. In Fraktio, where the concept of Starplayer did not exist, the team members in the case projects were focusing on the tasks for which they were most suited in terms of expertise and interest. This was also the case in Druid but when needed, the Starplayer could raise the attention to an issue not being taken care of. Even with the Starplayer, it was indicated that in the case projects, it was not up to one single individual to make sure the tasks were being performed, as it was still a team effort. The individuals took responsibility for their area of expertise and if help was needed, they turned to someone in the same case project or in the organization for support and knowledge. This supports that the knowledge sharing in the case projects happened both within the case members but also with the other employees in the case organization. The agile method highlights close team
relationships and interactions between individuals as important, which supports the agile working methods in both of the case organizations.

Role-expansion is identified as a response to an event where exploitation and exploration increases in order to solve the situation. This can be exemplified through a technical fault, which requires imaginative solutions and increased communication with the client, or as a more intense “business-as-usual” operation (Turner et al. 2016:15). Turner et al. (2016) also note that this generally falls on the manager but also that the solution is limited by both the client and organizational constraints. Case projects in both organizations illustrated that when a critical problem arose everything else was “dropped” and the team focused solely on solving the problem. This further strengthens the distributed ambidexterity. For the case projects, the relationships with the clients were of high importance when it came to the daily activities performed in the case projects. This was exemplified in Druid where the team “...went on a completely new trail...” which thanks to the client’s trust in the team resulted in a successful outcome. It was further exemplified that when the case team wanted to try a new solution the client replied “why are you even telling me this, go and try”. These examples show how creativity and experimentation is based on the client relationship and most importantly on the trust between the client and the case teams. In the case projects, some of the problem-solving rely on existing experience where solutions depend on past knowledge, however, also noticed in the projects are that creativity and allowed space for experimentation have a big effect on unforeseen problems arising. Prominent in this section is that the close collaboration and constant communication between the client and the team members are of high importance in problem-solving. Collaboration and constant communication are yet another indication of that the case organizations follows the agile methods.

In order to set the right tone for the work and to ensure adherence to the procedures, and collaborate with the client to favour flexibility, tone-setting is identified (Turner et al., 2016). Here, the manager sets the exploitative and explorative philosophy for the project meaning that a case member get a target with the intention to reach it, and if not possible, the case member should inform the rest of the case team (ibid). Tone-setting in the case projects is somewhat difficult to distinguish. In Case Beauty Salon, when a team member was asking
for which option to proceed with, the interviewee replied that it did not matter which option the team member chose, the interviewee would still pick the same. This exemplifies that tone-setting in this case highlights the fact that freedom to experiment and be creative with solutions is prioritized, even if the result is unsure. It is further explained that when a task is given, it is not provided with a clear picture of how it should be done. Being flexible is central in the agile methods in order to make quick adjustments and respond to changes. In the case projects, flexibility is connected to the relationship with the client and the organizational culture. In all the case projects the case teams had support from the organization and the client to freely choose how to work and to experiment and be creative.

Integration works as the “spine” in the model presented by Turner et al. (2016) and illustrates bringing together the knowledge and team members in the project to create a coherent whole, which also can include individuals who are not directly involved in the project. As seen in the case projects, they had peer reviews within the case projects to span the knowledge. However, it was further noted that knowledge in the case projects not only derived from the actual members in the team but also from employees in the organization. The project members were encouraged by the organizational culture to include knowledge and expertise from other members, which increased the knowledge sharing not only in the case projects but also in the entire organization. In the different case projects exploration was done through experimentation, and the solutions brought from this experimentation could later be used in other case projects. Further, by pursuing exploration in the case project can later result in exploitation somewhere else in the organization and in another project. According to Turner et al. (2016) this kind of integration is supported by the manager, but in most of the case projects this was embedded in the organizational culture and the freedom from the client.

In all four case projects both exploitative and exploratory outcomes were noticed, which validates ambidexterity in the cases. Overall, the outcomes in the case projects were not achieved by one single individual and the balance of exploitation and exploration is seen to be distributed among the case members. This highlights a distributed ambidexterity in the case projects, however, in Druid, the distributed ambidexterity can in certain situations be
seen to move towards a pointed ambidexterity. This due to the concept of Starplayer, as the Starplayer could be considered to have a better overview of the case project, which entails the knowledge of knowing which actions could be more beneficial to initiate.

Indeed, the distributed ambidexterity is found to be more prominent than point ambidexterity in the case projects. This comes down to the fact that the teamwork in the case projects with the extensive knowledge sharing, client relationship and organizational culture supports the agile methods of collaboration and close cooperation with the client. The importance of knowledge sharing cannot be stressed enough, and the knowledge sharing does serve as the spine and integration mechanism in the case organizations. As seen, knowledge sharing emerged both from within the case team through problem-solving, collaboration and communication, but also from people not officially part of the case project. The organizational culture and the agile methods support the case team to consult individuals not part of the case team due to their competence and expertise. What also could be noticed was the deliberate knowledge sharing in the Starplayer meetings. Without this intensive and broad sharing of knowledge, ambidexterity in an agile organization would be hard to reach.

5.3 Bringing it all together - Ambidexterity in agile software development organizations

The analysis has now reached the stage where the above-presented issues are combined and discussed together in order to gain more insight into the various relationships and effects between the different parts. So far, the analysis has discussed what kind of exploitative and exploratory activities were taken in the case projects and which approach to ambidexterity the case organizations had. Furthermore, a relatively new framework by Turner et al. (2016) was discussed in light of the findings. Remarkable in the findings was that many issues seem to stem from the characteristics of agile software development organizations, that result from the application of agile methods. Indeed, the characteristics of these organizations and the organizational environment they have arisen as the focal point of the analysis and create a foundation to further stem from. Therefore, this section will continue on further discussing the connections and relationships between the
characteristics of agile software development organizations and the issues regarding ambidexterity. In order to fully grasp the discussed issues, Figure 9 presents a model that was created to further display the key issues. The model presents the key findings and insights into how ambidexterity takes place in an agile software development organization. The model illustrates how the characteristics of agile software development organizations lead to a specific approach to ambidexterity that further leads to business units, which in this study are the case projects and their members, which are encouraged to simultaneously exploit and explore. Lastly, these exploitative and exploratory activities can be further brought back and distributed to the organizational level and applied in other case projects. The arrows below the model indicate more specifically the effect the characteristics of agile software development organizations have on each finding. The model is discussed in more detail in the following section.

**Figure 9. Summarizing framework of the analysis**
As the model illustrates, the characteristics of software development organizations that result from the application of agile methods serve as a starting point for how ambidexterity is achieved in these organizations. The characteristics entailed low hierarchical structures and emphasis on individuals being able to take own actions as well as high degree of communication and collaboration. Earlier in the analysis, we found that the main approach the case organizations had on ambidexterity was contextual. This was not surprising since contextual ambidexterity highlights the organizational context in supporting the business units and individuals to simultaneously exploit and explore and this was found prominent in the findings. Indeed, we propose that the explanation behind the contextual approach is strongly rooted in the characteristics of the organizations that promote this approach. The findings illustrated that agile software development organizations seem to have a distinctive organizational characteristics and environment where a supportive context is created for the case teams and their individuals to pursue exploitative and exploratory activities.

Moving to the third box in the model, as the contextual approach to ambidexterity entails, business units and individuals are encouraged to pursue exploitative and exploratory activities. Indeed, as mentioned, contextual approach provides the right environment and context for the case members to simultaneously exploit and explore in the case projects. In addition, how and what kind of exploitative and exploratory activities are undertaken in the case projects is again affected by the characteristics that stem from the agile methods.

In the case projects where actual software was developed, reliance on existing technologies and software became evident. This can be seen stemming from the simplicity aspect of agile methods that reflects the “reuse” of for example existing code base, something that was also pointed out in the findings. Furthermore, the application of project management systems in the case projects depicts agile methods as in Druid the case teams followed Scrum and in Fraktio the project management systems consisted of well proven processes. Therefore, it is suggested that the agile methods affect how exploitation and exploration arise in the case work and we suggest that exploitation arose as an execution aspect as the exploitation activities, i.e. applying existing technologies and project management systems, can be considered to be rather routine-based activities as
they were something that were generally applied in projects. More specifically, exploitation can be said to develop in the case projects through existent knowledge and experience that the members had of the technologies and processes from prior projects. The findings illustrated that exploration activities stemmed greatly from the organizational environment that encouraged and supported experimentation as well as through the client relationship that enabled the case teams to be creative and experiment. Both aspects can be drawn back to the agile method and the characteristics organizations that follow agile methods. Agile method advocates for supporting and trusting individuals, which are reflected in the specific organizational environment that enables individuals to explore. Client centricity and close relationships with the client, which are common in agile methods, often result in a high level of trust from the client towards the case team and this enabled the teams to apply new solutions in the case work, i.e. explore.

Furthermore, as brought forth earlier in the analysis, the case projects were found to have distributed type of ambidexterity. That is, exploitation and exploration activities were performed by a combination of individuals rather than a result of one individual’s activities. Once again, these findings can be grounded to the agile methods. Agile methods advocate equality and self-organizing teams. Moreover, the emphasis on collaboration in agile methods further strengthens the notion of exploitative and exploratory activities being performed by multiple individuals. Indeed, the distributed type of ambidexterity reflects that the case members together with collaboration are the foundation for the exploitative and exploratory actions taken in the case projects. The notion of informal role definitions and interrelated nature of the roles further supports the distributed type of ambidexterity. In general, based on the findings, it can be proposed that organizations that have less formal hierarchical structures and collaborative networks develop distributed type of ambidexterity.

Lastly, as the model illustrates, the exploitative and exploratory activities and solutions did not only remain within the case projects. The findings displayed a high degree of interaction and communication within the organizations and through these the solutions that were developed in one project could be implemented in another. Again, the high degree of interaction and communication can be connected back to the agile method. The
organizational characteristics resulting from the application of agile methods enabled the application of the exploitative and exploratory activities and solutions further to the organizational level, and through the organization to other case projects as well. Therefore, we propose that this organization-wide distribution of exploitation and exploration activities and solutions was enabled by the agile methods and the extensive knowledge sharing and interaction, both within the case projects as well as by the communication and “consulting” with other organizational members who were not assigned to the specific case projects.

5.4 Agile software development organizations as ambidextrous communities

As the last part of the analysis we take the discussion further and propose a community aspect in relation to how agile software development organizations achieve ambidexterity. As discussed in the previous section, the specific characteristics and nature of agile software development organizations have an essential role in how these organizations achieve ambidexterity and stemming from this notion community aspect is introduced. Why the term community is brought up is because we believe it represents well the organization-wide aspects that adherence to agile methods entail. We propose that following agile methods and applying them in business operations creates a common foundation and a belief system for the organization as well as provides a framework for various processes such as knowledge sharing and interactions. Furthermore, these issues arise in shared values regarding for example client satisfaction and high quality services and solutions appear to provide an underlying structure for the activities taken in the case projects.

As discussed earlier in this thesis, in prior literature and research ambidexterity has often been addressed from the managerial perspective emphasizing the senior management’s responsibility in managing and achieving ambidexterity, however, this view is not as suitable in the context of agile software development organizations since they may not have such strong managerial roles or positions. Therefore, the loose organizational structures further strengthen addressing the organizations as communities. There were no
strong hierarchical structures or bottlenecks through which interactions or communication needed to go through. Therefore, knowledge sharing within the community was continuous, meaning that the knowledge can be shared in the community at any point in time, from the case projects to the organization and also the other way around thus enabling cross-fertilization of ideas, both exploitative and exploratory. Indeed, this was reflected in the consulting aspect and that the case members, by the agile methods are encouraged to share information to and within a team through conversation, which is considered to be the most efficient and effective method. From this we thereby argue that a community perspective is suitable as it expresses the important and crucial role of knowledge sharing in an agile software development organization.

Another reason why it is suggested that agile software development organizations are referred as communities is the collective nature of these organizations. Agile software development organizations emphasize equality and self-organizing teams, thus, as mentioned earlier, strong hierarchical structures are not put in place. Even though we posit that agile software development organizations have contextual approach to ambidexterity, we propose that the community aspect provides additional insight into contextual ambidexterity. Therefore, providing additional insight into contextual ambidexterity, which often takes a managerial perspective in the sense that it is still the responsibility of the managers to create suitable organizational context, the community perspective highlights the common beliefs (adherence to the agile methods) and underlying systems as creating and promoting right kind of environment for ambidexterity.
6. Conclusion

The thesis has now reached its final chapter, which will present and summarize the main issues from the previous chapter. The aim of this chapter is to draw together the highlights of the findings of this study. This is done through referring back to the issues and research questions presented in the first chapter.

The structure of the chapter follows as such; firstly, the research and its sub question is reviewed and the issue of this thesis is discussed in light of the findings. Secondly, managerial implications are discussed and the limitations of the study are presented. Lastly, areas for future research are suggested. Following the explorative nature of this study, the main focus is not to create generalizations of the research topic rather than gaining an initial insight and understanding into how ambidexterity is achieved in agile software development organizations. Since the research topic remains somewhat uncovered in the current research and literature, the aim of this study is indeed to provide a basis for future studies where generalizations can be pursued.

6.1 Back to the beginning

This thesis stemmed from the aim to shed light into how ambidexterity can be achieved in organizations. Therefore, the findings presented in this study give additional insight into the continuous discussion in the literature about how organizations can balance exploitation and exploration, i.e. achieve ambidexterity. More specifically, the findings illustrate how this occurs in agile software development organizations.

Theoretical framework discussed how today’s organizations struggle with exploiting existing resources and capabilities to ensure short-term outcomes while simultaneously exploring new opportunities that support long-term viability. Indeed, exploitation and exploration pose a dilemma for organizations since they often face difficulties in balancing exploitation and exploration due to the different natures of the two activities, and resources
and processes they require and entail. Yet, the prior research and literature has expressed that balancing the two activities is crucial for sustainable performance and success. Here, the concept of ambidexterity is proposed to provide a solution as ambidextrous organizations simultaneously pursue exploitation and exploration. Managing and achieving ambidexterity has often been viewed from a managerial perspective where the role of the senior management is emphasized. There are, however, organizations with less hierarchical structures where business operations are conducted in an equal organizational environment with low hierarchical structures. Agile software development organizations represent such organizational context where extensive collaboration among the organization’s members is highlighted, and business operations are conducted in an organizational climate that emphasizes equality and trust. Therefore, it can be stated that the notion of ambidexterity being managed by senior management may not be applicable to agile organizations since they lack strong managerial systems.

### 6.2 Providing the answers

We will now move on to reviewing and responding to the questions posed in the beginning of this study. As described in the introduction, in order to grasp how agile software development organizations achieve ambidexterity, i.e. balance exploitation and exploration, a closer look into the exploitative and exploratory activities taken in the organizations and case projects was needed. Therefore, in order for the research question to be fully answered, the sub question presented below will first be examined and answered.

**SQ1: How do agile methods affect how these organizations exploit and explore?**

Both exploitation and exploration can be identified not only on the case level but also on the organizational level. This can be argued by the fact that the exploitative and exploratory activities were not only applied in single case projects but also distributed to other projects the organizations had at hand. Here, the characteristics of agile methods were identified to play a crucial role in the distribution methods. More specifically, intensive knowledge sharing and interaction with other members in the organization resulted in
exchange of information and ideas. The culture and climate in the case organizations, with a loose organizational structure, supported and encouraged knowledge sharing between the case members. However, the knowledge sharing did not only take place between the case members but also in a great extent with other members in the organization. In this way knowledge sharing enabled both exploitation and exploration to take place since the knowledge flourished throughout the organization, and further, in that way also work as the foundation for further exploitation and exploration actions to take place.

Agile software development organizations seem to embrace the dilemma between exploitation and exploration somewhat naturally. Indeed, constant development and learning are of great importance in these organizations, however, focusing solely on these activities, i.e. exploration, contains risks. However, exploitation is also required to provide an underlying structure and guiding for conducted work. The findings illustrated that agile software development organizations seem to comprehend the importance of and need to apply both activities. Furthermore, the findings of this study did not show extreme difficulties for the organizations to simultaneously pursue both activities.

This study uncovered that exploitation and exploration can be said to be treated orthogonally in agile software development organizations. Since exploitative and exploratory activities were performed simultaneously in the case projects, it seems that the two activities were pursued as orthogonal rather than on a continuum. Indeed, the findings did not illustrate an “either-or” aspects that is common when the two activities seen through a trade-off. Supporting the orthogonal view is the apparent interrelated nature of the two activities. As mentioned, the findings of this study support the notion of the interrelated nature of exploitative and exploratory activities. Indeed, as the literature suggests, even though these activities differ in terms of required resources and focus from the organization, they still are connected as some exploitative activities are often needed in exploring new opportunities and vice versa. The duality of exploitation and exploration found in the case projects demonstrates that doing exploitative activities included some degree of experimentation, as when building on top of the Scrum methodology. Similarly, conducting exploratory work to some degree also involved underlying exploitative actions, such as the application of project management systems and technologies serving as a
structure and guiding to the work pursued in the case projects. Since agile software development often proceeds in fast pace with short iterative cycles, it can be stated that both exploitation and exploration are needed to maintain high level of quality and also respond to changes.

Furthermore, the explanation behind why encouragement for experimentation, i.e. exploration, was prominent in the findings can once again be related to the agile method. Exploration is often viewed including uncertainty since feedback and returns might remain in the far future. However, it can be noted that the agile methods might be seen diminishing this unfavored effect of exploration. With the iterative nature of agile software development, feedback and testing loops are short, thus results from exploration activities can be reached rather quickly.

**RQ: How is ambidexterity achieved in agile software development organizations?**

The findings from the sub question provide insights and guidelines in how to further answer the research question. To begin with, this study found that agile software development organizations address ambidexterity through providing an organizational context that enable and support the individuals to both exploit and explore. Instead of employing the traditional approaches to ambidexterity, namely structural and sequential, this study indicates that agile software development organizations have a contextual approach to ambidexterity, which is creating an organizational context and environment that encourage and support the project members to simultaneously pursue exploitation and exploration. Furthermore, the characteristics of agile software development organizations enable distributed type of ambidexterity where the intense collaboration and equality result in that all team members pursue exploitative and exploratory activities rather than one individual (point ambidexterity).

From this, we can conclude that agile software development organizations achieve ambidexterity through balancing the exploitative and exploratory activities and solutions taken in the case projects. The balancing is enabled by the distribution of the activities and solutions in the organization. The distribution is enabled by the characteristics of the agile
software development organizations, most specifically through the extensive knowledge sharing and collaboration between the organizational members. It was further acknowledged that the organizational environment with emphasis on social interactions plays an essential role in the distribution and balancing of the exploitative and exploratory activities. Therefore, it can be concluded that not only was ambidexterity achieved on the case level, i.e. simultaneous application of exploitative and exploratory activities, but also being present on the organizational level.

The proposed model presented in previous chapter displays how the characteristics of agile software development organizations affect how they achieve ambidexterity. So far, what we have seen, this relation has not been presented or discussed in the current literature. Stemming from the model, we proposed the idea of agile software development organizations as ambidextrous communities. Even though it is already emphasized that in organizations that have contextual approach to ambidexterity the organizational context is of great importance, in agile software development organizations its importance seems to rise onto a new level. Since these organizations do not have strong managerial positions or high hierarchy, the organizational context serves as a kind of mechanism through which exploitative and exploratory activities are balanced, i.e. ambidexterity is achieved. This study provides insight that ambidexterity can indeed be achieved without strong, formal managerial actions.

Thus, arising from this study, the aspect of treating the organization as a community highlights the importance of common beliefs, systems and processes. Adherence to agile methods indeed seems to provide the organizational members a common understanding of and compliance to how to act and organize. This in turn brings up an interesting aspect to ambidexterity through reviewing it more strongly through a social perspective. Yet, this perspective is based on somewhat small organizations and one can expect the community perspective to become more complex to grasp and understand as the organization’s size increases.

Overall, organizations that follow agile methods seem to embrace ambidexterity rather well. As mentioned earlier, exploitation and exploration did not rise as an inherent dilemma
in the case organizations rather than being pursued somewhat naturally. The ease of pursuing both activities can be seen resulting from adherence to agile methods, hence, we conclude that the organizations that follow agile methods are somewhat automatically provided with structures and processes to applying both activities, i.e., being ambidextrous.

6.3 Implications

Ambidexterity literature is constantly evolving and growing. This study adds its share to this process through providing additional insight into ambidexterity in organizations. The findings of this study further strengthen the notion of software organizations having a contextual approach to ambidexterity (Napier et al., 2011; 2008, Ramesh et al., 2012). However, this study sheds deeper light into how the characteristics of agile software development organizations affect achieving ambidexterity through the contextual approach. In addition, from these findings this study puts forth a perspective to view these organizations as ambidextrous communities.

The findings indicate that the characteristics of agile software development organizations have a significant impact on how ambidexterity is achieved in the organization. In this regard, it becomes of great importance to nurture this kind of organizational context. Furthermore, since agile methods greatly affect the actions taken in the organizations, it is essential that the essence of the method is clearly spread and apprehended in the whole organization. The findings of this study illustrated that the importance of agile methods and their application in the business operations was high and therefore it is essential that all organizational members are in agreement on and understand how to act along the principles behind the method. In addition, since agile method hiring and retaining employees who understand the advantages of working through agile methods it is crucial to realize the full value it provides.

6.4 Limitations

When conducting the research, several limitations can be considered. Firstly, as the study was conducted with organizations that are relatively small, the results need to be
considered on that level as they might not be generalizable for software development organizations of greater size in terms of employees or turnover.

Another aspect that can be seen as a limitation is the choice and comparability of the case projects. Three cases out of four (Case Auction, University, Beauty Salon) involved actual software development while Case Research involved only pre specification for the software and systems that would be developed in the future. Thus, one might argue that Case Research does not provide the same basis for comparison and analysis, however, the decision to include the case in the study was based on the desire the represent the variety of projects and tasks agile software development organizations conduct. Additionally, Case Beauty Salon can be considered differing from the other cases as the team members were somewhat globally dispersed. This might be considered to affect the team dynamics and create different setup for the team work. Yet, similarly, the case was included in the data as it provides a good example of how agile software development can be realized.

Even though characteristics and activities of agile methods are common, they can be applied differently in different organizations. Thus, a way that one organization follows agile methods might differ from the way another organization does. Therefore, the findings of this thesis might not be applicable to all organizations that follow agile methods. Similarly, this study investigated agile organizations that operated in the software industry. However, agile organizations can also be found in other industries, and thus alternative findings can arise from other industry contexts. Therefore, we acknowledge that the findings are somewhat limited to the presented context and thus further research is required in other organizations and industries in order to increase the generalizability of the findings.

Another issue posing potential limitation for this study is the fact that majority of the interviews were conducted with the founders of the case organizations. This can have an effect on the responses as they might have differing views than, for example employees that were hired to the organization. However, since interviewing all members in the case projects was not possible due to time and availability restrictions, the chosen interviewees
were considered to provide the best possible sources of information of the case projects. Therefore, as described in the methodology chapter, the interviewees were chosen based on their involvement in and knowledge of the projects. Furthermore, regarding the interview guide, the risk that the main issues were not fully understood by the interviewees, even though the guide included a description of the main themes of the interview, is present. Additional risk can arise if the interviewees misunderstood the questions.

In terms of the covered concepts in this thesis, prior research and literature has acknowledged the importance of dynamic capabilities in relation to ambidexterity. Even though the findings of this study reflect the dynamic process of applying exploitative and exploratory solutions from the case projects to other projects in the organization's, dynamic capabilities of the individuals, teams or organizations were not examined in depth. The decision to not include notions of dynamic capabilities in the theoretical framework or examine them thoroughly in the empirics was based firstly on the fact that the researcher’s felt there would be a need to have a stronger individual focus in the level of analysis in order to fully cover the dynamic capabilities. Overall, since ambidexterity is not deeply studied with specific focus on the characteristics that agile software development organizations entail, the concepts were kept rather simple in order to let various aspects arise from the research context. However, this said, the researchers do not minimize the importance of dynamic capabilities in achieving ambidexterity and suggest that future research should pursue on this path.

6.5 Areas for future research

Even though the authors believe that the current study was able to provide additional insight into ambidexterity in agile software development organizations, following areas are suggested for future research. Firstly, as agile software development projects are implemented with extremely close interaction with the client, future studies could include the client perspective in how ambidexterity arises in agile software development projects. As displayed in this study, the level of collaboration and trust between the client and the case teams were high, it would be sensible to review the client’s side as well.
Secondly, since ambidexterity in agile software development organizations has not been examined that thoroughly in existing research, this study focused on uncovering the basic aspects into ambidexterity and thus purposively left out additional aspects, such as intellectual capital and dynamic capabilities that have been connected to ambidexterity in prior research. However, as the findings revealed the importance of knowledge perspective in how ambidexterity is achieved in these organizations, further examination of the dynamic nature of knowledge processes are called for.

This study focused mainly on the team level, however, the data was collected through key individuals in each case and not from all participants in the case projects. Therefore, valuable insight could be gained if the research topic was studied from the individual worker's perspective, that is to interview every member in the case teams in order to get a broader view. Here, more detailed and comprehensive view of the dynamics in the case projects could be attained. In addition, deeper insight on the relation between self-organizing nature of agile organizations and teams to ambidexterity could provide interesting standpoints.

A community aspect to ambidexterity in agile software development organizations was proposed in this study. Somewhat similar notions were made by Barlatier and Dupoué (2015), who discussed how contextual ambidexterity was achieved with communities of practice, which reflects similar issues as our findings. However, they studied ambidexterity in a multinational firm where communities of practice were intentionally created and developed. In addition, in their study the communities of practice reside at the business unit level. Therefore, as put forth in this study, conducting further studies through the perspective where the whole organization creates a community of practice could provide some valuable insight.

Finally, a longitudinal study of ambidexterity in agile software development organizations could provide interesting insight into potential evolution perspective of ambidexterity and whether for example the approach to ambidexterity changes or alternates during a longer period of time.
“I don’t know where I’m going from here but I promise it won’t be boring”

(David Bowie, 1997)
References

Interviews

A.I, Back-end developer and founding partner, Druid, 24th of March 2016, Helsinki Finland
V.K, User experience designer, Fraktio, 21st of March 2016, Helsinki Finland
P.L, Software developer and founding partner, Fraktio, 21st of March 2016, Helsinki Finland
V.M, User experience designer, Fraktio, 21st of March 2016, Helsinki Finland
J.P, CEO and founding partner, Fraktio, 21st of March 2016, Helsinki Finland
M.S, CEO and founding partner, Druid, 23rd of March 2016, Helsinki Finland

Books & articles


Fraktio. [online] Available at: https://fraktio.fi. [Accessed 31st March 2016]


Appendix

Appendix 1. Agile Manifesto and Principles

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile manifesto (according to Moran, 2015:235)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Twelve principles behind Agile Manifesto (according to Moran, 2015:235-236)
Appendix 2 – Interview guide

Interviews for Master thesis

This interview will serve as the empirical part of our Master thesis at Linköping University. The results will be used to increase our understanding of ambidexterity in an agile organization. The information will be collected through two individual cases, which were selected for this study due to their suitability for and being representative of the key concepts of the thesis.

The concept of ambidexterity refers to being capable of exploit existing knowledge, technology and capabilities and also explore new knowledge, technologies and innovations. The importance of balancing these two activities, i.e. being ambidextrous, has been highlighted by many authors as it is argued to result in superior performance and firm survival. In this thesis, we aim to shed light into ambidexterity in the context of an agile organization, as we believe it will extend to current literature with new dimensions through the distinctive characteristics of an agile organization.

Kindly note that there are no wrong answers, only different points of view and we encourage you to share your thought during the interview. The interview will proceed as such: the interviewers will ask a question at a time and the interviewees will response to the question in turn. We kindly ask you to respectfully listen as others share their views, however, we also encourage you to take part in the discussion. The interview will be audio recorded in order to make sure that all important information will be captured.

Many thanks for taking part in this study, we highly appreciate your time and input.

Emma Castrén
Malin Gylling
Interview Questions

Background

1. What is your position?
2. How long have you worked here?
3. Can you briefly describe the purpose of this case?
4. What was your role in this specific case?
5. How were you assigned to work in this case?

Case questions

6. How would you describe your expectations on the case before it started?
7. How much were you able to rely on and use some existing procedures and items for this case?
8. How much customization was required in the case?
9. Did you perform some procedures and create new items within this casework that you had not previously done in other cases?
10. To what extent could you be creative in your work tasks?
11. Did you feel that you had the freedom to experiment within work tasks?
12. Did you feel that you were supported by the organization to challenge existing work methods etc.?
13. How did you take into consideration potential new technological developments for this case?
14. When facing a problem or an obstacle, how did you go about to resolve it?
15. How would you describe your responsibility over your tasks?
16. How often were you in contact with the client?
17. How did you interact with the team members within the casework?
18. How would you describe communication between the team members?
19. What kind of information did you share with the team members?
20. Based on the communication and information, did you feel you learned something new from other team members?
21. Was the work done more interdependently with other members or individually?
22. Were your and the other members roles in the case clear to you?
23. How did the team members collaborate with each other?
24. How would you describe the trust between the team members?
25. To what extent did you feel you had full decision-making power on the actions you took?
26. Did you feel that all team members were working towards the same goal?
27. How flexible was your working?
28. Did the case team stay the same throughout the journey?

29. What issues do you consider important for being successful in the software industry?
30. How do you see your company differentiating from your competitors?

Is there anything you would like to add or discuss that did not come up earlier?