

ISRN LIU-IEI-FIL-A--12/01263—SE

A Multiple Case Study on Contradictions and Pre-conditions for Outsourcing Agile Software Development Projects

Maksim Buslovič

Samson Deribe

Spring semester 2012

Supervisor: Jonas Söderlund



Master of Science in Business Administration;
Strategy and Management in International Organizations
Department of Management and Engineering

Abstract

Title: A Multiple Case Study on Contradictions and Pre-conditions for Outsourcing Agile Software Development Projects

Author: Maksim Buslovič & Samson Deribe

Supervisor: Jonas Söderlund

Date: May 30th, 2012

Background: In today's turbulent business environment organizational success depends on its ability to embrace change and adapt quickly. The ability to satisfy customer is core to profitability; thus being agile is a prominent factor, because customer expectations are never static. One of the project management methods which is quite popular in the software development are is Agile Method. Agile methods depart from the classical project approach as it emphasizes more on interaction among participant, short iteration and continuous feedback to embrace the continuously evolving customer requirements. However, implementing Agile methods in a distributed project work seems to be challenging, thus limiting projects to optimize form their distributed resource as well as external parties.

Aim: The purpose of this thesis is to understand how Agile methods contradict with the Outsourcing practice, which ultimately lead to identifying the possibilities to successfully outsource project work based on Agile methods. In addition, the study aims in providing a good ground for future study in 'Outsourcing within Agile Methods' to fill the big theoretical gap identified in the area.

Methodology: The thesis used a qualitative approach that intends to build theory through iteration by waving back and forth between data and literatures in an inductive manner. The research design was based on multiple case study that used five interviews and one direct observation as an instrument to collect primary data along with secondary data; all three together ensure proper triangulation resulting in higher research validity. Open coding system was used to analyze data; and findings were presented by tables, figures, models and direct quotations.

Results: The study shows that values and principles of agile software development which gives much emphasis on proximity in order to have a daily stand-up meetings, visualizations, constant contact with customers and other team members, knowledge sharing and fun disappear when the project is involved in outsourcing partly the development process; because the need use more plans & documentation, clear contracts and less interaction implying a contradiction in both 'Agile methods' and 'Outsourcing' practices. However, the study identified preconditions that must be considered while involving in outsourcing part of Agile software development: Outsource only if the part to be outsourced is not related to core product; Put a complete team in a co-located manner; Minimize interdependence among distributed teams; and there should be enough time before delivering product to customers.

Keywords: Agility, Agile Method, Contradictions, Pre-conditions, Outsourcing

Acknowledgments

As the five months project of writing a master thesis on Agile method and Outsourcing comes to a end and we are about to graduate from Strategy and Management in International Organizations (SMIO) program from Linköping University, authors of this paper would like to thank particular people without whom this work and some of the best moments of life here in Sweden would not be possible.

First and foremost, we would like to thank God for everything and our families and friends for encouraging, supporting financially and being there all the time when we needed them.

We owe our deepest gratitude to Zenterio AB, and particularly Kaj Carelind, for his permission and cooperation to collect the empirical data through interviews and arranging observation. We would like to show our appreciation to Linda (Ericsson AB), Denis (GGA Software Services LLC), Fredric and Olle (Zenterio AB) who shared their precious time and valuable experience to contribute for our study by participating in the interview; and special thanks goes to the Scrum team in Zenterio for letting us attend to their daily scrum. In addition we would like to thank Frehiwot Mekonnen, Ekaterina Kalinina and Afshin Hemmati for establishing first contact with case companies.

We are heartily thankful for our supervisor, Jonas Söderlund, who has been guiding and supporting us during the entire research time and helped us develop a good understanding of the subject area. Moreover, it is an honor to thank all our teachers in SMIO program and University staff for the whole two year learning period and shared knowledge.

Finally, both authors would like to thank each other for the teamwork, without which this research would not be possible. As in every project, there were ups and downs, however through the support, commitment, patience and discussions, authors were able to produce higher quality research. In the end, we clearly see that a co-authoring enables us to learn from each other and provides reader with more interesting mixed flavor research.

Table of Contents

1. Introduction	1
1.1. Background.....	1
1.2. Problem area	4
1.3. Research purpose and questions	6
1.4. Delimitation	6
1.5. Structure.....	7
1.6. Target groups	7
2. Theory Review	8
2.1. Definition of Project and Project Management	8
2.2. Historical Overview of Project Management	9
2.3. Significance of Projects and ‘Projectification’ trend	11
2.4. Types of project, methods.....	12
2.5. Understanding Outsourcing	15
2.6. Globally Outsourced Software Development	18
2.6.1. Definition of Distributed Software Development.....	19
2.6.2. The Motives behind Globally Distributed Software Development	19
2.6.3. The challenges of Globally distributed Software Development	20
2.7. Agile Overview, Characterization and Principles.....	23
2.7.1. Agile Methods Background	23
2.7.2. The Agile Methods	25
2.7.3. The Agile Manifesto	25
2.7.4. The Agile principles.....	28
2.7.5. Definition of the Agile Methods.....	28
2.7.6. Characteristics of the Agile Method	29
2.7.7. Overview of the Popular Agile Methods	30
2.8. Challenges of Agile Software Development.....	38
3. Research Methodology	41
3.1. Research Approach.....	41
3.2. Research Design	42
3.3. Data Collection	45
3.4. Data Analysis and Presentation	47
3.5. Research Ethics.....	48

3.6.	Validity, reliability and standards of quality.....	48
4.	Empirical Data	50
4.1.	Case 1 - GGA Software Services LLC.....	50
4.1.1.	GGA Company Background.....	50
4.1.2.	Agile method in GGA.....	52
4.1.3.	Projects, Teams and Activities in GGA.....	53
4.1.4.	GGA's Outsourcing/Distributed team in Agile Development.....	53
4.1.5.	Benefits of Agile Development in GGA.....	54
4.2.	Case 2 - ZENTERIO AB.....	55
4.2.1.	Zenterio Company Background.....	55
4.2.2.	Agile method in Zenterio.....	57
4.2.3.	Projects and Teams in Zenterio.....	58
4.2.4.	Activity Coordination within Agile Projects in Zenterio.....	60
4.2.5.	Zenterio's Outsourcing/Distributed team in Agile Development.....	62
4.2.6.	Benefits of Agile Development in Zenterio.....	63
4.3.	Case 3 - Ericsson AB.....	64
4.3.1.	Ericsson Company Background.....	65
4.3.2.	Agile method in Ericsson.....	66
4.3.3.	Projects and Teams in Ericsson.....	66
4.3.4.	Activity Coordination within Agile Projects in Ericsson.....	67
4.3.5.	Ericsson's Outsourcing/Distributed team in Agile Development.....	68
4.3.6.	Benefits of Agile Development in Ericsson.....	69
5.	Analysis	70
5.1.	Using word clouds to analyze empirical data.....	70
5.2.	Drivers for introducing Agile method.....	71
5.3.	Emphasized Agile values and practices compared to Outsourcing trend and challenges.....	73
5.3.1.	Stand-up meetings.....	74
5.3.2.	Burn-down charts and visualization.....	75
5.3.3.	Constant contact with customer.....	75
5.3.4.	Co-located teams.....	76
5.3.5.	Knowledge sharing and focus.....	77
5.3.6.	Fun and Developer involvement.....	77
	Towards framework.....	79
5.4.	Outsourcing implications on Agile development Method.....	82

5.5.	Preconditions for Outsourcing Agile Projects	83
5.5.1.	Not Product Related Development	83
5.5.2.	Complete Team along with a Project Manger	84
5.5.3.	Minimizing Interdependence among Teams.....	86
5.5.4.	Availability of Ample Time before Delivering Product to Customers	87
6.	Conclusion	89
	Closing Comments	92
	Implications	92
	Limitations.....	93
	Personal Reflections	94
	List of References	vii
	Appendices	xvii
	Appendix 1: Scrum meeting observation report, Zenterio AB	xvii
	Appendix 2: Interviews Guide.....	xxi

List of Figures

Figure 1 A cascade of development objectives (Source: Turner & Simister, 2000, p. 9)	12
Figure 2 The goals and methods matrix (by Turner and Cochrane 1993 c/f Turner & Simister, 2000)...	13
Figure 3 Approaches to managing a project (Wysocki & McGary, 2003, p. xxvii).....	14
Figure 4 Number of outsourced activities versus firm performance (Adopted: Mol, 2007).	17
Figure 5 The XP project lifecycle (Source: http://www.agilemodeling.com)	32
Figure 6 Scrum Foundation.....	35
Figure 7 Scrum guidelines (Source: http://softhouseeducation.com)	37
Figure 8 Sprint Burndown Chart (Source – www.joel.inpointform.net).....	38
Figure 9 Positivistic-interpretivistic research philosophy scale (Adopted from Blumberg et al., 2005)..	43
Figure 10 Research question sequence (Adopted from Blumberg, Cooper, & Schindler, 2005).....	44
Figure 11 GGA Software Services logo and slogan (source - official web page, see ref.).....	50
Figure 12 Pictures of St. Petersburg facility (source - official web page, see ref.).....	51
Figure 13 Zenterio Historical path visualization (source - official web page, see ref.)	55
Figure 14 Steps of Agile development Methods (Source: From interview transcript with Kaj, Zenterio)	58
Figure 15 Ericsson R&D base in Linköping, (source - authors picture).....	64
Figure 16 Ericsson main business areas (source – Ericsson annual report 2011, p.5).....	65
Figure 17 Visualization of question asked and answers received (Word cloud tool: www.wordle.net) ...	70
Figure 18 Criteria determining Outsourcing level	79
Figure 19 Criteria determining Agile set-up level.....	80
Figure 20 A two dimensional Model of Agile-Outsourcing levels	81

List of Tables

Table 1 The Agile Manifesto Value Emphasis (Source: Beck et al., 2001).....	2
Table 2 Advantages and disadvantages of outsourcing (Adopted from - Mol, 2007, p. 23)	16
Table 3 The Environment - Stakeholder matrix (Source: Chen 2004, p.20).....	25
Table 4 Limitations of agile method in outsourcing (Adopted from Turk et al., 2002).....	40
Table 5 Drivers for using Agile method in case firms' projects	73
Table 6 Identified Agile practices and following contradictions with outsourcing trend	78

1. Introduction

1.1. Background

This research mainly centers around the relatively recent field of study and practice of project management called Agile Methods, which are common in software development; and investigates what implications could be found in outsourcing part of development process.

Change and change management is a widely researched topic by many management scholars and researchers who emphasize its inevitability and continuous existence. Many discuss change as being ongoing and never-ending process of organizational life (Van de Ven & Sun, 2011) and discuss why and in what ways organizations change (Meyerson, 2001; Van de Ven & Poole, 1995; Whittington et. al, 1999).

It is also often mentioned that Agility is an essential trait of a successful organization in facing environmental turbulence and change. An agile organization is often described as one which embraces change by adapting quickly and effectively to foresee or even initiate and take advantage of change, yet remains resilient enough to absorb any setbacks as a result of this change. A report by the Economist Intelligence Unit Limited (2009) stated as follows '... in today's globalised, free-market environment, the ability to satisfy customer expectations is core to profitability. If you're not agile, you can't do it, because customer expectations are never static'. O'Riely et al (2009) also argue that in times of hyper competition and turbulent environment, neither strength nor intelligence guarantees survival but rather the ability to adapt to such changes.

While Change and Change management through agility are ever-existing discussions in organizations, we find it even more interesting topic while dealing with project based organizations. On the one hand, projects are temporary in nature and involve a predetermined set of goals, deadlines and budget; they might present their own set of challenges in dealing with organizational changes. On the other hand, as projects are more autonomous and flexible they might present a more adaptive nature in handling and responding to change.

Sharp et.al (2000) argues that agility cannot be achieved unless organizations transform in to team based organizational structure. Teamwork allows organizations to be better responsive to changing competitive environment (ibid). Particularly, when looking into the software development projects, the majority are using catch words such as Scrum and Extreme Programming (XP), which are categorized as an agile project management as suitable frameworks to manage projects in the face of changing customer requirement.

Agile software development represents a major departure from traditional, plan-based approaches to software engineering (Dybå & Dingsøy, 2008). Agile is referred as an umbrella term that consists of different approaches, methods, and techniques making

use of short iterations and continuously getting feedback from customers in order for the project team to evolve the changing customer requirement (Moreira, 2010). Agile Method is characterized by self-organized teams with high autonomy, shared vision, rapid pace processes and less emphasis on detailed planning (Cohn & Schwaber, 2003). ‘A cohesive team is, very possibly, the key between success and failure in the agile environment.’ (Chin, 2003)

Most project, particularly those with a co-located team and work set-up, have been influenced by the theory of proximity. Gestalt’s Laws of Perceptual Organization provides us the Law of Proximity stating that ‘objects near each other tend to be together’. The proximity of team members has an important implication of collaborative works within team. Proximity of team allows team members to have first-hand and real-time observation to project work progress and facilitate the initiation of work sequence (Hoegl & Proserpio, 2004). The agile methods as well place proximity at the centre of their software development projects and conduct the processes by bringing cross functional specialists into a co-located set-up and coordinating tasks through collaboration of their autonomous teams, face-to- face interaction, light weight documentation and tacit knowledge transfer; the nature of such project works constrain project tasks to be done in small sizes and limit the option of distributing works outside the wall (Cockburn & Highsmith, 2001; and Cohen et al., 2004).

The ‘Agile Manifesto’ which was written by the practitioners in 2001, proposed that agile development should put more emphasis on the four core values agile Manifesto gave emphasis to values in the items on the left side over those on the right shown in the table below:-

Table 1 The Agile Manifesto Value Emphasis (Source: Beck et al., 2001)

Agile Manifesto Value	
<i>Individuals and interactions</i>	Processes and tools
<i>Working software</i>	Comprehensive documentation
<i>Customer collaboration</i>	Contract negotiation
<i>Responding to change</i>	Following a plan

Agile projects are highly praised by customers because they allow the customers to have constant control over development processes (Dybå & Dingsøy, 2008) and reduced time and cost to market (Ribeiro & Fernandes, 2009).

The above discussion highlights the extent to which firms’ profitability and their ability to gain competitive advantage in a continuously changing environment depends on the teams and individuals employees. ‘Any temporary organization needs to be designed by and around people. Temporary organizations are naturally also dependent on the

will, commitment and ability of individuals for their creation, development and termination.’ (Lundin & Söderholm, 1995, p.5). The agile methods mostly rely on people and their creativity along with their interaction with their teams than on firms’ tools and processes (Melnik & Maurer, 2002; Nerur, et al, 2005; and Moreira, 2010).

According to Dhar and Balakrishnan (2006) many software developing firms kept on going into global software development that involve outsourcing part of or the whole development process. To start with, there are a number of reasons and trends, which explain why the topic of “Outsourcing” is becoming so important these days. According to Gilley and Rasheed (2000) organizations are increasingly using outsourcing in an attempt to enhance their competitiveness. On the practical level it results in that “The business landscape is currently witnessing widespread migration of service functions from developed nations like the U.S to several foreign destinations as India, China, Ireland, Philippines etc.” (Kedia & Lahiri, 2007) However, it should be admitted that in the search of competitiveness, firms collaborate and outsource functions or tasks not only to offshore destinations, but for local specialized service providers as well.

It is a big challenge to come up with the ultimate definition of what is “Outsourcing”, since the perception might change depending on the level of analysis. We agree with Gilley and Rasheed (2000) stating that, while the definition is “somewhat uncertain”, many potential benefits are identified, including improved financial performance and various effects, like focusing on core competencies. Dhar and Balakrishnan (2006) also concluded in a similar way and added benefits of outsourcing as ‘improved productivity, higher quality, higher customer satisfaction, time to market, and ability to focus on core areas are some of the benefits of outsourcing.’

The importance of research on interaction of IT related issues and outsourcing topic is big. Not only is it historically related as Mohammed et al. (2008) suggest – “...outsourcing was pioneered by IT-enabled services originations...”, but the extent to which society, markets and technology are inter-connected is increasing with progression. Carr (2003) notes that “Today, no one would dispute that information technology has become the backbone of commerce. Hardly a dollar or a euro changes hands anymore without the aid of computer systems.”(p.5)

Some authors believe that cost is the main reason for outsourcing IT related issues – “In many large organizations, IT outsourcing is being considered as a viable cost reduction alternative” (Dhar & Balakrishnan, 2006, p.5). While others propose, that in order to succeed with complex software development projects, companies outsource parts of work in order to get access to unique skills of their partners – ‘Vendors have more expertise than their individual clients because they face more-varied issues. They also can have employees specialize in areas clients typically encounter only once’ (Barthelemy, 2001, p.60). Mohammed et al. (2008) suggest that outsourcing has ‘matured from its initial avatar of a cost reduction tool, to a more mature and strategic role’ (p.381). Meanwhile, there is a third opinion that if taking in account strategic

level, IT has lost its importance – ‘As information technology's power and ubiquity have grown, its strategic importance has diminished.’ (Carr, 2003, p.5)

As it is possible to see from above mentioned, the research on the boarder of general outsourcing and IT industry is developed and varies, but there are issues which are only partly addressed. Grover et al. (1996) suggests that studies that examine different variables, what in today’s context includes project based work methods, can “facilitate development of a theory of IS outsourcing.” The particular need and reason of going on the project level, identifying challenges and patterns of possible ways of meeting them in Agile projects can be supported by Mohammed et al., 2008 idea that “..the principal barriers as well as enablers for effective outsourcing and realization of the targeted benefits are “operational” factors.” (p. 381)

In the above section we depicted the two issues, Agile methods and outsourcing, from a theoretical angle. In summary, Agile project method on the one hand is mostly about delivering more value to customers through regular collaboration and frequent feedback (Bohem, 2002) which ultimately leads to achieving competitive advantage. On the other hand, as Karolak (1998) explained; globally distributed projects are more challenging than even complex in house projects, but many are involved in the global collaboration with the motive of cost reduction. Moore et al. (2004) also added in to the argument by stating offshore outsourcing and agile developments respond better to customer requirement at a lower cost. In the following part we will further discuss the problem area of theoretical discussion on Agile projects and Outsourcing in order produce relevant research questions.

1.2. Problem area

In the past decade business environment is characterized by fast changing technology, changing customer requirement, continuous business development, higher customer awareness, shorter lead-times and intense competition (Cooke, 2010; Gunaskaran, 2001; and DeVor, et al., 1997). Consequently the demand for speed in the workplace with a faster decision making is required if firms to survive in the high velocity environment (Kessler & Chakrabarti, 1999). This critical speed requirement can only be achieved if firms are able to obtain agility. Sharp et.al (2000) argues that agility cannot be achieved unless organizations transform in to team based organizational structure. Chin (2003) noted that it is the so-called ‘soft skills’ which are critical denominator of agile team members. The ‘soft skills’ include; the team members ability to create and maintain relationships; better interact with various stakeholders; demonstrate flexibility, adaptability and being a team player. According to Williams and Cockburn (2003) Agile software development is all about ‘feedback and change’ and the agile method is developed to ‘embrace rather than reject’ change in customer requirements.

As mentioned previously the ideal agile project team is small and co-located. This set-up allows team to communicate on a daily base, face-to-face, with an ease to walk over to a colleague desk to discuss issues (Cockburn and Highsmith, 2001). Agile projects

face challenges when there is large scale project that require bigger team size and distributed team. The challenge of distributed team exacerbate if it involves outsourcing the software development to a geographically dispersed areas.

Referring to Festinger et al. (1950), proximity theory is a means by which conformity to agreed-up on norms, information sharing and cooperation. The more geographically team members are dispersed the less social integration they will have and that will apparently lead to lesser opportunity for knowledge acquisition from team (ibid). 'Distributed Work environments are characterized by the lack of proximity between co-workers' (Marlon et.al, 2009, P.308). Distributed team members of projects create a barrier in quantity and quality of communication (Allen 1980; Jain & Triandis, 1990), making the integration and coordination effort more difficult (Keller, 1994 and Meyer, 1993). According to Meyer (1993) there is a greater opportunity for learning among team members (Katz & Tushman, 1979) through frequent face-to-face communication as it provide rapid feedback, and enabling decoding and blending of complex information.

Despite the mentioned challenges of distributed teams and project works, there has been a great trend by large organizations outsourcing their IT functions in search for lower cost, high customer satisfaction, higher quality, improved productivity and focus on their core areas (Dhar & Balakrishnan, 2006). Many software developing firms kept on going into global software development and involved in outsourcing part of or the whole software development process. One among the many reasons for firms to outsource their software development is for cost efficiency. According to Williamson (1985) transaction cost theory depends on the parameters of 'production cost' and 'transaction cost'; that help the firm to make sound decisions which enables developing software at a lower cost and maximize the value of customers. Hence, outsourcing the software development is pretty much influenced by the transaction cost theory (Williamson, 1985); which in turn gives an implication on gaining competitive advantage through creating value to customers (Grant, 2010). Herbsleb and Moitra (2001) added into the argument that factors such as: market proximity advantage, improved time-to market through 'round-the-clock' development and global resource pool for competitive advantage as driving factors for software development to be global.

Agile projects with their natural set-up of being small size and co-located has shown a great deal of evidence for responding to the high-velocity changing environment of software development. Eventually there is no single agile project that can accommodate a large number of team members and co-located in one place (Cohn & Schwaber, 2003). Hence, with a requirement for large software development projects, the need to have more than the ideal team size (seven plus or minus two) and the need to have geographically distributed team also raise (Abrahamsson et al, 2003; Meso & Jain, 2006). Outsourcing as one feasible option of distributed software development and very popular in the IT industry poses many challenges to the Agile software development. In

this study the terms outsourcing and distributed team have similar sense, because our intention is to look what implications does outsourcing part of a software development project can have on the agile software development, when teams are not in the same area. Hence, outsourcing and distributed teams represent the sense of not being co-located in the same place.

Despite the many studies which are evolved in the area of agile method, there are still gaps regarding agile methods in an outsourced set-up. One of the reasons for the theoretical gap in the area of our study is related with the fact that agile development is driven by the practitioners (Beck et al., 2001; Cockburn, 2002a; and Dybå. & Dingsøyr, 2008) and the academic world is trying to catch-up with the newly emerging methods of managing projects. Hence, previous literatures focused on identifying the challenges of outsourcing (distributed development) in agile software development, but we found gaps in the literature regarding the two crucial but contradicting issues: Agile Methods and Outsourcing (Turk et al., 2002). Following the same line of arguments this study will investigate two gaps; firstly, how outsourcing practice contradicts with the values and principles of agile software development; secondly, what factors must be fulfilled in order to successfully outsource agile software development projects.

1.3. Research purpose and questions

The general purpose of this research is to provide an understanding to the nature of Agile Methods and Outsourcing (distributed development) in a software development projects; through developing some main themes that could be used as an improvement in the area of software development projects. In a more practical way, the research is about exploring how firms succeed in using Agile project method and at the same time outsourcing process within same project work. In order to address the objective stated above we developed two research questions as follows: -

The two research questions are:

Question 1: How do in practice, Agile values and principles contradict with outsourcing trend?

Question 2: What are the pre-conditions for successfully outsourcing Agile projects?

1.4. Delimitation

This research is limited to only software development projects that are using the Agile method and that involves partially or fully in outsourcing its development processes. It is not also the interest of this research to go further into the technical (such as programming or testing) challenges and that agile projects face during the process of the development.

1.5. Structure

The research consists of six parts; firstly, Introduction- this part discusses the general background of the thesis along with the problem definition, purpose and research questions. Secondly, we dealt with the conceptual framework which brings relevant literature to the subject of research. Thirdly, the research method was portrayed and it served as a guideline though which the entire research is conducted. Fourthly, The Empirical part presents the data gathered from the different cases in coherent manner. Fifthly, it is the Analysis part which blends theory with empirical data, so that we can make sense out of it to answer the research questions. Finally, it is the Conclusion part which presents the main findings in a comprehensive manner.

1.6. Target groups

The target group of this master thesis work are both academic researchers and practitioners. On the academics' part, the paper will contribute with further developing on concept of integrating outsourcing in the Agile project management method. In addition to that, the paper will provide useful information, for future research in the topics of project management, Agile method and outsourcing. For the practitioners, the research can serve as a source of inspiration in experimenting with Agile method or as a valid source of comprised outsourcing related issues in Agile set-up.

2. Theory Review

To start with, this paper rests on the two solid blocks of Business Administration area, these are Project Management in general, and Outsourcing (work distribution) concept. It would be unwise to get a reader in any particular discussion before thoroughly examining these two areas. The following theory review part, Chapter 2, is aiming to bring definitions, historic perspective, classic and current debates in both areas of Project Management, in particular Agile method, and Outsourcing, including software development industry. This chapter also helps authors to link Project Management and Outsourcing with particular Agile method project set-up and IT related work environment, which are the field of empirical study in the following Chapters of the paper.

2.1. Definition of Project and Project Management

“We use project management because we have faith that it works.”

- Turner & Simister (2000, p.2)

Even though Project Management is a well-researched area, it would be hard to find two similar definitions either from two random textbooks from the library shelf or recent academic articles. However, the differences in definition serve as a proof how multi-dimensional is Project Management field and that different contexts might be applicable. Lenfle and Loch (2011) refer to The Project Management Institute, highly recognized body by academics and professionals, for the definition of Project Management. The Institute defines project management “as the application of knowledge, skills, tools, and techniques to project activities in order to meet the “triple constraints” of scope, time, and cost.” (p.32) This broad definition uses “triple constraints”, or other well-known term – “iron triangle”, as an umbrella concept, stressing that a project is something what is based and limited by three key pillars of scope, time and cost.

Project Management Body of Knowledge Guide, a highly recognized source of standardized concepts for Project Management, issued by the above mentioned Project Management Institute, suggests that the concept of Project might be better understood by the differences it has if compared to the classic functional work division:

“Projects and operations differ primarily in that operations are ongoing and repetitive, while projects are temporary and unique. The objectives of projects and operations are fundamentally different. The purpose of a project is to attain its objective and then terminate. Conversely, the objective of an ongoing operation is to sustain the business. Projects are different because the project concludes when its specific objectives have been attained, while operations adopt a new set of objectives and the work continues.” (PMBOK-Guide, 2004, pp. 6-7)

Some authors probably overemphasize the importance of triple constraints in defining what is and what not a project is. Meredith & Mantel (2010, p.37) suggest that “perhaps up to 80 percent of all “projects” are not actually projects at all, since they do not include the three project requirements for objectives, budget, and due date.”

Gower Handbook of Project Management is a rich source and many authors contributed the work, that is why it is not surprising that some alternative ideas are presented – for instance the notion of novelty is presented: “Thus projects are novel endeavours, undertaken to deliver new development objectives. They are necessarily transient, since if they become established they become part of the routine operations of the organization” (Turner & Simister, 2000, p. 65). This idea of project work in longer run contributing to the routine operations of organizations is different from the above presented PMBOK Guide description of clear cut difference of project and function. Despite that difference both perceptions are possible, meanwhile they emphasize that Project might be seen differently depending on time horizon.

2.2. Historical Overview of Project Management

The aim of this section of paper is to provide a reader with some historical background regarding Project Management field. Lenfle and Loch (2011) stand as primer source with their recent and very critical paper upon Project Management history and lost novelty. This section is important so that the reader could better understand why and how the changes in Project Management resulted with introduction of Agile methods; how it corresponds with trends related to Outsourcing.

Even though, the construction of Egyptian pyramids is often provided as a classic example of how old and useful the concept of Project could be, many authors agree that the routes of modern Project Management can be found in the 1940s and be related to the Manhattan Project, which resulted in developing first atomic bomb, as well as different ballistic missile projects (Atlas, Polaris, Apollo, etc.) run by US government in the tense periods of global history (Lenfle & Loch, 2011; Meredith & Mantel, 2010; Martino, 1964). To emphasize and remind this historical sequence of events Meredith & Mantel (2010) stress that “business and other organizations learned from government, not the other way around.” (p. 1) One single project on it’s own – Polaris, had a huge impact on current state of Project Management field:

“The Polaris project developed the first submarine-launched ballistic missiles (SLBM) carrying nuclear warheads. These offensive weapons, almost impossible to track and attack, became a key element of nuclear deterrence. The Polaris project is today credited with developing the “scientific approach to project management” with the first large-scale application of computerized planning techniques, particularly PERT (Program Evaluation and Review Technique), a formal planning method with computerized flow charts.” (Lenfle & Loch, 2011, p. 37)

According to Wysocki & McGary (2003) the next step and a big sign of coming change in development of Project Management were early 1960s. The use of computers spread to the business areas, such position titles as “programmer, programmer/analyst, systems analyst, and primitive types of database architects” (p. xxiii) emerged. As we can conclude with an option of retrospective view, these changes in area of technology had a great impact on development of Project Management; internationalization, globalization and outsourcing; future of markets, trade and technology. Such technological and economical boost was something contrary to the mindset of some researchers of the end of 19-th century. As Martino (1964) refers to one of that time authors, who in 1891 wrote that “in his estimate, all scientific measurements would be completed by the turn of the twentieth century.” (p.8). In other words, the span of technology, had a huge impact on society development, surely including Project Management. In that context, the research conducted in this paper is meaningful, since it combines study of classic Project Management and Outsourcing field in the extremely high-tech state-of-the-art software development industry.

Keeping in mind that 1960s were the starting point when business could contribute its processes by Project Management techniques, in general Project Management was still at its premature stage: “the discipline was still in its infancy—it lacked a recognizable academic status as a field, and it also lacked professional recognition; project manager was still a new role.” (Lenfle & Loch, 2011, p. 41) According to the same source, “the institutionalization process” started with establishment of the U.S. Project Management Institute in 1969.

But it wasn't only a smooth sailing in the 1960's as Lenfle and Loch (2011) suggest. According to them, waterfall model of Project Management was preferred by the U.S. state authorities, due to the doubtful low cost, control and visualization motives. The concurrent methods of 1950s were perceived as non-optimal and risky. As Lenfle and Loch (2011) describe, it was Bernard Schriever who called the approach as a “concurrency” – simultaneous (or overlapped) performance of logically sequential tasks. Such method in fact has positive aspects:

“The advantage of a parallel strategy is not only the time but also the information gained from the trials, even if they are ultimately abandoned. The result may be a better end result and, in addition, a lower cost (which many managers find counter-intuitive) stemming from the better design ultimately chosen.” (Lenfle & Loch, 2011, p. 38)

At the same time waterfall model was accused of “too little duplication; too little competition; premature, optimistic, and over-detailed advanced system requirements; excessive centralization of decision making; premature commitment of large funds; and too little emphasis on the early stage of R&D.” (Lenfle & Loch, 2011, p. 44)

However many changes occurred as we have entered the era of 1990s. According to Leonard-Barton; and Angelmar “Parallel trials and iterative experimentation were also rediscovered in the innovation domain, but took longer—well into the 1990s.

Experimentation was revived by innovation researchers, who referred to it using terms such as “product morphing,” “probe-and-learn,” or “agility.”” cited in (Lenfle & Loch, 2011, p. 46). The method, was not the only change in 1990s, projects went from construction and engineering into new industries of systems and information technology. (Lientz & Rea, 1999) According to (Lientz & Rea, 1999) business process, change, global competition and outsourcing were some key trends in business in the end of 1990s. We can observe that since decade has passed, these trends are still among most important ones.

To sum up this part of paper, we can conclude that Project and Project Management went through the series of changes linked with social, technological and possible political development. Agility and Agile method is one of the latest evolutionary spins in the Project Management, that is why on it’s own it is a point of strong academic interest. Some voices have been raised on the future of Project Management: to finally overcome one-size-fits-all belief by introducing different types of projects; link Project Management with strategic decisions and innovation. (Lenfle & Loch, 2011; Wysocki & McGar, 2003) Finally, heavily cited Lenfle & Loch (2011) proposes that Project Management should play crucial role for organizations, especially in the “aftermath of the economic crisis of 2008.” (p. 51).

2.3. Significance of Projects and ‘Projectification’ trend

“We are able to achieve goals through project organization that could be achieved only with the greatest of difficulty if organized in traditional way.”

- Meredith & Mantel, (2010, p. v)

In the Introduction of “Project Management and Control” book from year as early as 1964, Martino (1964) suggests that reasons why [project] management is a tough business are not only tight margins, but the fact that some issues are hard to control, these are: constant technological change, increased competition and shorter product lifespan. This type of reasoning regarding the importance of project management is still often used in our days, because it is still valid (Meredith & Mantel, 2010; Wysocki & McGary, 2003). Some authors go beyond the idea in change of environment to the change of deeper specializing in particular work, where project work is one of possibilities - “We are entering an era of hyper specialization – a very different, and not yet widely understood, world of work.” (Malone et al., 2011)

As a result of intensive integration of projects and project work in the firms, previously organized in the more of classical functional way, some authors have suggested the concept of ‘projectification’ (Midler, 1995; Maylor et al., 2006). For the past few decades there has been a clear evidence of growing interest by companies in organizing their resources in temporary projects (Weaver, 2007). The project structure is more favorable when there is high level of - uncertainty, task complexity, importance, time criticality is and technology is new (Youker, 1977). The ever growing use of projects reflects evolution in markets and technologies – projects are now used not only in

classic areas of construction, aerospace and shipbuilding, but in software development, finance, banking and education, if not in all industries. (Turner & Simister, 2000).

Large scale and complex projects, as many of IT projects, have high failure risk, but if achieved can reward at the same high level (Keil & Mähring, 2011). In general, and in particular within IT industry, projects include new technological breakthrough, what might be a challenge for the project team; moreover, such technological or business changes might happen during the project (Lientz & Rea, 1999). That is why the management of IT related projects is of high research importance.

2.4. Types of project, methods

“A mountain can be climbed from more than one side.”

- (Tonnquist, 2008, p. 60)

From the previous sub-chapters we know that Project Management has over than six decades of history and the recent importance rose due to the ‘projectification’ of organizations and general popularity of executing tasks through the project set-up. This part of paper will introduce a reader with some concept of how organization and project can be structured depending on the needs and method used. In addition, Agile method will be “located” with help of several developed models. Often it is not that easy to decide which structure or method is optimum due to the advanced technology impact on business. As Wysocki and McGary (2003, p. xxiv) propose: “Technology has put most businesses in a state of confusion.”

According to (Meredith & Mantel, 2010) organization can be classified in three main ways according to the way they run project, these are: functional organization, pure project organization and matrix organization. As we will see later in the section dedicated particularly to Agile method, it is most likely that a firm using Agile method is a pure project organization. However, before that in order to better understand the level of analysis, we suggest to use Gower Handbook visualization of project level in the firm (Figure 1), we can see that it includes: project itself, work area and team, but not firm’s development, programme of projects, nor individual level.

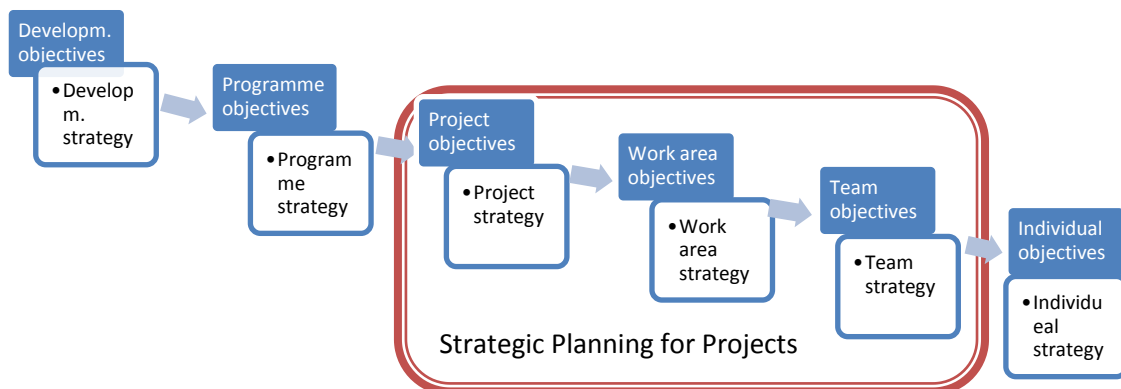


Figure 1 A cascade of development objectives (Source: Turner & Simister, 2000, p. 9)

As we touch upon the discussion on differences among classic and other methods of project we can turn to “The goals and methods matrix” developed by Turner and Cochrane in 1993, in Figure 2. According to the authors depending on projects with well-defined method and goal are very likely to succeed, while projects with poor method and goal are risky and often fail.

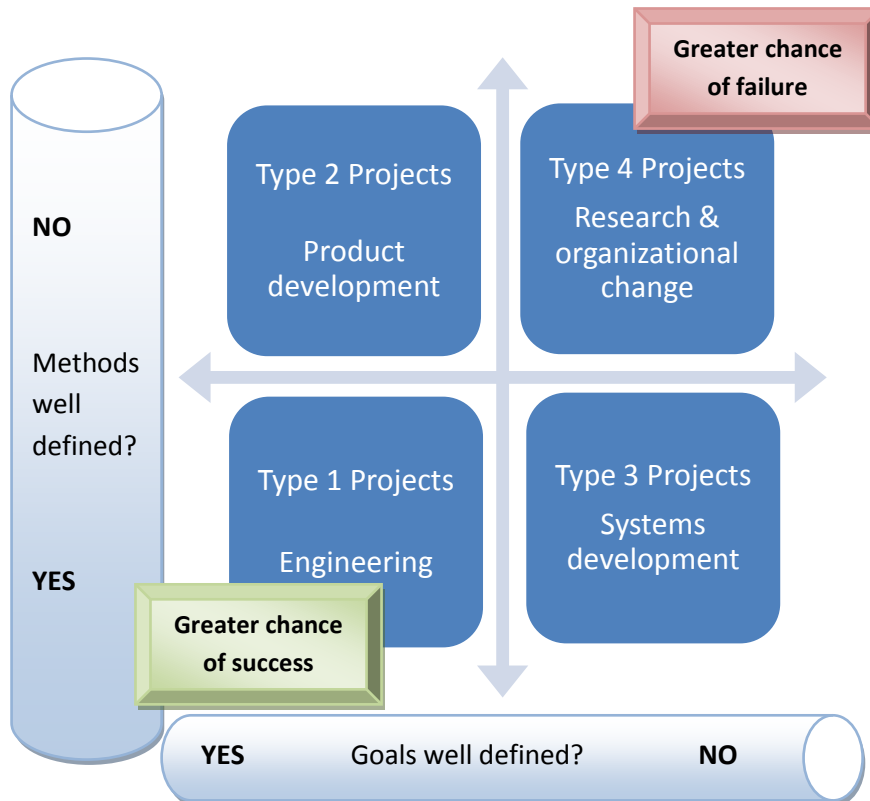


Figure 2 The goals and methods matrix (by Turner and Cochrane 1993 c/f Turner & Simister, 2000)

Type 3 Projects from the figure above, represent projects connected with software development, and in particular can include Agile method. Such projects “are planned in terms of life cycle stages. The goals are defined in conceptual terms, but their specification is refined through the stages of the project.” (Turner & Simister, 2000, p. 74)

Wysocki and McGary strongly emphasize the need for corporations to improve their project methods to more adaptive ones: “We contend that the traditional world of project management belongs to yesterday. “Change or die” was never a truer statement than it is today.” (2003, p. xxv) At the same time, same authors strongly claim that there is a need and a gap for a third type of Project method, since neither Traditional (or Classical) nor Agile method can suite particular type of demand – at that stage they introduce Adaptive Project Framework, which is in the middle of two radical types,

Figure 3. Despite the fact that in theory it looks reasonable, we were not able to find evidence that such typology was recognized in a broader way.

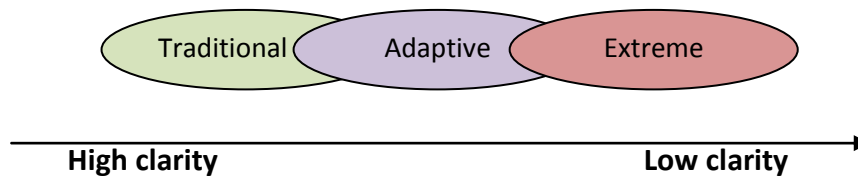


Figure 3 Approaches to managing a project (Wysocki & McGary, 2003, p. xxvii)

It has to be admitted that Traditional project management, despite many technological change and development of the field often works very well. (Meredith & Mantel, 2010; Tonquist, 2008; Wysocki & McGary, 2003). As for example, Tongquist (2008) points out: “To sequentially develop the different phases in the project is a very risk-free method” (p. 109). This method is also widely known as a Waterfall method, since one activity falls into the other and a whole process can progress only that direction.

“Modern project management has evolved from the need to bring integration and control to technically complex, multi-organization engineering and systems projects. Some of the discipline’s most fundamental principles are universally relevant, whatever the project. These include the concept of the project having life cycle and that of the project manager being the single point of integrative responsibility.” (Turner & Simister, 2000, p. 43)

Apart from Waterfall and Concurrent Engineering (Tonquist, 2008) suggests two more methods: Rolling Wave Planning and Dynamic Development. While Rolling Wave is a more flexible version of Waterfall method, Dynamic Development in fact stands for Incremental development or in other words Agile method. By some authors (Lenfle & Loch, 2011) it is been considered as a rebirth of Concurrent Engineering method of 1940s, but under requirements and technology by 1990s.

According to Parkinson’s Law - “Work expands to fill the time available”. In such perspective limiting the time to accomplish the project and concentrating on in fact fulfilling and only fulfilling customers’ demands might be reasonable. The fact that “the project duration is limited” is recognized as the major benefit of concurrent type of project. (Tonquist, 2008, p. 110)

To conclude, the essence of Agile method and other related Project Management techniques “are built on the assumption that there will be changing requirements as the client gains better focus on what they actually need. Sometimes those needs can be very different than the original wants.” (Wysocki & McGary, 2003, p. xxiv)

2.5. Understanding Outsourcing

“Outsourcing is a complex phenomenon that ought to stretch academics beyond a singular perspective.”

- (Mol, 2007, p. 189)

To start with, Mol proposes that outsourcing and international trade can be as closely related as “two sides of the same coin.” (2007, p. 14). International trade and globalization itself is a big and broad topic. The modern classic debate on ‘flat’ (Friedman, 2005) and ‘spiky’ (Florida, 2005) worlds alone is a proof of two different viewpoints upon globalization. In this research we see globalization as a one of the main enablers of outsourcing.

Researchers in Linköping University referred to transformation of national and global markets as a source for “creating new market conditions and breaking up established ways of developing and manufacturing products and systems” (Berggren et al., 2011, p. 5). Same authors continue with saying that “*Firms are rarely self-sufficient*” and this serves as a reason to integrate ‘complementary knowledge’ of other parties (p.148). Moreover, same authors point out that specialization and “reinforced tendency to focus on core competences” led to separations of tasks among different firms in supply chain and further specialization – this practice is valid in industries like financial services, software development and pharmaceuticals.

Even though outsourcing as a term was almost unheard twenty-twenty five years ago, now it is in vocabulary of regular people. Meanwhile, the nature of outsourcing evolutions and it becomes important phenomenon for research; other than classical context research is quite limited and should be performed in order to improve understanding of outsourcing, which is currently heavily affected by US automobile industry studies (Mol, 2007). Lei and Hitt cited in (Mol, 2007, p. 3) define outsourcing as “the reliance on external sources for the manufacturing of components and other value-adding activities.”

We can clearly see three waves of outsourcing: mid to late 1980s – wave of manufacturing outsourcing; early to mid 1990s – wave of IT outsourcing; and late 1990s to onwards – wave of business process outsourcing. (Mol, 2007) In all cases the objective of company has been to gain higher level of performance and increase in number of outsourced activities is likely to be explained by “belief that outsourcing does increase firm performance.” (Mol, 2007, p. 17) Surely, the ‘belief’ of increased performance was not the only perception, and a number of outsourcing advantages and disadvantages can be found in Table 2 below:

Table 2 Advantages and disadvantages of outsourcing (Adopted from - Mol, 2007, p. 23)

Advantages:	Disadvantages:
Strategic focus/reduction of assets	Interfaces/economies of scope
Complementary capabilities/lower production costs	Hollowing out
Strategic flexibility	Opportunistic behavior
Avoiding bureaucratic costs	Rising transaction and coordination costs
Relational rent	Limited learning and innovation

Additionally to three waves of outsourcing (Berggren, Bergek, Bengtsson, Hobday, & Söderlund, 2011) suggest three types of outsourcing: manufacturing outsourcing (reason – cost), component outsourcing (reason – supplier knowledge) and capacity outsourcing (reason – mix of previous two). By putting together both authors’ perspectives we can better see what the outsourcing action was and what the reason behind was.

Outsourcing according to Berggren et al. (2011) clearly shows that the notion of trade-offs in outsourcing is often forgotten and replaced by the advantages or drawbacks in make-or-buy decisions. The previously displayed table is not an excuse, single advantage and disadvantages do not form the understanding of what advantage triggers which disadvantage. Same authors suggest that neither trade-offs matter has been explored in antecedent literature. The authors of this thesis find the concept of outsourcing trade-offs very useful, interesting to check empirically and will come back to it in the next chapter. One of possible challenges identified is integration related issues:

“The reason why conflicts occur between outsourcing drivers is that even when an activity is externalized, that is, outsourced, it still has to be integrated across firms and fulfill a need in the buying firm.”
(Berggren et al., 2011, p. 242)

Apart from outsourcing advantages, disadvantages and tradeoffs some other issues might be touched upon. For example, (Chesbrough, 2006) introduces us with two syndromes relevant to highly technological industries and outsourcing related challenges – “not invented here” (NIH) and “not sold here” (NSH). NIH is an internal resistance based on xenophobia, it blocks internalization of ideas from elsewhere due to the fact that it is not from us, different and can not be trusted. NSH syndrome is based on mentality that if we don’t sell it, nobody should do it either, or if we do not find any use out of technology, no one can make it either. Then implementing outsourcing models, both NIH and NSH syndromes should be taken into account.

Chesbrough (2006) also goes beyond the border line between core and non-core activities and encourages firms to “use external ideas more in their own businesses and allow their unutilized ideas to be used by others.” (Chesbrough, 2006, p. ix) He gives an example of open source software where open innovation and proper business model can create significant value for many parties.

If we continue on analysis of IT related industry reactions towards outsourcing, Mol (2007) identifies that there are different social reaction and inertia level towards outsourcing in different departments of a company. Alike from Purchasing, which has positive attitude, or Accounting and HR, which have negative attitude, Information systems area might vary from negative to positive. The reason for that is that outsourcing might cause the rise of complexity in system in order to integrate data from suppliers. At the same time this might be a new and interesting challenge. This is an interesting point to investigate in our research – not only which contradictions are found, but to what extend in every case, firms and interviewees are optimistic about outsourcing in their project work.

Another line of argument by Mol (2007) is the introduction of ‘outsourcability’ dimension that is the concept of measuring how much activity is worth to outsource in respect to the change in firm’s performance. “Optimal performance is reached when all activities are correctly outsourced or integrated. The pattern this produces is a negative curvilinear relationship between outsourcing and performance, with the top of the curve presenting the performance optimum.” (Mol, 2007, p. 188) (see Figure 4) The optimum number of outsourced activities depends on industry and firm’s strategy, Mol (2007) gives an example of dotcom firms in the late 1990s, which outsourced too much and their value dropped; and an example of companies, which ‘against the trends of outsourcing’ and are able to construct successful business models - Zara, Ryanair and Southwestern.

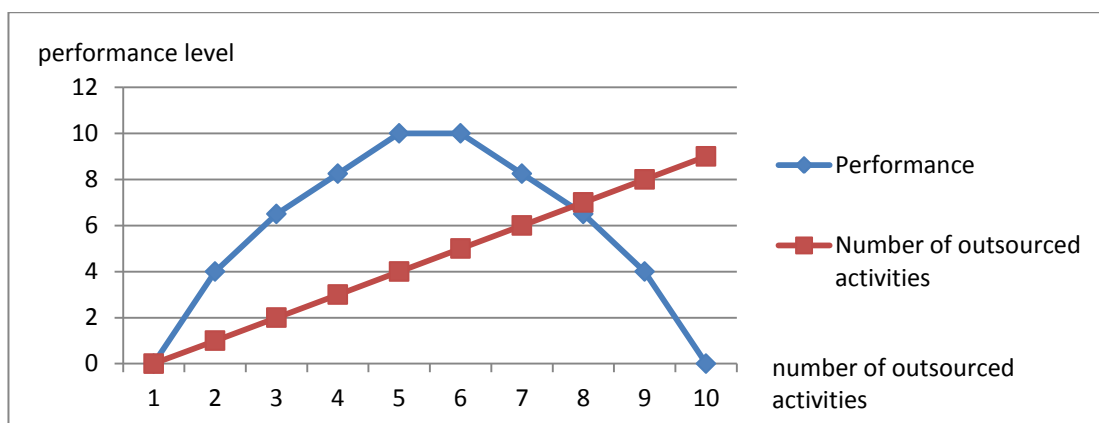


Figure 4 Number of outsourced activities versus firm performance (Adopted: Mol, 2007).

One of principal theoretical contribution of Mol (2007) is that he is able through a persuasive discussion to argue and claim that two important assumptions are “fundamentally incorrect”. First is related to the idea that non-core activities should be

outsourced – author suggests there cannot be only core and non-core activities in the company. Moreover, it is not a clear cut that it is most effective to get rid of non-core and retain core ones. Second, the grounds of “strategic outsourcing” are tested significantly, since Mol (2007) suggests that “Much of the rhetoric around strategic outsourcing seems to be driven by suppliers, such as Accenture, EDS, and IBM, wishing to improve their bargaining situation.” (p. 148). Many companies buy into the false circular reasoning – “Outsourcing is becoming more strategic; because it is becoming more strategic firms ought to outsource more; and by outsourcing more they actually make outsourcing more strategic.” (Mol, 2007, p. 151)

To conclude Mol’s ideas, it is very hard to link outsourcing decisions and performance outcomes, therefore outsourcing in it’s nature is an experiment; in large corporations the overall level of outsourcing and insourcing is never stable; and most managers start to realize that outsourcing ins not a panacea for all problems.

2.6. Globally Outsourced Software Development

One of the 21st century trends which can be recognized as a globalized economic world has brought significant changes in almost all industries. The above mentioned outsourcing trend by manufacturing industry has provided some evidence which lead the software industry to be involved in the globally distributed software development. Herbsleb and Moitra (2001) described the past decades as a steady and irreversible change in the globalization of businesses and particularly the high-technology businesses, bringing new issues in terms of competition and cooperation across boundaries. The way software is managed, developed and maintained has evolved from being concerted at a single location to being physically distributed across the globe and this phenomenon is referred to as “global”, “distributed” or “multi-site” software development (Sengupta et al., 2006). According to Lee et al. (2000) globally distributed software development started in early 1970s as a form of contract programming that could be considered as an outsourcing. In fact the globally distributed software development has become truly global in the 1990s with the revolution of PC (Carmel, 1999).

Over the previous few years, ‘Software has become a vital component of almost every business’ (Prikladnicki et al., 2004 p.1) in their day-to-day work, in some cases it is even considered as a source of competitive advantage (Carmel, 1999). Software developers have started to be aware of the direction through which globalization is heading and what implication it has in the success of businesses. This consequently brought the understanding of how globally distributed software development efforts should be tuned towards enabling multinational and virtual cooperation (Herbsleb and Moitra, 2001). Of course, the rapid growth of computer networks, telecommunications and internet technologies have provided a good ground to enable firms globally distributed software development. (Hossain et al., 2011)

2.6.1. Definition of Distributed Software Development

In dealing with distributed software development, for the purpose of this thesis, we picked-up the definition of Ågerfalk et al. (2005) on “development” which broadly refers to ‘any software development lifecycle activity’. This gives an extended activity beyond “pure” development and includes such activities as deployment and maintenance of software. In addition, this broad definition does not restrict the thesis analysis strictly to new software development. Ågerfalk et al described the term “activity” in a loose sense to include any individual or collective human action at any level of the development for a particular purpose. Referring to activity theory, Engeström (2000) defined an activity as something that transforms an input (object) into an output (outcome). Thus, ‘development activity’ can be considered as an individual (or collective) action that transforms something abstract (or concrete) into something of value (e.g. product) in the framework of a software systems lifecycle (Ågerfalk et al., 2005).

The notion of ‘distributed’ in the above definition implies that team members not being co-located or geographically spread out. For this thesis purpose we used the definition stated by Cohen and Mankin (1999) - classifying a project or development team that are globally distributed as, interdependent work group consisting of culturally diverse team members that reside in more than one country and implementing the global strategy of software development by the firm. Here, Ågerfalk et al. (2005) strongly argued that, the core aspects of distributed software development are not only related to the ‘geographical distance’, rather to what referred as ‘socio-cultural distance’ and ‘temporal distance’. According to Ågerfalk et al. ‘socio-cultural distance’ refers to the perceptual difference for different situations due to their socio-cultural background and belonging; whereas, ‘temporal distance’ referring to an experience of time dislocation by two a actors who wish to interact. Hence, understanding different dimensions of team set-up such as physical and temporal distance along with the cultural and organizational boundaries are important aspects in analyzing distributed software development.

2.6.2. The Motives behind Globally Distributed Software Development

The globally distributed software development allowed great number possibilities for companies (Sahay, 2003). For some authors like Carmel and Agarwal (2001), the drivers of moving into global software development are strategic and basically are divided into two critical aspects: cost advantage and large labour pool. Alike the early adopters of outsourcing manufacturing through relocation of production to cheap labour market in different developing countries, the software development swing to global level was in search for low cost. Moreover, the shortage of skilled labour in the software development put area put a lot pressure on firms to look further beyond their localities and access into a global resource pool. It has also been testified by Dhar and Balakrishnan (2006) that, most part of the outsourcing tendencies are in search of lower cost, but it is also claimed that the globalization effort goes outside the simplistic

‘manufacturing thinking’ of search for cheap labour and includes access to skilled resources across national boundaries (Barthelemy, 2001; and Herbsleb & Moitra, 2001).

Bartelt et al. (2009) claim that global software engineering promises can be boiled down to three main significant forces:-

- *Economically*: include Cost, particularly the dramatic cheap personnel costs, to be one of the reasons for developing in Asia. The increasingly global networks necessitate firms to develop complex software in an efficient manner to gain competitive advantage.
- *Organizationally*: in today’s globally acting companies with a global structure, distributed development seems to be natural style of project organization. Because organizational resources are spread all over the place and can better be tapped with global pool.
- *Strategically*: local presence by firms in different markets and producing localized software by the developers who know better about customers brings benefit in terms of alleviating cultural challenges and improve time-to-market. In addition, it supports psychological and political aspect by only employing local labour which in turn facilitates mergers and acquisition by meeting requirements from government (ibid).

Similar factors to the aforementioned factors above, by different authors, are also fuelled the trend as follows: capitalizing on global resource pool, business advantage of proximity to market, quick information of virtual team cooperation, pressure to improve time-to-market with the help of ‘round-the-clock’ development and the need for flexibility to capitalize on mergers and acquisitions opportunities (Herbsleb & Moitra, 2001; and Karolak, 1998).

2.6.3. The challenges of Globally distributed Software Development

Though the aforementioned drivers of globally distributed software development provided several opportunities, they also entail crucial impediment that require the developers and firms due attention. Karolak (1998) argued that that ‘globally distributed projects are more challenging than even complex in house projects’. According to Herbsleb et al. (2001) there is a strong evidence of survey showing that development tasks take much longer time than co-located tasks and communication and coordination are mainly accounted as a reason. The dispersed team member’s set-up in projects has different effect on the development of software at many levels (Herbsleb & Moitra, 2001). The globally distributed software development involves different stakeholders in terms of national and organizational cultures; separate location and time-zones, and intensive use of information and communication technologies (Conchuir et al., 2006). The difference in geographical, temporal and cultural aspects are found to have great impact on how distributed team members work together; such conditions apparently introduce challenges in relation to communication, coordination and control have been found (Ågerfalk et al., 2005). The co-located development

projects teams through shared processes, habits or vocabularies; usually over time build up communication and coordination mechanisms, but with global distribution of members these mechanisms suffer a lot and generate high demand for coordination and communication (Herbsleb & Grinter, 1999; Bartelt et al., 2009).

- *Communication* refers to an activity that involves sharing of complete and unambiguous information by the sender to the receiver with the purpose of reaching in a common understanding by both (Carmel & Agarwal, 2001).

Communication plays a major role in transferring knowledge and information between actors; it involves the facilitation of interactions through the use tools (Ågerfalk et al., 2005). In the ideal scenario, communication is to be best (Carmel E, 1999) when it is on its basic prototype form- ‘face-to-face’ (Clark,1996), however, due to the fact that global development consists of distributed team, the context of communication has to take a technology-mediated form and thus limited one (Ågerfalk, 2004). It is evident that using those technology-mediated communication tools over temporal distance creates significant delays in communication (Ågerfalk et al., 2005). Sometimes it is also difficult for ideas to put them in a written form, thus creates a risk of misunderstanding (Damian & Zowghi, 2002).

- *Coordination* is the act of integrating tasks that reside in different organizational units, so that all units contribute to the general objective. (Carmel & Agarwal, 2001)

There is always a coordination problem when two individuals have a common goal to achieve and when the action of one depends on the action of the other (Clark, 1996). Hence, all software developments have inherent problem of coordination. The nature of software development in itself is quite complex with less routine work and demands for costly coordination effort (Espinosa and. and Carmel, 2004). It becomes apparent that the coordination problem amplifies when the development activity is done with distributed team, across cultural border and over time and space (Ågerfalk et al., 2005). In distributed software development, demands for coordination among specialized individuals are greater due to conflicts in teams with inter-related tasks (Sidhu & Volberda, 2011). The distance also damage the feeling of ‘teamness’ (Battin et al., 2001; Herbsleb & Mockus, 2003) as team members may not be fully aware of each others’ remote work and consequently opening a room for misunderstanding and conflict(Pyysiäinen, 2003). Another obvious problem of geographically distributed team is trust; limited occasions of face-to-face communication and lack of common experience prone significant challenge such as on trust among team members, thus reducing the willingness of team members to communicate and collaborate (Pyysiäinen, 2003).

- *Control* is a means of holding on to the predetermined goals, policies, standards, or quality levels (Carmel & Agarwal, 2001). The process of control involves management and reporting means in place to make guarantee progress of activities in the software development. Control is one aspect of project

management where formalized structure is set to ensure the development of software in time, on budget and with the desired quality. (Ågerfalk et al., 2005)

Control has synchronization role to maintain consistency among project artefacts (Braun et al., 2003), hence, projects normally use centralized configuration management tools to facilitate their control processes. Pre-defined milestones and clear entry and exit criteria become vital to effectively implement concurrent engineering principles as the distributed development lack good tools that support collaboration across time and space (Herbsleb & Moitra (2001). The control problem may also emanate from the socio-cultural distance where employees in high-cost economies may feel insecure that their jobs are under threat from their colleague in the low-cost economies, thus developing a fence with ‘we’ versus ‘they’ mentality (Casey & Richardson, 2004). Moreover, in case of no clear definition as to where the centre of control power lies, political conflict may arise between different teams (Kiel, 2003). In distributed software development, control demand greater understanding of the different local formalized norm structures such as laws, tradition and regulation so that the process of control incorporate the perception of authority and hierarchy across different team environment and culture (Ågerfalk et al., 2005 and Krishna et al., 2004)

According to Herbsleb and Moitra, (2001) geographically distributed project team members have varied challenges and effects on many levels. Here are some of dimensions of the problems:-

- *Strategic issues:* Once the project site is determined, detail works division across the sites is difficult as it is dependent on the resources available at the sit and the level of expertise for various technologies. The ideal set up requires individual sites to operate on their own and provide an easy, flexible and effective communication among sites. However, the different models by each sites and requires different coordination mechanisms. Another critical challenge is misalignment between middle level and senior management due to (1) perceived benefit of the globally distributed development and (2) Individuals threat of job loss, control loss, fear of relocation and extensive travel (ibid).
- *Inadequate communication:* The fact that developers are not co-located the opportunity to have informal- ‘corridor talk’ is less and obligate the developers to rely on technology dependent communication modes like telephone, e-mail and VOIP. Especially at the early stages of the software development an intensive communication is required. Outsourcing also adds into the complexity when the involved parties use filtered communication due to fear of losing intellectual property (ibid).
- *Knowledge management:* Projects that are engaged in product development need to integrate the wide range of specialized knowledge bases, because the integration of such dispersed knowledge improve the development performance and is vital for competitiveness (Enberg et al., 2010). In a distributed development set-up, unless there is effective information and knowledge-

sharing mechanisms, managers may not be able to grasp the fruits of globally distributed software development (Herbsleb & Moitra, 2001).

Software development projects as part and parcel of an organization are affected by changes that occur around their environment. The centre of our thesis, agile methods, deals mainly with change in software development requirements during the implementation of the projects. Hence, sound understanding of what agility and Agile method means and its typology is needed, thus these issues will be discussed in the subsequent parts.

2.7. Agile Overview, Characterization and Principles

The purpose of this section is to introduce the concept of agility and agile method by going through theoretical literatures on software development to better understand what characterizes agile methods and the challenges associated with it.

2.7.1. Agile Methods Background

Firms' ability to survive in the turbulent business environment depends on their change tolerance capacity (Charette, 2003). Charette characterized change tolerance as firms' ability to continue operating well in the face of high market turbulence; where the market turbulence can be defined as some kind of instability, lack of control and uncertainty, in the firm's environment. He further highlighted that; firms by being change-tolerant can adapt to and cause market turbulence for their own benefit (Charette, 2003). Different authors consistently argued that business agility is the better way of handling the challenges of change. "... in today's globalized, free-market environment, the ability to satisfy customer expectations is core to profitability. If you're not agile, you can't do it, because customer expectations are never static." (Economist Intelligence Unit Limited, 2009, p.2).

The survey conducted by Economist Intelligence Unit that included responses from 349 business executives around the world, to assess the challenges and rewards of agility has presented the following findings: nearly 90% of the survey shows organizational agility is a core differentiator in today's rapidly changing business environment; Yet most companies admit they are not flexible enough to compete successfully; more than 80% respond that Internal barriers such as slow decision making, conflicting departmental goals and silo-based information stall agile change efforts and finally the survey also emphasized the importance of technology in enabling organizations to be more agile.

Referring to oxford English dictionary Agility is 'the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion'. The rapidly growing and volatile software industry along with the constantly improving mobile application environment are demanding firms to deliver faster and nimbler software development processes (Abrahamsson et al. 2002). Nimbleness in the above sentence implies firms' quality of being agile. A closer definition to business and software area of agility is proposed by Brinklow (2006, p.3) as '...sensing and responding to change...' where such

capabilities depend on the skills and knowledge of agents about the enterprise and business environment. Hence, uncertainty in the market conditions necessitate better and cost-effective products, thus imposing companies to be more agile (Ribeiro & Fernandes, 2009) and this in return creating a vital issue in the software engineering circle to deliver faster, better and cheaper software solutions (Dybå & Dingsøy, 2008).

More definitions of agility by different authors in a narrow approach to the software development also are presented as follows- Agility can be defined as persistent behavior or capability of an entity that exhibits flexibility to accommodate any kind of change quickly and follows a short time span, in an economic way, with the help of applying simple and quality instruments supported by prior knowledge and experience that has been acquired from internal and external dynamic environments (Qumer & Henderson-Sellers, 2006). Erickson et al. (2005) also defined agility as a means to minimize the heaviness of development which is commonly associated with the traditional method so that it allows to promote quick respond to changing environments, thus accommodating change in user requirements, accelerating project deadlines and the like.

Similar to the organizational level, change also is inevitable and affects projects differently. Verma (1997) identified two types of major changes that can significantly affect projects: scope changes and baseline changes. Scope change refers to changing the scope of a project through making additions, modifications and deletions to the end product or service. Here the common sources of changes could be: project specification changes, design change, technological changes, business cycle change and personnel changes. On the other hand, Project baseline change is regarding change in the project specifications, required standards and targets with respect to budget, schedule...etc.

The aforementioned challenges come with different frequency and magnitude entailing different impact on projects. The classical project management as mentioned previously is a very comprehensive framework which is proven to work in diverse project types and situations. However, in highly uncertain and changing requirements, the validity of the method has come into question.

Chen (2004) outlined a criterion which enables organizations to decide on appropriate methodologies for their project. The first criterion is the project environment; it consists of operational environment, the product/process development and technology development environment. The second criterion is organizational stakeholders; this questions, if it is a single organization or multiple, or if it is a hybrid that involves both. A summary of the criterion with proposed management method is shown here in the matrix below.

Table 3 The Environment - Stakeholder matrix (Source: Chen 2004, p.20)

	Multiple External Stakeholders	Multiple Internal Stakeholders	Single Organization
Operational Projects	Classic	Classic	Classic
Product/Process Development Projects	Classic/Agile	Classic/Agile	Agile
Technology/Platform Development Projects	Classic/Agile	Agile	Agile

The above matrix indicate that for technology (platform) development projects where there is a single stakeholder (organization), agile method of project management to be appropriate. The center of this thesis is distributed software developments, which can be categorized as high technology development projects, thus relates to the agile method described on the above matrix. Agile software development represents a major departure from traditional, plan-based approaches to software engineering (Dybå & Dingsøy, 2008). Plan-driven development methods begins with initiating and documenting a “complete” set of requirements, tailed by specified high level design and architectures on how development and inspection to be done (Lindvall et al., 2002). Around 1990’s practitioners started to get frustrated with the project initiation documentation and the impossible to work with, project architecture and design steps, thus they started to use Agile method (Highsmith, 2002). In the subsequent part of the thesis definition, characteristics and challenges of agile methods are discussed in detail.

2.7.2. The Agile Methods

It was in late 1990’s the software development started to get increasing public attention due to the emergence of new methodologies namely, agile, and these methods uses a combinations of old ideas, new ideas and transmuted old ideas (AgileAlliance, 2000). The primary goal of Agile Methods is to permit organizational agility, but what does agile mean? The methodologies as described in the agile alliance, tended to highlight collaboration among developer and customers; face-to-face communication rather than through documents; fast delivery of deployable business value; small, self-organizing teams; and crafted in such a way that requirement changes are embraced easily. The Agile software development became a new paradigm to the development process and provided a means of organizing complex multi-participant development whilst embracing constant project change (Strode et al., 2012). Agile software development evolved from the personal experience and collective wisdom of the different consultants and software developers (Dingsøy et al., 2012).

2.7.3. The Agile Manifesto

The Agile Manifesto was emerged in February 2001 after a meeting by a group of independent practitioners and consultants of several programming methodologies (Beck et al., 2001; and Cockburn, 2002a). Even though the practitioners did not have

the same opinion in everything, they arrived at a consensus around four central values that agile software development must uphold (AgileAlliance, 2000). The manifesto is written as follows-

'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.' The manifesto characterized the values of agile methods and provided distinguishing dimensions from that of traditional methods (Cohen et al., 2004). An important aspect here is that the values on the left hand above are important but in agile methods more emphasis is given to those on the left side. The above table is briefly discussed here below:-

*Firstly, **Individuals and Interactions** over Processes and Tools*

Project success is mostly dependent on individuals, thus having a good process may not be a guarantee on its own unless there are strong players (Martin & Martin, 2006). The authors further argued that even having a group of strong players may fail unless there is a teamwork. Glass (2001) stated that traditional developers are caught-up by the process and most of practitioners themselves have recognize that people matter more than processes. Agility is about relationship and communality of developers and the human role in the success of projects as opposed to institutional processes (Abrahamsson et al., 2002). Martin and Martin (2006) also emphasized that '... building the team is more important than building the environment' (p. 29) and after the team is built, they can configure the desired environment on their own. Hence, the agile community is promoting that success is dependent on programmers who work well with others through communication and interaction than those who follows the process well but fail to interact.

*Secondly, **Working software** over Comprehensive Documentation*

It is literally impossible to develop a software without documentation (Martin & Martin, 2006), but the traditional development made the documentation fetish and its prime goal by overlooking the ultimate objective of the project, which is developing a software (Glass, 2001). Martin's first law of documentation states that 'Produce no document unless its need is immediate and significant' (Martin & Martin, 2006, p. 31). One of the vital objectives of the developing team is to continuously generate tested and working new software releases on a frequent base (hourly, weekly or monthly), and that urges the developers to keep the codes simple and technically advanced, thus

reducing the documentation load to an appropriate level (Abrahamsson et al., 2002). The agile method here stresses that teams should not be caught-up by the pursuit of over documentation instead rather focus on delivering working software.

Thirdly, Customer collaboration over Contract Negotiation

There is a deep-rooted wrong assumption of commoditizing software and simplifying the development process into conventional project set-up where customers enter into a contract with the developers by stating certain description and agreeing upon fixed price and schedule (Martin & Martin, 2006). Such attempts always have been a failure due to the fact that customer taste and preference change and terms of specification become meaningless long before the projects are completed. Although the importance of properly drafted contracts increase with the growth in the size of project, negotiation should be viewed as a means of achieving and maintain a good relationship with the customers to deliver sustainable value (Abrahamsson et al., 2002). In this respect agile method focuses on delivering business value as early as possible through short-cycles in collaboration with customer, thus reducing the risk of contract non-fulfillment; but undertaking significant collaboration also requires proper contracts (Glass, 2001; and Abrahamsson et al., 2002).

Fourthly, Responding to change over Following a Plan

Projects ability to respond to change is often considered to be the determinant factor for its success or failure, thus plans must be built in such a way they can be adjusted to accommodate changes (Martin & Martin, 2006). The main argument by Martin and Martin is ‘...software development cannot be planned very far into the future’ because (1) business environments constantly change causing software requirements to change, (2) After the system starts to function users usually require amendments and (3) even if we know the requirements are not going to

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

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

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

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

Source: <http://agilemanifesto.org>

change, it is difficult to estimate the time it takes to develop them. Glass (2001) analyzed the issue from two angles; On the one hand, at the outset of software projects, customers and users do not always know what they want, thus they must be allowed to change their requirements during project execution; but on the other hand, the main reason for software projects failure is requirement change. Therefore, Glass noted that, the extent to which change should be embraced in the software development depends on several aspects. Abrahamsson et al. (2002) also emphasized that responding to changes creates more value to customers and relevant in delivering quality software provided that (1) the development group, which consists of both the developer and customer representative to be well-informed, competent and authorized to make amendment, (2) the existing contract support and allow such adjustments.

2.7.4. The Agile principles

The Agile principles which are illustrated in the Agile manifesto (AgileAlliance, 2001), gives more weight on the motivated and empowered software developers – relying on technical their excellence and simple designs – It delivers business value through working software to users with a regular short period. The principles consist of many practices that are believed to create more value to customers. Basically these practices entail the idea of self-organizing teams where team members are not only co-located but also work in such a way that sustains their creativity and productivity. The principles also support practices that accommodate change in requirements during different stages of the development process. In addition, customers are actively participating in the development process through feedback and reflection mechanisms, thus leads to more satisfied customers. In fact, the principles do not define agility, but they provide guidelines for delivering high-quality software in an agile manner. Despite the fact that the principles are not totally unfamiliar to the software development, they were combined and put together in a new way as a ‘theoretical and practical framework’ to allow developers better deliver quality software for customers (Williams & Cockburn, 2003). According to Charette (2003, p.19) opinion ‘... agile development rests on five core principles: support the customer; keep things as simple as possible; embrace change; account for behavior; and build virtuous cycles’.

2.7.5. Definition of the Agile Methods

The newly emerging development method is different from that of the traditional one, as stated by Cockburn and Highsmith (2001, p.122) - ‘what is new about agile method is not the practice, but their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness and maneuverability. This yields a new combination of values and principles that define an agile world view’.

Some authors started to give a formal definition of agility, for example, Anderson (2004) in a simple way described agile software development as a more profitable process mode than traditional software development. Highsmith (2002 and 2004) defined agility as a means generating profit in a turbulent business environment as a result of balancing flexibility and stability via creating and responding to changes.

Agile developments are also described as a set of work methods and tool boxes with an objective of improving the ability to quickly respond to changes in market request; cutting development time and waste; and reducing developers' stress while increasing productivity (Softhouse education, 2012).

There is still no universal definition of agile software development, but the Agile Manifesto principles better frame the concept (Conboy & Fitzgerald, 2004). Conboy (2009) has gone through a lengthy process of formulating a definition and formative taxonomy for agility in an information systems development (ISD) context and finally articulated it in such a way:

'Final Definition of Agility- the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment' (p.340).

In fact, the very notion of agile can be fully understood from the Agile Manifesto and the principles; however, we find the definition by Cohn and Schwaber (2003) to be more comprehensive and understandable. Hence, the definition agile for this thesis purpose is as follows:- Agile is defined as rapid paced and incremental approach to software development; which is performed in a collaborative manner by highly autonomous self-organized teams with shared vision; producing high quality solutions in a cost and time efficiently manner with less emphasis on detailed planning and documentation (Cohn & Schwaber, 2003)

2.7.6. Characteristics of the Agile Method

Agile methodologies in generally can be boiled down into the following four characteristics (Abrahamsson et al, 2003; and Meso & Jain, 2006):

- Incremental- implying that there is a small software releases to allow early and continuous delivery of working software with rapid development cycles that may range from a couple of weeks to months, thus providing customer satisfaction.
- Cooperative – the method also allows customers to work with the developer on a daily basis and the interaction enhances the quality of software development. It also encourages team members to build trust by providing the desired environment and support.
- Straightforward - Its design and architectures emanate from the self-organized teams and allows members to easily learn and modify through team reflection sessions, continuous attention to technical excellence and principles of simplicity (maximizing the amount of work not done). It also promotes efficient and effective communication mechanisms of software development such as face-to-face conversation and downplays the significance of formal processes by only relying on sufficient documentation.

- Adaptive – The unique nature of the method is also related to its welcoming feature and ability to late changes in requirements from users, thus offering a competitive advantage for customers. This ultimately ensures the delivery of working software which is assumed to be a measure of project performance (ibid).

After the agile manifesto was articulated, different authors and practitioners have been developing and elaborating the method to help stakeholders understand its different dimensions. Advocators of agile methods claim that, at its core, agility entails the ability to respond to changes in a flexible, simple and speedy manner (Cockburn & Highsmith, 2001). Having minimal formal process and light weight documentation are the other aspects of agility that include the concept of leanness with the essence of ‘light’ methodology that promote speedy response and easily maneuverability (Erickson et al. 2005; and Cockburn, 2007).

While the different agile techniques vary in practices and emphasis, some of the characteristics that have been assumed to be shared practices by all include- iterative development, focus on interaction, face to face communication, and the reduction of less relevant intermediate artifacts (Cohen et al, 2004). According to Cohen et al. developing in iterations enables the development team to embrace the quickly changing requirements and a co-located working set-up facilitate the communication and fast decision making; besides, reducing intermediate artifacts that do not add any value to the final deliverable will help the resource and effort shift to the development of the software, thus shortening the development period.

As mentioned before, agile software development is a common name for a set of frameworks. The range of methods (Abrahamsson et al. 2002; Boehm & Turner, 2004) under the umbrella of agile software development are: Dynamic Systems Development Method (DSDM, <http://www.dsdm.org>), extreme Programming (XP, Beck, 1999a and 1999b), Crystal Methods (Cockburn, 2001), Feature-Driven Development (FDD, Palmer & Felsing, 2002), Scrum (Schwaber & Beedle, 2002), Adaptive Software Development (ASD, Highsmith, 2002), and Lean Development (Poppendieck, 2001; and Charette, 2003). Although the methods differ in focus and presentation their basic assumption steam from the values and principles of the Agile Manifesto discussed above (Hanssen & Faegri, 2007).

2.7.7. Overview of the Popular Agile Methods

Discussing agile methods on a general level may not be enough to fully explicate and understand the underlying concepts, because, agile, as mentioned earlier, is an umbrella concept that consists of different methods. Hence, further discussion of specific methods is essential and relevant to observe and scrutinize fundamental issues of the method, but to address all types of methods is also beyond the scope this thesis. Therefore, this thesis considered only two types of agile methods. The most popular and widely used agile methods are Scrum and XP (Sutherland & Shwaber, 2011) and their creators were the signatories of the Agile Manifesto. In the subsequent part we

presented both methods in a systematic manner that helps to structure the theoretical framework based on the following dimensions: Process, roles and responsibilities, practices and scope of use.

❖ *Extreme Programming*

Extreme Programming (XP) is the hottest agile method (Cohen et al., 2004) which was originally emerged to improve the long development cycles of traditional methods (Beck 1999a). According to Hayes and Andrews (2001) XP advocates the ‘... values of community, simplicity, feedback and courage’ (p. 10). XP as discussed by Beck (1999b) is a team based software development method that better respond to uncertain and changing customer requirements. XP claims to have distinctive features than other methodologies in terms of - early feedback from short release cycles, ability to have flexible implementation schedule, incremental planning approach, evolutionary design approach that lasts with the system and reliance on partnership of programmers with ordinary skills.

According to Beck (1999b) some of the fundamentals to XP are: producing software in pair; writing unit test before programming, implying that programming is driven by test; start projects with simple design that evolve to incorporate the needed flexibility; putting minimum system into production and grow it to valuable direction; following the development a number of integration and testing within a day to ensure quality. In order to integrate individual and company interest, consistent values must be embraced by XP. The four XP values are communication, simplicity, Feedback and courage.

XP Processes

The XP life cycle includes exploration, Planning, Iteration to Releases, Productionizing, Maintenance and Death (Beck, 1999b). The *Exploration phase*, according to Beck, includes development of the architectural spike and customer’s user stories. The story card captures the features which are expected to be added in the program. Teams are also familiarized themselves with the practices, tools and technology. This phase may take from few weeks to few months depending on team familiarity with the technology.

The *Planning phase*, is regarding prioritizing stories and reaching an agreement with customers on when the smallest, most valuable release to be made. The time for the first release does not normally exceed two months, but the planning process may take a couple of days.

The *Iterations to Release* phase encompasses the major development efforts such as modeling, programming, testing and integration. The schedule in the planning is broken down into small iterations with a time period of one to four weeks each to implement. An important aspect here is the customer decides each story before iteration and tests are run at the end of every iteration; thus allowing the system to be ready for production at the end of the last iteration.

The *Productionizing phase* is about producing a release that can be used by customer. This phase needs additional testing (acceptance) and checking of release performance before they are handed over to customers. There is still a room for amendments in the releases and suggestions are documented for later phase (maintenance phase). The iteration during this phase is much shorter averaging form one to three week.

The *Maintenance phase* is a normal state of XP projects that involves the customer support task; by keeping the first release running in the production while producing new iteration. In this phase the development may slow down and additional people may be required to be incorporated into the team, thus changing the structure of the team.

Finally, the *Death Phase*; at this stage customers nearly do not have any stories to be implemented. Reliability and performance of the system is assessed in terms of customer satisfaction and finalizing the development process by properly documenting the system so that no change is made on architecture, design or code (ibid).

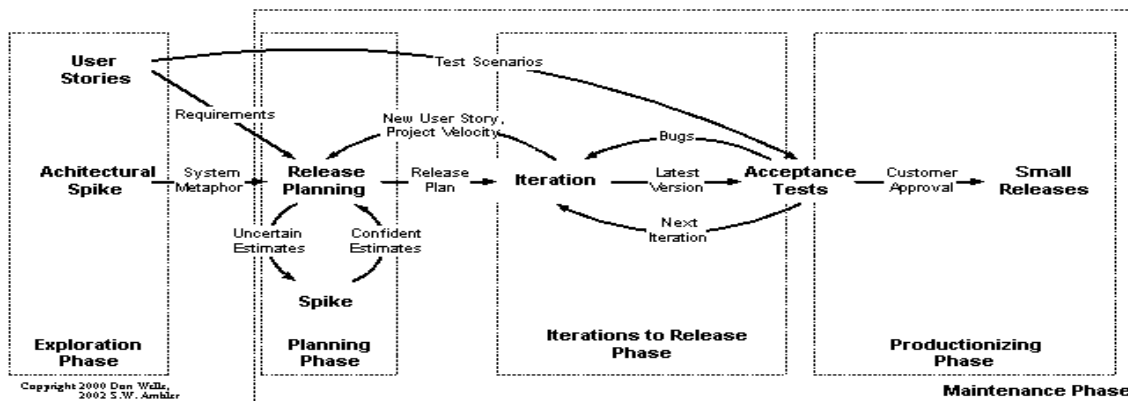


Figure 5 The XP project lifecycle (Source: <http://www.agilemodeling.com>)

XP Roles and Responsibilities

Beck (1999b) identified different roles and responsibilities in XP during the development process. Here are briefly discussed the roles along with their responsibilities: Customer, Programmer, Tester, Tracker, Coach, Consultant and Manager.

XP Practices

According to Beck (1999a) XP involves a variety of ideas and practices which are drawn from already existing methodologies. The aim of XP as outlined by Beck is to enable successful software development regardless of the constantly changing requirements by creating small to medium sized team, pair programming, involving customers, short iteration, small release, rapid feedback, better communication and coordination, light documentation continuous integration and testing with collective code ownership. Hence the following practices are provided to support the above mentioned objective and characteristics.

- Planning game- XP requires close interaction among customers and programmers. Here, quickly determining the scope of the next release by consulting customer's priorities and technical estimates is critical
- Small Releases- refers to putting a simple system into production quickly (at least every 2 to 3 months).
- Metaphor – which is defined by the programmer and the customer provides guidance for the whole development with a simple shared story of how the system works.
- Simple Design - extra complexity are removed immediately as they are discovered.
- Testing – XP is a test driven development. The unit tests are implemented before the code and run flawlessly during the development. Customers also write tests on functional features.
- Refactoring – the restricting of system without changing its behavior by removing duplication, improving communication, simplifying and adding flexibility.
- Pair Programming - the production code is written with programmers at one computer.
- Collective ownership – Anyone can change any code at anywhere in the system at any time.
- Continuous integration – New codes are integrated into the code-base whenever it is completed and ready. Thus, integration and building the system is done many times a day. Tests are also run for the change in the code to be accepted.
- 40-hour week – Maximum working hours per week is 40 and overtime for two weeks in a row is considered as a problem
- On-site Customer – the customer is live user on the team and expected to be present and available full-time to work in the project.
- Coding standards - programmers should follow the coding rules that emphasize communication through the code.
- Open workspace –preferred if the room is large with a small cubicles and Pair Programmers

Scope of XP use

The XP methodology is not suitable for all types of development projects (Beck, 1999b). The fact that, XP is aimed for small and medium team size that holds from 3 to 20 project members; and physical proximity requirement to facilitate communication and coordination among members (Beck, 1999b); has an implication on its less tolerance nature for big projects and scattered programming. For XP projects geographically distributed teams are 'two teams working on related projects with limited interaction' (Beck 1999b, p. 158). Another aspect that affects the development project is business culture (Abrahamsson et al. 2001). XP has its own culture, practice, roles, processes and others; any resistance against these practices and principles from the management, customer or project members may lead to failure (Beck, 1999b). Beck also added that technology that does not support change or demands long

feedback time may not be suitable for XP processes. XP was also found to be hard to introduce it in a complex organization, but apparently easy in other kinds of organizations (Dybå & Dingsøy, 2008).

❖ *Scrum*

Schwaber and Beedle (2002) indicated that, the term ‘scrum’ was first mentioned in an article of Takeuchi and Nonaka (1986) referring to an adaptive, quick, self-organizing product development processes. The term ‘agile’ originally was derived from the game of rugby denoting ‘getting an out-of play ball back into the game’ through teamwork (Schwaber & Beedle, 2002). Sutherland and Schwaber (2011) claimed that ‘By far the most popular agile method is Scrum’ (p. 15). The authors asserted their claim by stating that, in 2011 out of those who implemented agile methods worldwide, scrum takes 75% of the share.

In the ‘Scrum guide’ by Sutherland and Schwaber (Scrum, 2011), Scrum is defined as ‘... a framework for developing and sustaining complex products’. Scrum is developed to manage the systems development process (Abrahamsson et al., 2001), and it is grounded in empirical approaches using the ideas of industrial process control theory to systems development (Schwaber & Beedle, 2002). Scrum employs an iterative and incremental development technique with its set of value and practices based on the thinking of environment and technical variables (eg. time frame, resource, requirement, technology) likely change during the development processes, thus the incremental approach optimize predictability and control risk (Abrahamsson et al., 2001; Schwaber and Beedle, 2002; and Sutherland & Schwaber 2011).

Scrum is designed in such a manner that people can address complex adaptive system and it consists of three roles which are considered to be as its foundation: Developers, Scrum Master and Product Owner (Softhouse education, 2012). Scrum is founded on the philosophy that asserts knowledge comes from experience and decision making on known facts. Hence, the entire framework relies upon three pillars known as transparency, inspection and adaptation (scrum, 2011).

Scrum Process

Scrum processes have three groups of phases known as Pre-game, Game and Post-game (Schwaber 1995; and Schwaber & Beedle 2002):-

- a. Pregame- consists of two sub-phases: Planning and Architecture.
 - Planning stage is regarding the definition of the system being developed and Product Backlog is created based on all currently known requirements. The requirements are prioritized along with the estimate of cost and time and the stage also include definition of project team, tools, required technology and training...etc.
 - The architecture refers to the high level design of the system which deals with how the backlog items will be implemented. It is planned

based on current items on the Product Backlog and when the need for a change or enhancement occurs design review will be held to assess problems with implementing the new system.

- b. Game- is mainly about the development phase and is described by Schwaber and Beedle (2002) as the ‘black box’ to imply its unpredictable nature. Environmental and technical variables such as timeframe, quality, resources, requirement, development methods and others involved in scrum tend to change during the ‘Sprint’ (discussed in the sections below) of the development phase. Hence, scrum finds a way on a constant base to control the development process and flexibly adapt to changes. Schwaber and Beedle (2002) outlined a working model known as ‘Sprints’ which delivers the desired goal in thirty days. Scrum develops software in Sprint that allows the development of new release functionality in an iterative cycle. Before the system is ready for distribution, multiple iterative development sprints or cycles are implemented to make-up the whole system. As mentioned above, sprints on average take from one week to one month.
- c. Post-game – is the closure of system release. At this stage the agreement is assumed to be met and environmental variables such as requirements are completed. The system is ready for release and all tasks such as integration, system testing (pre-release testing) and documentation and other issues are closed.

Scrum Roles and Responsibilities

Scrum is founded on three main roles that have different tasks and responsibilities during the process of the software development (Softhouse education, 2012). These roles namely are Scrum Team, Scrum Master and Product Owner.

- Scrum Team - is the project team member that performs the actual work of problem solvers and designers. The team may consist of 5 to 9 people, a group that researches and experience has shown to be best for development work (Softhouse education, 2012). In order to achieve the goal of each sprint; the team has the authority to organize themselves and take necessary decisions that suits the arrangement and assignment of tasks, thus members do not have specific project role and allows everyone to swap tasks with another member.
- Scrum Master – ‘... is a combination of coach, fixer and gate keeper’ (softhouse education, 2012, p. 7). It is the Scrum Master who ensures that the development is carried out according to Scrum rules, vales and practices and progressing according to schedule (Schwaber & Beedle, 2002). The Scrum master is the

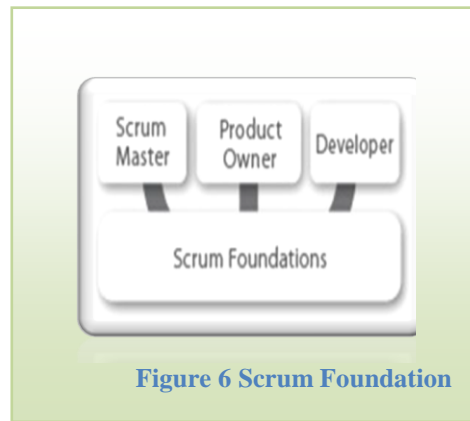


Figure 6 Scrum Foundation

mediator between the customers and management; He/she ensures that the developers are not disturbed and facilitate the best possible circumstances to realize Sprint goals by removing the impediments.

- Product Owner - is a representative of the customer and responsible for the project, managing and controlling and making the Product Backlog list. He/she makes a final decision on priorities of Product Backlog and estimating development effort for Backlog effort (Scwaber & Beedle, 2002).

Additional roles illustrated by Scwaber and Beedle (2002) are customers and management : Customers, participation in the development process by setting and prioritizing Product Backlog tasks; Management, involving in decision making and setting standards and conventions for projects to follow and following up progress of projects.

Scrum Practices

In fact scrum does not have or provide a specific development practice, rather employ certain management tools and practices that help avoid the unpredictability and complexity of the process (Schwaber, 1995). Here below are the tools and practices used in Scrum as discussed by Schwaber and Beedle (2002).

- Product Backlog - defines what is required in the final product based on current knowledge. It consists of prioritized and constantly updated information on business and technical requirement for the product being built or improved. The product owner is responsible for creating, controlling, updating and prioritizing the Product Backlog.
- Effort Estimation – is an iterative process by which the Product Owner jointly with the Scrum team performing the effort estimation.
- Sprint – Scrum is based on Sprint, which can be considered as a focused effort towards a fixed goal with in a 30-day period (Softhouse education, 2012). It is also labelled as ‘... a set of development activities conducted over a pre-defined period...’ (Scrum, 2011, p. 14). Schwaber and Beedle (2002) also defined sprint as a procedure of embracing changes in the environmental variables (e.g. requirement, time or technology) and it involves Scrum Team by organizing themselves to produce new executable product increment. Out of the Sprint’s 30 calendar days, the first tasks are set aside to create a Sprint Backlog. The Sprint Backlog is a set of Product Backlog items selected for sprinting; it serves as a forecast about what functionality of the product will be in the next increment and the required work to deliver that functionality (Scrum, 2011). Each Sprint adds new functions and improves the product market value (Softhouse education, 2012).

To make the development effort more productive, Scrum lays down four sets of formal events for inspection and adaptation: Sprint Planning Meeting, Daily Scrum, Sprint Review Meeting and Sprint Retrospective (Scrum, 2011). Refer to Scrum guide line, Figure 7 below.

Scrum guidelines

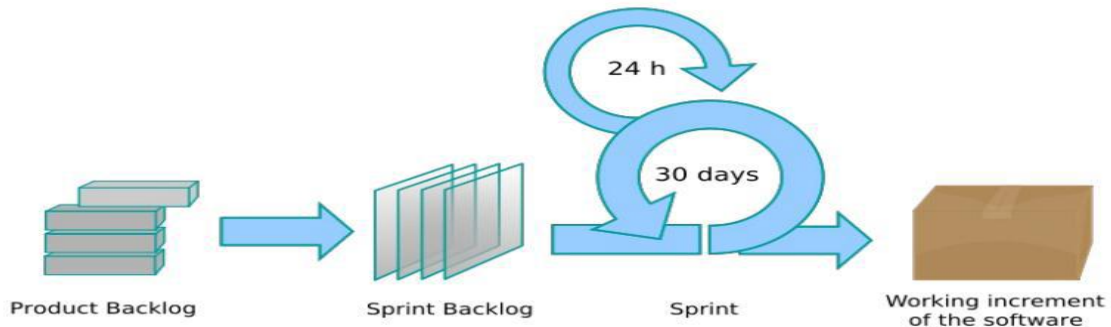


Figure 7 Scrum guidelines (Source: <http://softhouseeducation.com>)

- **Sprint Planning Meeting:** consists of two phased meetings arranged by the Scrum Master (Sutherland and Schwaber, 2011 (1) involving users, customers, management, Product Owner and Scrum team mainly dealing with ‘what will be done in this sprint?’ In other words, the first phase meeting is to decide the next Sprint’s goals and functionality. The second phase of the meeting is held among the Scrum Master and Scrum team focusing on ‘how do work in order to deliver the spring goal?’ The meeting is time-boxed eight hours every month (30 day Sprint) and believed to create a collaborative work among the whole Scrum team (Scrum, 2011).
- **Daily Scrum** – is a 15minute meeting to assess progress toward sprint goal (Sutherland and Schwaber, 2011). It is conducted at the same place and time to minimize confusion and during the meeting three questions are raised by the Scrum Master and explained by each Scrum team (Sutherland and Schwaber, 2011; and Scrum, 2011).; the questions are :-
 - What has been accomplished since the last meeting?
 - What will be done before the next meeting?
 - What obstacles are in the way?
- **Sprint Review** – At the end of every sprint a time box of 4 hours is spent on meeting held by Scrum Master along with the Scrum team, product owner and stakeholders to present the result of the Sprint (Sutherland and Schwaber, 2011; and Scrum, 2011). The meeting includes demonstration of completed functionality, identifying opportunities, constraints and finding presentation and things to improve.

- Sprint Retrospective meeting- is a meeting about 3 hours held after completing each Sprint among the Scrum Master and the Scrum Team to review ‘what went well and what should be improved in the next Sprint?’ (Softhouse education, 2012, p. 12)
- Burndown Chart- is ideally a downward sloping graph which is on a trajectory to arrive at “zero effort remaining” at the last day of the Sprint (Softhouse education, 2012). It is a better means of measuring project progress and it has two forms: ‘Sprint burndown’ (refer to Figure 8- shown below) and ‘Release burndown’ charts (Scrum, 2011).

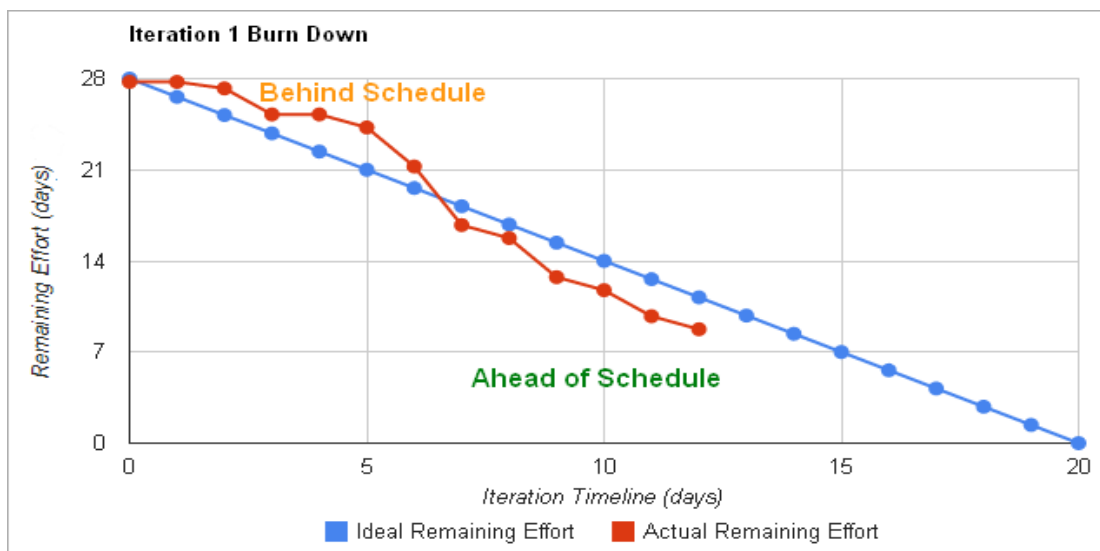


Figure 8 Sprint Burndown Chart (Source – www.joel.inpointform.net)

Scrum Scope of Use

According to Abrahamsson et al. (2001) Scrum is suitable for small sized team that consists of less than 10 engineers. Schwaber and Beedle (2002) also suggested that team should comprise of 5 to 9 members and as size gets more, multiple teams should be formed.

2.8. Challenges of Agile Software Development

Even though agile methods seem to be a better way of dealing with software development projects, they are not success guarantee. Agile methods alike the classical project management have challenges. Boehm (2002) argue that agile methods focus more on the process than the deliverable and this gives a room for the developers to engage in unplanned and undisciplined programming. Nerur et al. (2005) in their research, Challenges of Migrating to Agile Methodologies, also added the challenge of changing the role of leadership from command and control to support and collaborate. Another challenge is Knowledge; in agile method much of the knowledge is tacit and resides in the heads of team members, hence, there may be a shift in balance of power from management to teams. For example one of the problems with XP is its dependence on source code for documentation, as the result of that in-depth knowledge

of software product exists in the heads of the developers, thus loss of these developers leading to loss of organizational memory (Turk et al., 2002). In addition the team members may face difficulty in decision making, which is referred as pluralist decision making, thus leading to conflict due to the accountability issue as a group. Boehm and Turner (2004) along with Nerur et al. (2005) underscored the importance of suitable organizational culture to sustain agile methods. The challenge here is; it takes a great deal of time to build a culture which is required for agile thinking. In addition, when a company consists of different cultures at the same time there exists a conflict; for example, a culture which is suitable for agile projects along with a culture convenient for line function (department) may not co-exist, thus demanding a change in organizational structure (Nerur et al., 2005).

According to a study conducted by Dybå and Dingsøy (2008) on 'empirical studies of agile software development up to and including 2005', there is not adequate evidence to conclude that, agile methods are efficiently applicable for large distributed projects. Based on earlier discussion on globally distributed software development; the difference in geographical, temporal and cultural aspects generate challenges related to communication, coordination and control. It is apparent that agile methods will also face similar problems as they start to work in a distributed project set-up.

The agile principles and values do not support the distributed development environment (Turk et al., 2002). According to Turk et al., when the developers are not located in close geographical proximity to one another and also are not closer to the customer, then issues related to the principles and values such as face-to-face communication, customer interaction, light documentation and other agile assumptions will be in question. Unfortunately for distributed development process documentation happened to be more important due to the limited options in which developers and customers have to interact, thus the agile assumption of on minimizing documentation becomes a challenge (Turk et al., 2002). Some additional challenges of agile software development which are identified by Turk et al. (2002) are briefly presented in table 4 in the next page:

Table 4 Limitations of agile method in outsourcing (Adopted from Turk et al., 2002)

<p>Limited support for subcontracting</p>	<p>Outsourcing of software development to external subcontractors based on specific requirements agreed on the contract. The challenge which is related with such set up is (1) is it possible to co-locate subcontractors with developers and customers? (2) If the subcontractors have a predefined agreement on the contract, is it possible to accommodate the notion of evolving requirement as stated in the changing requirement assumption agility?</p>
<p>Limited Support for Development Involving Large Teams</p>	<p>When there are large projects that require large teams; the agile principles of team communication, face-to-face interaction, and documentation may not hold. Because large teams require less agile approach as agile only focuses on managing small sized teams. With large teams, more documentation and informal communication and more coordination are inherently needed.</p>
<p>Limited Support for Building reusable Artefacts</p>	<p>In agile the sense of ‘big picture’ is missing; the agile team is simply focusing only on the current applications, thus designing reusable artefacts (codes, analysis and design documents, patterns, etc) that could be used for future projects do not seem to fit well for agile development.</p>

At this point, the relevant theory on Agile methods and Outsourcing software development has been reviewed and presented. In the next part the reader will be introduced with the research method which guides the entire study.

3. Research Methodology

In the following chapter on Research Method, a reader will be provided with description on what kind of research approach, design, data collection and analysis methods or techniques were selected for this study in order to answer research questions, make research results consistent and reliable.

“Social science research is an art as well as a science, and the skills and knowledge needed to be a researcher can only be acquired through experience over time. There are always judgments to be made and decisions to be taken about how best to go about research.” (Somekh & Lewin, 2005, p. 4)

3.1. Research Approach

The fact that Agile Method was originally developed by practitioners and limited research on the specific area puts a challenge to the methodological approach in terms of having a clear theoretical framework that guides the data collection and analysis of the research. Hence, this research mainly borrowed some theories from the well-researched organizational literatures and applied **an inductive theory**, which is described as drawing generalizable inferences out of observations (Bryman & Bell, 2007); this approach provides better insight into the complex nature of the development projects and helps to understanding every bits and pieces of the practical approach to continuously build a theory through iterative approach by waving back and forth between data and theory.

According to Bryman and Bell (2007) research strategies are broadly categorized into two: qualitative and quantitative. Qualitative strategy emphasizes the importance of investigating the natural setting of variables in which they exist (Smith, 1983). Some of the reasons **why qualitative strategy was chosen** for our research can be found in The Thesis Handbook of Oklahoma State University mentioned qualitative research features: theories and hypothesis are evolved from the collected data; data are more subjective and reflect the perception of individuals in their environment; human is the primary collection instrument; research is conducted under the natural condition-naturalistic conditions; results are allow to form ‘real’, ‘reach’ and ‘deep’ data (Thesis Handbook, 1997).

The variables in this thesis use narrative descriptions and explanation with less emphasis to numerical quantification. Hence, the research strategy for this paper is qualitative research; this approach is defined as a scientific research that consists of an investigation that seek answers to questions (Blumberg et al., 2011) using words than quantification particularly in collecting and analyzing data (Bryman & Bell, 2007). Unlike quantitative approach that seeks to confirm hypotheses about phenomena, we use qualitative approach since it seeks to explore phenomena (Blumberg et al., 2011). By using qualitative method we generate more information of the complex context of

software development from the experiences and opinions of participants. This approach allowed us more flexibility in terms of being spontaneous and adaptable to relevant aspects while interacting with the study participants.

In fact there are built-in limitations of qualitative strategy but we duly noted and dealt with them through different mechanisms. Some among the issues that are identified to be the drawback of qualitative researches are: subjectivity that leads to procedural problems; difficulty in replicating the study; bias by the researcher; limiting scope due to use of in-depth data gathering (Thesis Handbook, 1997; and Bryman & Bell, 2007). However, in the subsequent parts of the research method the different measures are taken to ease the challenge and improve reliability and validity of this thesis are presented.

3.2. Research Design

One of principles on which research of this paper based is to provide both theoretical (academic) findings and meaningful suggestions to the management, since business studies in the end should be about practical solutions. One of the ways how to implement this approach is to define management dilemma (Blumberg et al., 2005) alongside with academic debate. In our case the defined research questions in the Chapter 1 can be both seen as academically interesting issue as well as a managerial dilemma for practitioners in the area of Agile projects and Outsourcing practices.

We define our study as a type of **explanatory/predictive study**. This type is chosen over reporting and descriptive study, since it includes analyzing complex patterns and possible giving valuable suggestions. This type of study also fits well with qualitative research, rather than in case of reporting or descriptive approach we should include larger statistical data, quantitative method and the spirit of study would be large scale, but lower depth. Even though, both qualitative and quantitative research can be applied to the same issue (Blumberg et al., 2005), qualitative research in this case fits better with researchers approach and preferences.

Next, the **multiple case study approach** was chosen. One the most popular definitions of case study is the one provided by Yin (1989, see below).

“Yin defines case study as ‘an empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used’. (cf. Yin, 1989)” (Blumberg et al., 2005, p. 190)

Case study was chosen among four other approaches: narrative, phenomenology, ethnography and grounded theory. Each of approaches has its strong and weak points; however case study, despite it's drawback to inability to generalize statistically, was preferred due to possibility to focus on several firms, an option to compare cases, conduct research “in a sequential, step-by-step, manner” and use cases since “they are especially appropriate for answering ‘Why?’ and ‘How?’ questions.” (Blumberg et al.,

2005, pp. 190-191). While narrative and ethnography approaches seem to fit better with other social sciences, there are not enough characteristics in our research questions for phenomenology or grounded theory approach. The fact that different types of Agile set-ups are used and outsourcing is involved to different extent do not form a strong phenomenon to explore; neither there is a big area of activity which was unstudied previously in order to use grounded theory.

Authors of this paper have chosen the **positivism research philosophy** over the interpretivism in general. That means that research will assume that data and transcripts of interview are objective information and the researcher takes an independent role (Blumberg et al., 2005). Moreover, the researchers will take a **role of observers** rather than interpreters of why particular data was obtained. The positivistic approach is favorable for this kind of study, since it will let researchers to focus on several explanations of researched agenda, while other aspects can be neglected and left for further studies. On the other hand, since the research is tightly connected with a case firm, we have to admit that some biased reflections from researchers side might arise. Even more, researcher's interpretations might be socially constructed. In the end, that means that keeping in mind clearer positivism philosophy, in practice the research is most likely to **end up in realism philosophy**, since the social effects and imperfections are likely to add some of interpretivism.

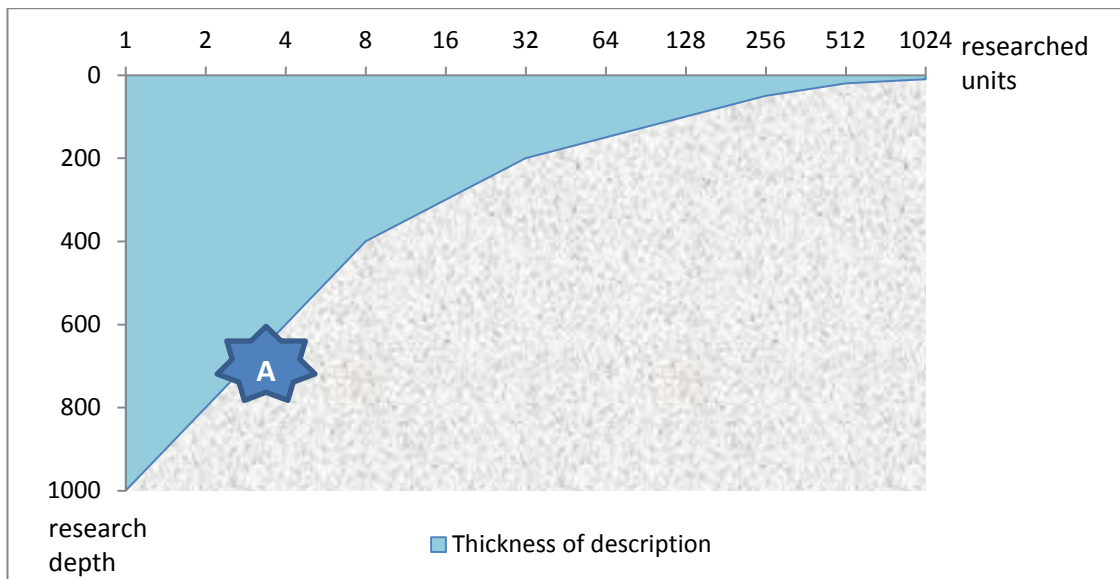


Figure 9 Positivistic-interpretivistic research philosophy scale (Adopted from Blumberg et al., 2005)

In the above Figure, you can see the adopted concept of how to visualize what thickness of description or in other word depth of study should be conducted depending on the method and philosophy chosen. The two extremes would suggest that very positivistic (probably quantitative) research should include large sample (lets say 1024 units) but in such case deal with small proportion of general issue. Other extreme – is in

depth case study – focusing on single sample and covering all possible factors involved (depth) would represent qualitative and possibly interpretivistic research.

In our research we would assume that the optimum place on the curve would be place “A”, since the research will be based around several firms, meaning that the depth of study will be deep, but not as deep as in case of single sample. In practice, to be as close as possible to the desired place “A”, three different firms were chosen. In order to deepen the depth of study additional observations and analysis of secondary data were involved – it also allowed **using triangulation technique** and improving validity of research (Bryman & Bell, 2007).

However such way of thinking, with all respect to good visualization, is limited since we believe that both positivistic approach research can be very thick as well as interpretivistic research can concentrate on single aspect. Despite that, we find this method of visual research placement somewhat useful and hopefully helpful to the reader in understanding research design.

It is important to stress that this research is a **cross-sectional study**, meaning it will analyze and deal with the issues present at the moment of study, at particular point of time (Bryman & Bell, 2007). Authors admit, that it would be a great value to have possibility to conduct longitudinal study, however the time and resources are not sufficient for a longer time horizon study under the requirements of Master thesis research. The benefit of cross-sectional study is that even though it might be close to other studies performed previously, the fact the time of study and cases are different brings new light in research.



Figure 10 Research question sequence (Adopted from Blumberg, Cooper, & Schindler, 2005)

Another Figure can be used to visualize the sequence of different question formulation during the research. As it is possible to see above, first the general research dilemma was formulated, followed by research question and investigative question. Differently from original hierarchy model, the management question comes later, after the first insights in the real industry situations, research and investigative questions were reformulated. Finally in the empirical part measurement questions are formulated and the research is closed by decision, or in other words findings and conclusion.

Moreover, the research is designed **to use two research questions** in the research purpose and analysis parts. Using research questions rather than hypothesis in our case (qualitative research) is more suitable way, since research questions can address several factors and explain complex issues (Bryman & Bell, 2007).

3.3. Data Collection

Eisenhardt (1989) stated that case studies are well-suited when the research areas for which the existing theory seems inadequate. In the early parts of this thesis we have explained that there is a big theoretical gap in the agile software development of an outsourced set-up, thus a case study seem to highly compliment the theory building effort of the field of interest in its natural setting. The thesis is based on a multiple-case study of software developing companies, where two of the companies **are located in Sweden (Linköping) – Zenterio and Ericsson; the third one in USA and Russia (Sankt-Petersburg facility in particular) – GGA Software Services**. Detailed descriptions of each case can be found in following Empirical data section. The reason for that is to produce sound and complete case description, including presenting each participant and using their experiences in following Analysis section. Ghauri and Grønhaug (2005) argued that it is appropriate to use multiple case study design when the study does not involve rare, critical and revelatory cases. The objective of this design is to investigate the research questions in different projects within different companies that seem to be engaged in a similar set-up and understand the same thing form different environments: Big and small companies from inside Sweden and from abroad; rather than just focusing on a single company and single project.

The main **unit of analysis** for this study was **agile projects**, which can be defined as the activities of a single project that uses scrum or XP involving an outsourcing. The thesis gives less emphasis to generalize to the entire software development but explore the specific conditions by which outsourced agile developments of the companies under study understood and handled. We will use one of the **non-probabilistic sampling** methods which is called **purposive sampling**. Purposive sampling refers to an approach to the sampling problem with a specific plan in mind (Blumberg et al., 2011). This type of sampling method allows a series of strategic choices while considering the subject of study mainly on where and how we have to conduct our research, thus amplifying the samples to be more attached to our objective of understanding the practice and experience of agile projects in the outsourced setup. The above mentioned companies are partly and or fully engaged in outsourcing their software development and purposive sampling enabled us to better address the intention of the thesis.

In order to investigate the business units, agile projects, we used **primary data collected through in-depth interview and observation**. Interviews were conducted through the **experience survey**, assuming that participants perceptions do not influence gathered information (in line with positivistic approach). Primary data will allow researchers to generate data which is specifically touches upon research area and questions, unlike secondary data would rather fit to some extent, since it was originally gather with different purpose. Moreover, **semi-structured interviews**, differently from closed questioners, allow authors to ask prepared question, follow the line of interviewee arguments, and if necessary or unclear ask additional pop-up questions, which in the end are valuable for context analysis. According to Blumberg et al. (2011)

semi-structured interview is fairly open framework to find out about peoples view on predetermined topics; it allows for focused, informal and both-way communication.

We had **five interviews** that lasted from 20 minute up to 1 and half hour with project managers, scrum master and software developers from the three companies (see Appendix 2 for Interview guidelines) and also made a **direct observation** of one project team Scrum meeting (see Appendix 1). The instruments we used to collect data were fundamentally critical and convenient for the iterative research design and in providing relevant information; hence, one interview and or observation was utilized as a means of modifying, adjusting and fine-tuning the next interview or observation and the direction of whole thesis. We were able to establish the validity of our instruments as we move along the iteration and reliability of instruments were also handled by once again going back to the interviewee for further clarity with few main questions. In addition the inclusion of different respondent such as project managers, scrum masters and project team members enabled study to be more valid and reliable.

Direct observation as defined by Bryman and Bell (2007) is a means of collecting data regarding activities, behaviors and physical aspects by ‘seeing’ and ‘listening’ without depending upon the respondents’ ability and willingness to provide information. The observation method included non-verbal behavior, which includes body movements, group behavior, closeness to each other etc. Also some of extra linguistic behaviors were observed, including vocal loudness and timbre, temporal rate and duration of speaking. Finally, spatial relationship observation is important, that includes how people organize their space, how closer or far they put themselves from others.

We, as a researchers choose **third observer-participant method**, which is simply being present to the meeting and observing as much details as possible (Blumberg et al., 2005). Since our task is to get as closer impressions on what is real practice, the active participation or directions would be not welcomed. Our role in the stand-up meetings was as a non-participatory external observer, but our presences didn’t much affect the behavior of the participant, because Scrum meetings are naturally informal and interactive. However, there were some difficulties related with the language of the meetings and in some cases there were technical terms that we needed to ask after the session is completed. Physical traces observation is also likely to be performed based on the gathered pictures that includes – how much notes could be places/removed from the Scrum board, what amount of written data, memos participants bring to the meeting, other traces that researchers could identify.

In the whole data collection method the critical challenge that we have experienced was the difficulties in setting proper time for interviews and getting consent and time slot for the observation; we also found transcribing interviews as a time consuming process. However, the instruments were complementing each other while confirming some aspects from different dimensions throughout the data collection process; thus elucidating the entire process of building our research questions and shaping the relevant theoretical framework.

3.4. Data Analysis and Presentation

Following Creswell (2007) recommendations on analysis strategies, researchers plan to perform analysis in three stages: first – organize the data (preparing transcripts, picking-up useful secondary source, aligning notes and pictures of performed observations); second – reduce the gathered data into themes by using coding, color highlighting and applied theories; third– the data will be represented as discussion and visually presented to reader by providing comprehensive table or figure.

The study mainly used two types of coding- ‘**Word Cloud**’ and ‘**Open Coding**’. We firstly tried to make use of the recently emerged data coding mechanism called ‘word cloud’; a word cloud (tag cloud) is a means of visual representation for text data that particularly use to depict keyword on websites or visualize text through representing the importance of each words with font size or color (Halvey & Keane, 2007). The coding technique was useful to quickly perceive the most prominent terms from the data collected through interview. Secondly we borrowed the traditional coding mechanism, which is popular in developing a grounded theory, and referred as Open Coding. According to Creswell (1998) *open coding* refers to categorizing the data and investigating the features and dimensions to define actions within the data. We chose *open coding* because it enables to build a multi-dimensional initial framework that can be used for later analysis and, as it constructs directly from raw data *open coding* establish validity of the research (Strauss & Corbin, 1998).

Moreover, we decided to pick-up most reasonable ways of above mentioned analytic strategies from Creswell (2007, p. 149) summary of some other author’s suggestions on implementing strategies: highlight certain information in description, make metaphors, write codes and memos, note patterns and themes, identify patterned regularities, contextualize in framework from literature, create a graph or picture of the framework, make contrasts and comparisons, display findings, compare cases. We find the mentioned techniques useful for our study in Agile software development area. Since the research will include interviews, observations and secondary data analysis, identifying themes and displaying the findings visually seems very reasonable and reader-involving way of analyzing the case.

As authors of this thesis work, we have to admit that both research structure and strategy of analysis accordingly, have changed and evolved from the initial start of the work early in the semester. As an example, it could be mentioned that first we intended to study challenges associated with outsourcing parts of Agile projects. However after deeper analyzing literature and conducting first interview we decided to research in a narrower aspect of contradictions between principles of Agile development and Outsourcing. Furthermore, as more empirical data was gathered and coded, the second research question on pre-conditions for successful outsourcing in Agile set-up became apparent. We see it positively as an evolution of research topic, finding particular context and maturing as a researchers. According to Dey c/f (Creswell, 2007, p. 150)

qualitative researchers often “learn by doing” and the qualitative data analysis rests on three “I’s – “insight, intuition, and impression”.

It is obvious that “qualitative research can produce vast amounts of data” (Pope et al., 2000, p. 114); thus, it is time consuming and cumbersome. In this study, we used different **data display techniques** to reflect the truth of phenomenon. Data displaying as referred by Miles and Huberman (1994) is a means of organizing, compressing the assembly of information in a manner that permits conclusion about the findings. Therefore, the study used firstly ‘word clouds’ to demonstrate key words out of the data and construct themes and groups for further analysis. Models, tables, figures and direct quotations were applied to support the narrative description of the study; because use of such techniques provides advantageous to illustrate meta-data, creates relational data and permits cross-case accuracy of analyses (Miles & Huberman, 1994).

3.5. Research Ethics

In the research ethics area the **deontological approach** was chosen concerning interactions with respondents. Would it be interview, survey or observation, no deception techniques were used. Authors believe that the nature of this study, especially keeping in mind positivistic approach, does not include such complexities that teleological approach should be put in place.

All participants of the research were provided with fair and **ethical treatment**, including: the aim and purpose of study was explained; the consent to data gathering and usage in research was obtained; interviewees were informed about interview being recorded and were given an option to read the transcript later; interview would exceed the agreed time only by interviewee initiative. In case of confidential information (legal names, customers, agreements and etc.) the information will be removed or the paper will be considered as confidential paper.

Finally, the paper is based on **the high standards of academic writing**, meaning that there is no place for plagiarism, data falsification, fake analysis or other unethical behavior from the authors. Linköping University as being home base institution for researchers and a reviewing-grading institution can be assured of high ethics standards of this research.

3.6. Validity, reliability and standards of quality

In order to ensure high quality research and valid findings it is important for researchers to take a look on themselves, to the participants and readers. (Creswell, 2007) In order to cover all three parties and produce high quality research some perspectives from Creswell (2007) were used in this master thesis: **triangulation** (using three independent source of information), **transferability** (thick description of data from participant to final reader), **authenticity** (if different viewpoints and voices were presented) and **integrity** (researcher’s self-criticism). In addition to that, two validation strategies were used during the research: first – **peer review strategy**, second – **identifying researcher’s bias**. In term of peer review authors to a great extend served

to one another as a peer reviewer, moreover master thesis supervisor and peer groups during seminars helped to improve research design and quality. Any researcher's bias were aimed to be clearly stated throughout the work, including past and research experiences, clarifying own positions, assumptions, propositions or other interpretations.

In term of research reliability, we can mention that all primary data was gathered with using high quality devices: during interviews - voice recorders and prepared questioners and during observation – photo camera and sketches with memos were used. Throughout coding process, each piece of data was analyzed, when coded with number or color in order to form/find particular themes or patterns. Finally, the guidelines of Linköping University for SMIO program master thesis writing were used in research and delivery of findings through this master thesis paper.

4. Empirical Data

In this chapter the raw empirical data of the research will be presented. The data will be aligned towards each particular case in order to form separate well-studied case of real company in software development industry. By presenting a well-grounded and proper depth ‘portfolio’, the authors seek to familiarize reader with firms, which were involved in the research; thoroughly describe and transmit the context of industries and firms, situation those firms are currently in, and what is also important an overlook of interviewed representatives. In addition to that, each case will include some piece of gathered secondary data and, if applicable, some of observational findings.

4.1. Case 1 - GGA Software Services LLC

The first empirical case in chronological order, which this research rests on, is the company called GGA Software Services (GGA). In this sub-chapter a reader will be introduced with company background, some facts from secondary data and one interviewee’s views on research issues.

4.1.1. GGA Company Background

According to Bloomberg Businessweek web company overview database GGA Software Services LLC provides software engineering, algorithm development, content development, and other consulting services in the United States and Russia (Bloomberg Businessweek web.) Founded in 1994, the company itself suggest that they are the world-class services provider in the above mentioned areas with headquarters based in Cambridge, MA, USA and a strong development facility in St. Petersburg, Russia (GGA Software Services web.).



Figure 11 GGA Software Services logo and slogan (source - official web page, see ref.)

Company has strong industry expertise of developing software, building databases, mining data, and generating digital content for their clients in four dynamic industries: publishing, pharmaceuticals/biotechnology, material sciences, and financial services. According to official information, these clients can get specific services in new technology development, software applications, database development, content services and IT consulting. Moreover, in 2009 GGA Software Services decided to share with industry and academia some of their own cheminformatics products through providing open-source applications.

We were able to interview other the internet a project manager whose name is Denis. He works in St. Petersburg facility and is in charge of pharmaceuticals related projects and serves as a business analyst for external clients in that industry. Therefore, the St. Petersburg facility and pharmaceutical area will be shortly presented below.

St. Petersburg Development Facility

Even though company's headquarters are located in Cambridge, Massachusetts, a strategy is to use facility in St. Petersburg, Russia, for speed and expertise. Citing company web-page we can see that this is being seen as a big advantage:

“By leveraging our St. Petersburg workforce, we can offer our clients a 24-hour “virtual workday,” as project work can continue in Russia after the business day ends in the United States. In this way, we can respond rapidly to customer requests, sometimes even delivering results by the next business morning.” (GGA Software Services web.).



The facility employs more than 200 software engineers and mathematicians, many of those hold Master or Ph.D. degrees in the related fields.

Figure 12 Pictures of St. Petersburg facility (source - official web page, see ref.)

Pharmaceuticals

“Discovering and bringing one new drug to the public typically costs a pharmaceutical or biotechnology company up to \$800 million and takes an average of 10 to 12 years.” – says the official company web page. No surprise, that firms which can improve or speed-up this process by bringing in new solutions are in demand. Our interviewee, Denis, added – “our specialization is pharmaceuticals and companies in that industry find us on their own.” Since authors of this research are not familiar with pharmaceuticals research, we asked Denis to comment on how software is being used in pharmaceutical firms. The answer was quite impressive and persuasive:

“As an example, this is a software which speeds-up drugs development. Drug development is an iterative task, not completely, but mainly iterative task. The correct software automates and speeds-up this process 10-s or 100-ds times. Therefore, the drugs are found out faster. These are also genetic research and its very much linked with software. Also systemizing the data from customer, organizing their database according to different chemical structures.” (Denis, GGA Software Services)

In addition to that, we also found from official web page, that for different clients, GGA has introduced more than 40 commercial software packages, including “advanced tools for statistical analysis, interactive molecular model development, data manipulation, and 3-D visualization of complex molecules. One of these packages received the prestigious Reader's Choice Award from Scientific Computing & Instrumentation magazine three times.” Even though, all big pharmaceutical firms have

their own software development units, Denis explained why they still work with specialized partners as GGA Software Services:

“It’s quality and all about quality and productivity. The cost is not that big saving [for a client], we are lucky, our firm has most expertise and lead in the market. It is easier for them [pharmaceutical firms] to address us, than work alone. What they did on their own was not that successful and of good quality, at some point internal team is not able to develop it further. Then they pass it here to us to make it work.” (Denis, GGA Software Services)

To sum up, we can see that there is a strong mutual interest and integration between key pharmaceutical companies and firms like our case company, which can provide some specialized expertise and accomplish projects.

4.1.2. Agile method in GGA

A big part of interview was dedicated towards understanding how a case firm is executing their projects, how the Agile method functions (we knew in advance that company is into Agile). First, the company’s web page gives us a clear structure of project methodology and values at AGG Software Services:

“GGA employs a rigorous project methodology, which is premised on a holistic view of systems engineering. This approach recognizes the vital importance of human interactions – between our management team and our clients, as well as between our clients and the systems we build. This philosophy informs every stage of our software development process – from gathering project requirements to delineating detailed specifications, executing the project, and testing the resulting systems. After all, meeting or exceeding client expectations is our ultimate goal.” (GGA Software Services web.).

This statement looked in line with what most often cited Agile method researchers say (Beck, 1999a; Cohen et al., 2004) regarding the key principles of Agile method and why it is different from traditional approach. However, we were surprise by the answer of our interviewee on the advantages of Agile method over classic waterfall model:

“Classical Agile, like XP or Scrum, in my opinion, up to 80 percent is a kidding/playing. Practices should be used selectively and they need to be adopted to customer... . In my case, different practices are used, the concentration is done on stand-ups, twice a day: morning and evening.” (Denis, GGA Software Services)

When asked on the future of Agile method – is it a trend or long term practice, Denis replied:

“If we take Agile as a full range of practices, it is a popular trend, also as Scrum, it will pass by. What will stay is the number of picked up good practices.” (Denis, GGA Software Services)

At this point we realized, that our perception that GGA Software Service was a pure Agile method using company was wrong. And practices are selected “selectively”. This is an important finding, and a reader should remember it, since it will be addressed later in Analysis part of the paper. As for GGA Software Services, we can see that they use Agile practices like stand-up meetings, but in overall they use it to some extend and selectively.

4.1.3. Projects, Teams and Activities in GGA

A big part of interview questions were related to the issues of how projects are being handled, what type of team compositions and ways of working are used. It would take too much space to present all the answers, so instead a compressed description will be presented.

The stand-up meetings are used daily and they last for 15 minutes, sometimes starting as late as 12-13pm. Currently Denis has 8 people in the team and thinks that a comfortable and efficient number of team members vary between 7 to 9. He had recent experience of running a team of 15 members and that was too hard. The interaction between customer and team actions is very quick. Denis has team conference with customers in the evening [we should keep in mind time zone difference between US and Russia] and is able to communicate new information to developers the next morning.

Teams are cross-functional, each person specialized in particular area. There are some trainings, lectures or internal seminars, but those happen ‘on a revolving time’ during a project, but not in-between projects. In case there is some high priority tasks, or some employees are sick and deadlines are close, Denis himself or other employees can participate in the project as a developer and accomplish tasks. As a good coordination practice, project manager identified the issue of co-locating a team: “very good practice, I don’t recall if Agile has it or not, it is to gather team in one room, it shortens communication distance, yes it’s a lot of noise, but is very funny and productive.”

Finally, when asked about should Agile method’s specification be strictly followed as designed and recommended by experts, Denis replied:

“I think, as an every model it is formal, but does not take into account the human factor. Absolutely not taking in account human factor. This model might work fine, but if you build and make a team to apply some particular model is not efficient, it is better to adopt, take the best practices and team will be more effective and interesting.” (Denis, GGA Software Services)

4.1.4. GGA’s Outsourcing/Distributed team in Agile Development

GGA offers a cost-effective way for client to utilize external resources for both product development, thereby improving the abilities of client’s technical expertise. Moreover, case firm has experience of dealing with technologically complex projects, hence that be financially attractive for their clients, rather than investing internally (official web-

page). As was already said, some project leaders and specialists are located in Cambridge, Massachusetts, while many software engineers and scientists are at the St. Petersburg facility, Russia. As company suggests, such set-up offers their clients “the best of both worlds”.

So, taking the outsider view, the company seems very much integrated in the global outsourcing processes. But then we tried to look deeper, and see how much the firm itself is engaged and willing to outsource their project work, we were surprised to find a different perspective. First, our interviewee Denis, pointed out the difficulties for software developers to interact on their own:

“Software developers they are all technical people, these kind of people have problems speaking the same language and to find some compromises, insist, co-ordinate something that it would work. They don’t know how to communicate.” (Denis, GGA Software Services)

Denis also was skeptical upon the practice of using freelancers, since it is hard to keep an eye on them. He prefers and from time to time hires a specialist, for example in chemistry. This person would perform a task in the facility, get paid and leave the project. These two seems to be a challenge on the personal level, but since the research is focus on the project level, we kept on asking what other issues are preventing company to go beyond sticking to work in the facility and outsourcing more. The answer now was about coordination and poor result as a consequence:

“They do not coordinated things; they are over deadlines and keep on hold each other, one team can work efficiently on time, other is not that efficient and it will take more time – naturally you have to find what job should be done by the first team in meanwhile. The problem is that, they perception can be different, they will not ask each other how you did it, not see how others worked, it will be more time spend, more bugs will come out.” (Denis, GGA Software Services)

To conclude, we see that in case of GGA Software Services outsourcing is surely there: a company provides such services; project manager is aware of possibilities and has positive personal experience in other companies. However, we could not see much of evidence of outsources software development, as a whole or as partial tasks within projects.

4.1.5. Benefits of Agile Development in GGA

According to our interviewee, Agile practices are not only trendy, but they have a number of particular benefits to the organization. Denis emphasized higher productivity and stand-up meetings as the most notable advantages in Agile set-up.

We were able to see one more hidden issue, which was not described in the literature, that is according to interviewee – the fact that Agile practices are used in developer’s

everyday life, the importance of age of software developer and the need to socialize at work:

“..Agile allows, different from Waterfall, to emphasize best practices and use it in life, in life of developer. Waterfall was giving big steps, what to do and when. Agile is oriented towards students and post-students. These guys will to work a lot, to go further. Mature people are not that good in Agile set-up, they don’t have that much drive. Young guys need a game, that is why for them Agile works. They want to communicate, they want to gather, socialize, to share same work then one develops the other double checks it. It should be Fun!” (Denis, GGA Software Services)

This is an important finding, since it can bring new perspective into research questions, and will be addressed in the analysis part. The first case has been covered and it is time to move to second case firm, which we were able to meet live and spend quite a time with different employees.

4.2. Case 2 - ZENTERIO AB

Second case in research empirical base, is company named Zenterio. The case on this firm will consist of primary and secondary data. Interviews were conducted with three employees – Kaj, Fredrik and Olle, who hold different positions in the company – project management office manager, project manager and software engineer respectively. Finally, the observation of Scrum meeting was performed in order to get more insights on Agile practices. It is important to mention, that the amount of time spent and data gathered at the Zentario is much greater than other two cases. Hence it can be analyzed deeper and findings are likely to be more valid towards this case.

4.2.1. Zenterio Company Background

Zenterio is one of the renowned Swedish software design company providing software solutions for service operators, system integrators and set-top box (STB) vendors. The core team of the company is composed of individuals with a greater knowledge and experience, due to having been in the digital TV industry since 1994. Zenterio was founded as a spin-out of Nokia Home Communications in the year 2002. Its head office is located in Stockholm and its Research and Development, which consists of the majority of the employees, is in Linköping. The company also has



Figure 13 Zenterio Historical path visualization (source - official web page, see ref.)

marketing and sales offices for different regions such as Central Europe, Southern Europe, Latin America, and Asia in Czech Republic, Italy, Brazil and China respectively (Paragraph based on Zenterio official web site).

Currently there are around 60 employees working as a permanent employee but the total number of employees who work every day on average are around 100, and this include the external consultants and students working in the internship program. Out of the 100 employees 60 are basing in the design centre, Linköping. The number of permanent employee increased from 25 in 2011 to 60 along with the recent companywide expansion. Kaj, the project management office manager, commented on the growth of the company:

“..we are right now on our way going from company that was very stir from our customers that does software development almost acting as a consulting firm moving towards company that will be more product oriented.” (Kaj, Zenterio)

The company delivers a platform or software for Satellite TV (STV) and Digital Video Broadcast (DVB) set-of-boxes; these set-of-boxes include (Internet Protocol) IP and hybrid IP-DVB STVs. Recently There has been a great shift in the company overall direction from developing software and acting as a consulting firm towards product oriented company that provide a standardized software to the entire market with a room for customization to specific customer requirements.

As described at Zenterio official web page, the above mentioned vast experience of the core team helped the company to create Zenterio OS – a reliable, complete and customer value centred digital TV operating system known as ZiDS software stack. The stack is suitable to the new multi-transmission landscape. So far the ZiDs stack has been deployed in more than one million STBs across a large number of network configurations.

In addition to the immensely growing product recognition by the software development industry, Zenterio has implemented a strong yet flexible partnership strategy that allows the company to to support every customer need in the rapidly changing technological landscape. Zenterio’s Marketing and Sales department managed to have wide network of operators, partners and customers some among the clients include, companies of all sizes and nationalities, such as: the world renown Motorola and Nokia; along with Humax- a the leader in the most state-of-the-art set-top box technology; Sagem- a French high-technology group with global operations specializing in broadband terminals, convergence and energy solutions which claims to be No. 1 in Digital IP-TV and digital set-top boxes for terrestrial; Open Tech- a Korean leading corporation that provide best quality Digital STB with a cutting-edge and fully up to date system. Its innovation relies on the comprehensive cooperation with highly professional partners and working with companies like Ericsson, Access, Beesmart, Dreampark, Conax and SourceConnect (ibid).

4.2.2. Agile method in Zenterio

Starting from March, 2011, referring to Kaj, Zenterio have implemented agile software development method in all its software development projects. It uses the agile method particularly referred as Scrum. Before implementing Scrum, the company was using PROPS - a project management model developed by Ericsson . Some crucial external factors such as an increased demand to respond to changing customer requirements and to involve customers are among the reasons for the shift in Zenterio's project management.

Kaj also mentioned that, the need to align and mirror to big industry players such as Ericsson and others who already implemented agile methods can be one among the drivers to join the club. In addition, being flexible and adjusting to the interest of clients does not only improve quality of the software developed but also maintaining a good relation. Fredrik, a project manager, who is also responsible for dealing with clients, was very clear on how focus on client is important:

“So, just a lot of what I do revolves around doings favors for the customer, staff that they ask for is technically outside of whatever the commercial agreement we have, small favors like , “oh could you check this for us?”, or “we are thinking of re-righting this specification, could you one of your guys just look at this? Just to make sure”, we don't paid to do that, but its an hour extra work, we are talking about project which has, you know, scope of 5 or 10, 9 years of life-time of working, sure, why not, I'll have a look. And little pieces of good will like that, matters a lot.” (Fredrik, Zenterio)

However showing “good will” to clients is not the only action to be taken – there is a complexity of how to address client demands, understanding of demands and internal capabilities do deliver prodocuts. As Fredrik continued:

“A lot of managers and project managers like to say we try to give the customer what they need and not what they want, which sounds great, but the fact is that even if you give what they need and not you don't give them what they want, they are still going to be upset, because if they had an expectation.” (Fredrik, Zenterio)

Such close relationship and need of communicating with client might another external driver to operate in Agile set-up. Switching to the internal driver for Zenterio to implement Scrum was; Scrum was already industry proven development method and by bringing such method to increase productivity. The process which was used before the implementation of Scrum, Kaj called in Swedish - “pang på rödbetan”, meaning ‘Start at once and just do things’. The introduction of Scrum has brought a huge improvement in the quality of the product and services that the company deliver and productivity was way better than the previous project management methods.

The different entities, as explained by Kaj, that actively participate in the entire Scrum development are the Project Manager, who is in charge of resources; product owner- representing the customer in creating and prioritizing the product backlog; the Scrum Master- responsible for the scrum meetings and coaching; and the Scrum team for doing the development work that includes- designing, developing testing...etc. The process of development in Zenterio uses an iterative approach with an objective of delivering superior value to customers and devising feedback mechanisms to enhance quality. The development is based on Sprints that lasts from two to three weeks and it involving the following steps:

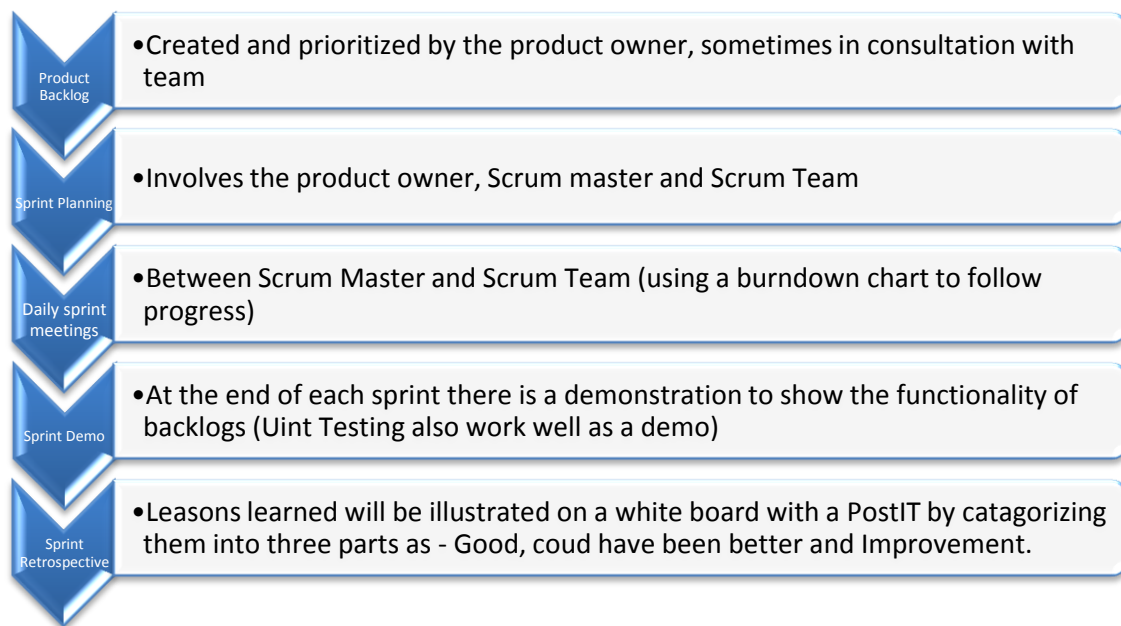


Figure 14 Steps of Agile development Methods (Source: From interview transcript with Kaj, Zenterio)

4.2.3. Projects and Teams in Zenterio

Based on interviewees' response, currently there are around 15 different ongoing projects with some more under pre-study stage. The numbers of Scrum teams vary from one project to another depending on the size of the project. In most projects the numbers of teams are one or two, but there were few projects with three teams. Each team consists of 6 to 8 individuals or developers. However, in some projects, the team size could go down to half of the average, depending on the resource requirement for a specific project. The project duration from start until delivery time approximates to 12 months.

The teams are cross functional with diverse knowledge base and contributing to project work in their own field of specialization. There has been long time debate between different people in the firm whether it is better to have the same competence or different in the scrum teams, but the team composition is still based on varied skills. When there is a need to solve a unique issues or problems that require specialist then

individual of same skill could be assigned together. It is possible to hear from the interviewees that switching to Agile brought some clearness into the work:

“Changing projects and putting team members into other projects mess everything up, so of course it is very important to have a control of the cost and what is happening before starting change and removing people around different project. Because you have to look at the long term, in to what the effects are. But I would say it makes it more visible and see what status in all projects and look into the future little bit better and than other methods.”

(Kaj, Zenterio)

Due to the fact that teams will remain in the same group throughout the entire project period, a special attention is given when teams are formed to have a balanced competence distribution among different teams. After completing a particular project the teams all together switch to a new project without returning to their functional department. In case, there is another project is running out of time to deliver their product, a different team as whole, not individuals, will be assigned to support the project. But rarely when individuals are assigned to support other teams or projects, then they immediately return to their original team after completion of the assignment.

“I would say it is not a problem anymore [allocating resources to a project], since we started working with teams and Agile and Scrum today. But before, it was quit big problem, because there were people from management running around, moving people 80% of the time and people starting fixing the environment and setting up things and one day when they finish that work they should move into another project and start setting that environment up and after four hours they should move back again.” (Kaj, Zenterio)

And that is not the single quotation on the team formation and keeping same teams along the projects, in fact all three interviewees showed strong support of such practice. The team members normally sit in their own office while working but there are a couple of special rooms called ‘Scrum rooms’ where daily scrum meetings are held over and their wall are covered with boards illustrating the progress of a project.

Finally, constructing on the finding from the first case, we asked if the “Fun” part plays any role in the company performance and Agile method. Once again, now in the Zenterio, we could clearly see that people are happy to work that way:

“I feel the same [that scrum is fun], that is what I have experienced, under this quite short period from March last year until now, that we have changed into Scrum method, working like this and people get much more engaged. I think they look happier – what they say is anyway it is more fun to work this way. In some parts this internal education, they educate each other in the teams anyway. .. What I heard, they are very positive not

moving around all days, being kept together in teams and work like this.”
(Kaj, Zenterio)

As was mentioned in the introduction to this case, the authors of this paper also performed an observation of daily scrum meeting (full report can be found as Appendix 1). The aim of this observation was to get insider perspective view one of the core Agile method practice – stand-up meeting. By experiencing the meeting, researchers can better understand how people act in the team, what and how is being discussed, what is the purpose of the meeting. As a consequence, it lets to investigate on the contradictions of Agile method and outsourcing possibilities. The findings of observations will be used in the Analysis chapter of the paper.

4.2.4. Activity Coordination within Agile Projects in Zenterio

The development process begins with customer requirements and this requirement will go through pre-study and broader action plan in order for the project manager to assign it to a specific project along with the resources. Once the project is started detail processes are dealt by the different roles of the projects. The availability of more than one team in the same project put one team to be dependent on another and vice versa. In Zenterio such activities are facilitated through the Scrum Masters who gives assignment for each individual. It uses a sprint plan in order to coordinate the activities in Scrum development, thus involving the Product Owner, Scrum Master and Scrum Team to decide on what to do with in the next two or three weeks; then product requirements (product backlog) is prioritized by the Product owner before each sprint. Every morning from 9:30 to 9:45 there is a Scrum Meeting (Daily Scrum) which is moderated by the Scrum Master in the respective Scrum Rooms and there are three main agendas that each members of the team have to report on: What they have done since last meeting? What they are planning before next meeting? And what barriers are there preventing them from being as effective as possible? (Based on interview transcript and observation)

Even though Zenterio’s stand-up meetings seem to be very short, the developers are located in the same floor and that creates the possibility for the teams to interact with each other when there is a need. Besides all the employees share the same lunch (break) rooms where at least everyone go twice a day to eat or drink a cup of coffee, thus creating a good environment for informal interaction. The product owner is the one who owns the code, thus responsible for controlling the same solutions are not developed for one product or issue. Therefore, on a weekly basis the different product owners meet to update their status and make sure that the same tasks are not done in different teams. The projects use such tools as Burn-down Chart indicating the progress of the project on a daily base and guides how much scheduled work are remaining; and accordingly project manager communicate to customers if there will be a delay or if there is a need to re-prioritize their requirements. At the beginning of the project specification for acceptance test will be handed to Zenterio and agreements will be made on what test procedures and test tools to use; thus simultaneous testing are made

while developing the codes; customers also use acceptance test before receiving the software, in some cases there will be a demonstration to customers when sprints are done in either this company or customers' premises (ibid).

On the practical level, it was interesting to find out how the Agile method helped to improve the everyday work and upcoming work overview, for example Olle was very good at depicting how the work was different previously:

“What I see is that we have more focused approach, so we know what we need to do. And what he had before, was more like chaos when the managers came up running and say: “hey Olle, you have to implement this now it’s very very very urgent and critical. And now we have more, maybe a bit more, overview, also as developers we see in beforehand what was going to come. Yeah, of course you still have those fires, that you need to rush into and just fix directly. But, you have a bit more overview of what is happening. That what I would say is the difference.” (Olle, Zenterio)

Not only software engineers could see the difference, but the changes are notable for higher managers, as Kaj was also well aware of that the problems used to be:

“I think a year ago we all were running to the best guy for everything. Then we ended up having five guys having everything and the others were like they could do something!” (Kaj, Zenterio)

However while talking to Fredrik, a project manager, we could realize that some old practices still remained and he was very open to us in term of sharing what is left to be improved, how the work of project manager and software engineer can contradict:

“We have the facade of an agile organization, we have development teams, they do their daily stand-up meetings, their scrum meetings, right? At least in my teams, we also have sprint planning, we have story boards, we have the backlog, we have points assigned with every tasks, we have burn down charts, we check how much is done. So all of the, sort of fluff around, works well. The stuff, differently put, the stuff that I can affect works well. What still doesn’t quite work, is that the teams aren’t [searches for a word] – ‘self governing’. They should be..” (Fredrik, Zenterio)

As mentioned above, Fredrik was open and self-critical:

“They take ownership of the product, and to large extent it’s my fault, that they haven’t gotten there, because of course whenever things do get pressured, it is really convenient for me to run into someone’s office and say “Can you just do this?” or “Can you just check this?” So, yeah, it’s complicated.” (Fredrik, Zenterio)

Finally, both Fredrik and Olle concluded that “It’s also a little bit of company culture” to run back and forth to each other office, since it is convenient way to communicate and not be bored on long meetings.

4.2.5. Zenterio's Outsourcing/Distributed team in Agile Development

In Zenterio outsourcing part of its development process is a common practice; especially after the number of customer has increased significantly; it became difficult to handle all customers on its own and consequently compelled them to outsource some part of the development processes. However, the decision to outsource or keep in-house a particular part of the development process depends on certain factors. The first factor is whether the particular part which going to be outsourced is related to product or not. After the recent shift of Zenterio into developing a standardized product, the firm prefers to keep those related to the product and outsource those which are of less interest for the company and related only with application developments. The reason for keeping the core is to keep their hand on their product; hence, the agile method allows flexibility to customer requirement by delivering customized product in accordance to their demand.

The second factor for outsourcing is the need for specialized skill; for example, when the company need to add a particular language in the software which is under development, they outsource that particular application to native speaker developers in other company or country. Third factor is the time to deliver product, when the outsourced activity is handed in early in the development process of the project, it allows the internal employees to give enough feedback before the product goes to the customers; Hence, outsourcing decision is sound. But when time to deliver is short and questions on the product are raised from customers after the delivery, the time to respond to customer by going back to third party takes more time; therefore, in-housing the development is desirable. As mentioned before the numbers of employees beyond the permanent one (60 people) come from consultant firms working on specific area, where the projects require specialized knowledge or competence. Here the physical location of outsourcing takes two forms: (1) in-house consultants where the developers are seating in Zenterio's premises and considered as part of project teams and projects; (2) the developers could be working from their premises and there are product managers and technical leaders from Zenterio who support the third parties to facilitate the development. In outsourcing agreements, Zenterio usually try to specify everything before starting the activity, but naturally customers will change their mind after a while and consequently the need to change the requirement in the outsourced activities will also be inevitable.

In Zenterio the developers, project managers and technical leaders are kept together in the design centre, Linkoping. It has a very strict rule of keeping all participants of the development process together in a co-located manner. There is a belief that, if teams are distributed to different locations, then there will be a huge problem of coordination among the distributed teams. In fact, all types of distributed teams are not seen negatively, for example, to have a testing unit or person in a different time zone could complement the process by working during the night and development to continue in

the morning, thus allowing round the clock development. However, Zenterio is still hesitant to work in a distributed manner due to problems related with language, culture and physical distance.

Olle has no experience in working in the distributed/outsourced set-up, neither he has experienced integration of third party developed ‘non-core’ tasks for Zenterio. When asked if he would prefer the company to be more exposed in outsourcing activities or to keep them in-house, Olle replied:

“I think it’s yeah [smiles]. No I think it’s good do things yourself, then you know what you’ve done. It’s easier to follow up, if something is missing, it’s easier just to continue and add those stuff otherwise you need to get into someone else’s code, of course it’s the same if it is internal, but at least you can go and ask that person who wrote the code.” (Olle, Zenterio)

4.2.6. Benefits of Agile Development in Zenterio

Kaj emphasized that Agile method has brought a lot of improvements in the entire software development process of Zenterio. The primary benefit of implementing agile software development method is the usual- flexibility in embracing change in customer requirement. It is almost difficult to set all the specification and requirements at the beginning of the project; since agile method tolerates changes in the middle of the project, Zenterio started to create more values for customers. As Scrum uses visual mechanisms in developing software the entire process of the development became too easy to understand. For example, the use of tools such as boards, post IT note and burn-down chart allowed better understanding of the progress in the development of the software. A year ago with the traditional software development method, it was difficult to predict the status of the project and take corrective action, but now with the help of the mentioned tools, it was able to communicate to customers when there is a delay and re-prioritize their requirement beforehand. Kaj, the PM office manager, was happy to see that company was able to gain control of processes:

“A year ago or more than a year ago we didn’t have control, I would say. [Now] I think we have quite good control over where we are and where we are heading.” (Kaj, Zenterio)

Even more, we could see that this control and clear direction is shared and facilitated by software developers as well, as Olle emphasized:

“we are a bit more focused and know more which way we are going, at least for two weeks.” (Olle, Zenterio)

It is not only control and predictability which matter, in Zenterio working with Scrum is also fun; employees seem to be happy and engaged in their work. It also encouraged employees to share knowledge within Scrum teams. Efficiency and productivity are also among the main benefits that Zenterio enjoyed with Scrum. The sprints are very short and allow the team to control their activities on a daily base, which also involves

a meeting with the customer once every week to follow up the development. According to Fredrik, the project manager, two main advantages of Agile method are distribution of knowledge and visibility. For the knowledge distribution aspect, he emphasized that now a whole team owns the task, no like the one-two most expert workers previously. Visibility is very linked with being on time:

“Visibility, huge! We actually know at a very early stage if we are on time or not, we had no idea before” (Fredrik, Zenterio)

When Olle was asked if he in the same way agrees to the importance of the knowledge sharing, the answer was more than convincing:

“I also agree, that you have information sharing. It’s always someone who knows someone else from you. You will notice in the mornings [refers to scrum meetings] than you raise the question or say that “I am working on this” and someone say “I think they already did that, or I already did that.” Yeah, that’s good thing.” (Olle, Zenterio)

To conclude, it seems that Agile method in Zenterio is not only beneficial, but also hardly causes any serious drawbacks yet – on the questions if there is something negative about Agile, Kaj reflected – “May be there are, I don’t know if I can point out the negative things...” At this point the second case is covered and it is time to move to the last company researchers decided to analyze.

4.3. Case 3 - Ericsson AB

The third and last empirical case, which this research takes into account, is the globally known Ericsson AB. The case in the same line with previous two consists of primary and secondary data. Although, we had a chance to interview one employee – Linda, a section manager closely related with projects, the amount of raw data is limited to do the restrictions of recording an interview. On the other hand, that is compensated with a great number of secondary data available.



Figure 15 Ericsson R&D base in Linköping, (source - authors picture)

It would hard to cover all the areas where Ericsson operates, as the chairperson of the Ericsson board Leif Johansson said in the yearly letter to shareholders (Annual report

2011) concluded that even for him with technical background was hard to create a whole picture of all activities:

“Over the past year, I have been fascinated to realize that the architecture of a mobile system is much deeper and more complex than I imagined.”
 (Leif Johansson, Ericsson Annual Report 2011, p.18)

Neither it would be wise to overload a reader with much of information on the company, since the different and broad secondary data is available (Ericsson official web page; Annual reports; Taxen, 2011; Ivceň & Galinac, n.d.) So instead of building a complete picture, in the case of Ericsson we will concentrate on the basic information and insights researchers could gather during an interview.

4.3.1. Ericsson Company Background

Ericsson is a world-leading provider of telecommunication equipment and services to mobile and fixed network operators. The Swedish giant was founded in 1876 and supplied more than 40% of the world’s network equipments. Ericsson’s headquarters is located in Stockholm and it approximately has 90,300 employees working all over the world. Its presence in over 180 countries and provision of end-to-end solutions for all major mobile communication standards puts the company to be one among the few playing key roles in the evolution of global communication. The products and services that Ericsson provides for its customers are: networks for mobile communication-particularly broadband; Multimedia which focuses on software-based solutions, for real-time and on-demand TV, consumer and business applications, and business support systems(BSS) for telecom operators; and Global Services for operators along with other adjacent industries such as TV and media, Public safety and Utilities; the services includes professional consulting for ICT sectors, System integration, network rollout and customer support (see Figure 16).


	NETWORKS	GLOBAL SERVICES	MULTIMEDIA
	Segment Networks develops and delivers mobile and fixed infrastructure equipment and software. We are a market leader in 2G/GSM and 3G/WCDMA mobile technologies. We now provide all-IP 4G/LTE networks as the evolution of mobile broadband. Our portfolio also includes CDMA solutions, as well as xDSL, fiber and microwave transmission.	With more than 56,000 services professionals globally, we deliver managed services, consulting and systems integration, customer support and network rollout. We manage complex projects with advanced IS/IT competence and multi-vendor experience, using a mix of local knowledge and global expertise.	Segment Multimedia develops and delivers software-based solutions for operations and business support systems (OSS and BSS), real-time, multi-screen and on-demand TV and consumer and business applications. Revenue management, i.e. software based solutions for charging and billing, is part of BSS.
NET SALES (SHARE OF TOTAL)	SEK 132.4 billion (58% of total sales)	SEK 83.9 billion (37% of total sales)	SEK 10.6 billion (5% of total sales)
MARKET SHARE ESTIMATES	38% in mobile network equipment. Twice the size of the second largest competitor	More than 10%. Larger than any of our competitors	Three markets with different dynamics and players
MARKET POSITION	Number 1 in mobile networks	Number 1 in telecom services	Number 1 in real-time charging & billing

Figure 16 Ericsson main business areas (source – Ericsson annual report 2011, p.5)

The case under investigation is Ericsson's unit which was established in autumn of 1987-located in Mjärdevi, Linköping. It currently consists of 1,269 employees; the operations of this unit includes from research and testing facilities to delivery of finished products.

The unit is responsible for several key elements of Ericsson's Global System for Mobile Communications (GSM systems); and also coordinates and directs Ericsson's worldwide product development, support, delivery and other activities in the product lifecycle. Currently the unit is engaged in big project; developing the next generation of mobile telecommunications technology- known as 4G or LTE.

4.3.2. Agile method in Ericsson

Software has been one of Ericsson's success stories for several years. Especially after successfully implementing agile software development in spring – 2007; significant improvement has been achieved in customer responsiveness, reduced lead-time and higher quality. The agile approach plays an important role in fulfilling the end-to-end streamlined software development objective of Ericsson. The drivers for Ericsson to The previous development model, waterfall, with its sequential approach bases its development process on the assumption that the future is predictable. Hence, a lot of time was spent on over analyzing and trying to define the project scope, which led to an elongated pre-study phase. In addition, the lead time for a main release in the development process used to be around 72 weeks, thus limiting the opportunity for customers to create value often delay as early as possible. Normally, the testing in the earlier method was after completing the development, but this also increased significantly the amount of administrative overhead from the rework and waste; plus beyond the very costly back end testing the quality of the product was poor.

The unit in Linköping is now using a Scrum method for the software development process and, thus involves different roles such as; project manager- who is responsible for forming the team, assigning scrum masters and allocating resources in consultation with the different stakeholders; Scrum Master- facilitator and mediator of the customers and scrum team, plus in charge of the scrum daily; Product owner is accountable for each requirements and prioritization of backlogs before the Sprinting process. Scrum team – responsible for developing software.

4.3.3. Projects and Teams in Ericsson

In the unit under investigation there are many projects under progress. The number of Scrum teams with each project is on average 3 to 4; each team consists of a minimum of 5 and a maximum of 9 people. However, the numbers of teams in each project vary from time to time, because a single team may be assigned to different projects simultaneously. This multi-project assignment consequently creates the need for the Scrum team to frequently switch from one project to another, thus creating a significant amount of emphasis to prioritizing projects. In Linda's case, the team normally consists of 7 people; she also believes that 5 is too few people, while 9 might be too much.

We were surprised to find that Linda stressed the physiological and motivational meaning of scrum meetings. In addition, she was couple of times emphasizing the role of proper democratic and team oriented leadership style. Going back to scrum meetings, she believes that the aim of meeting is not only share information, but also greet each other every day, give every member of the team a chance to speak. There is a culture of promoting working together, improving, setting personal development goals, each team is working on how to improve knowledge. But all this improvements and goals should be aligned with project group work in feasible future. As an example she gave, if both Java and C++ are used, and only one person knows Java, more people should be involved and learn Java language.

4.3.4. Activity Coordination within Agile Projects in Ericsson

At the centre of the software development, Sprints serve as a mechanism of coordinating the different activities. A single sprint with a new team will last within 2 weeks, because the team will need a short feedback for every release; but when teams have stayed together for a long time the sprint duration may last in 3 weeks. There are two types of prioritization are done before every development: Firstly, each customer requirements, backlogs, will be prioritized for the different sprints of the projects that teams are working for. Secondly, project prioritization will be made so that each team addresses the requirement in different projects.

When the conversation turned to the managerial skills of running a team, Linda told that knowledge and skills, possible learning, - all should be aligned with the team work plan for longer horizon. Manager is responsible for forming a team, setting up a team is complex issue, and that is why manager gets some support from others, also includes developers into the process. The leadership style is group oriented, not autocratic. Linda has a strong attitude toward “firefighting” situations which could result in abandoning some of agile principles, she says that it is not practices here, since once you do that and show such attitude towards the team, you will spoil everything.

Such attitude and mature view on managerial issues seems to correspond with Ericsson declared ambitions of high quality project leaders on the whole company level:

“Local services competence and highly skilled project leaders are both prerequisites for success in telecom services. Ericsson has invested USD 1 billion in processes, methods and tools in order to secure common global frameworks and ways of working.” (Ericsson annual report 2011, p.23)

In total, even during one single interview we could see a strong difference in team management style, attitude and values to the compared other cases. Ericsson seems to have strict ‘rules’ of the game and stick to them. On the other hand, the practice of assigning team to several projects at the time is not pure Agile method.

4.3.5. Ericsson's Outsourcing/Distributed team in Agile Development

For a global player like Ericsson with an immense operation, being involved in outsourcing is critical and inevitable. Ericsson with its long stand of delivering products and services to a broader customer base by combining local capabilities with the global expertise has long tradition of working in distributed as well as outsourced software development. Some among the distributed development units are located in Sweden, China, Italy, Canada and Germany. Even more, four large scale network operation centers where established round the globe to provide high efficiency to network operators:

“We provide efficiency by drawing on our global scale. Our four global service centers all house global network operation centers (GNOCs) for remote delivery of network management. These are based in Romania, India, Mexico and China. As an example, more than 20 European operator networks are run from the GNOC in Romania.” (Ericsson annual report 2011, p.9)

By operating such operation centers Ericsson is able to promote the strategy of providing services for local operators, in other words to be a provider for outsourced tasks by operators, and they believe that this is an “opportunity to generate more business”. The Nordic region is mature and well covered by 3G technology and new LTE technology is commercially available. Operators in this region are structuring their business models into sharing networks and outsourcing network operations.

On the other hand, on the part of the production Ericsson themselves actively use outsourcing. Production of electronic modules and sub-assemblies is “mostly outsourced to manufacturing services companies, of which the vast majority is in low-cost countries.” (Ericsson annual report 2011, p.41) While manufacturing of radio base stations is normally performed “in-house and on-request. Final assembly and testing are located in Ericsson’s 17 manufacturing sites in Brazil, China, Estonia, Italy, India and Sweden.

As turning back to Linköping R&D center, Linda reflected that they have some development and experience with projects run in China. But she still strongly believes that every team needs to be co-located and working with the same backlog and sprint. If such conditions are fulfilled it should be possible to run projects parts in different countries, however also competence matters a lot. One of the reasons, why to outsource is to seek for some particular competence. Overall, the level of case unit involvement is quite limited, however on the company level it is very high.

One of possible reasons for difference of outsourcing levels might the risk management strategy, since so many business operations of the company are outsourced it could be too risky to outsource core R&D functions to big extent, since as Ericsson declare that company very much depend on “security and reliability measures of external

companies. Regardless of protection measures, our systems and communications networks are susceptible to disruption due to failure, vandalism, computer viruses, security breaches, natural disasters, power outages and other events.” (Ericsson annual report 2011, p.121)

4.3.6. Benefits of Agile Development in Ericsson

Ericsson is an organization very much involved in Agile practices: on their web page it is possible to find dozens of links to articles, conferences and employee experiences on that area. Moreover, employee at Ericsson say that whole company is steering into agile management philosophy, whatever it could be. Agile development empowers the employees and creates an environment where everyone controls their own work. Strangely, but there is no single word on Agile in the Annual report, it could be considered a specific term to be used for wider audience.

Linda, a section manager, pointed that the organization cannot afford to have time boxes with planning the release in 6 months, now market and customers want to have it constantly. Development has to be running on the program basis. Linda’s own teams work in agile for only couple of months and getting much more involved into “cross work” between different departments is seen as a big benefit of new method.

When we asked about possible challenges the unit faces in new set-up compared to tradition project management, we heard that most emphasized issue is the work in cross-functional teams. Some people, according to Linda, feel a lot of pressure due to the fact that they can not specialize only on one area, in a sense they used to work previously. Most likely, many people will have to learn new skills. Moreover, since such pressure on some individuals exists, tasks and work should be divided carefully, not to decrease the overall motivation and spirit of individuals and team, and this requires a careful approach by project leader.

5. Analysis

In this chapter of the thesis, a reader can find the analysis part, which is the last part, before authors conclude on the research findings. The analysis is based on both theoretical frame of references and empirical findings in three cases under investigation. The paper is structured in the way that the prime aim of researchers is to answer the research questions. The path we have chosen to do that is to try to identify some patterns and group them in the important themes (Creswell, 2007) so that the questions could be answered from the diverse angles.

The ‘route’ of developing analysis, in short, is following: starting broad by using visualization analysis method of word cloud, then address research question one, later introduce a framework, and finally the framework will guide us to the second research question. The conclusions and discussion will be presented in the final chapter.

5.1. Using word clouds to analyze empirical data

Since the three empirical cases are still fresh on the mind of the reader, the analysis will start with general overlook of all five conducted interviews, using the method of creating a world cloud (tool: www.wordle.net). As McNaught & Lam (2010) suggest word cloud is ‘fast and visually rich’ method which emphasizes often used words by making them occupy ‘more prominence in the representation’ and recommended to be used for primarily analysis.

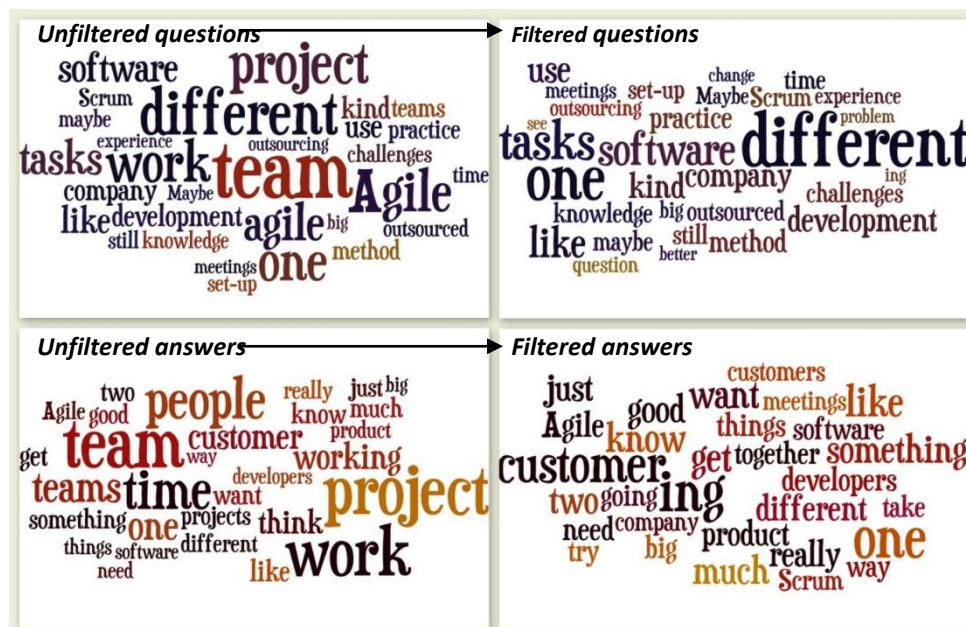


Figure 17 Visualization of question asked and answers received (Word cloud tool: www.wordle.net)

As we have asked in total more than 85 questions, it is possible to identify some patterns in those questions we asked and answers we have received. The procedures for

preparing a visualization were: first, all answers were put together (around 12,500 words), general words as “people”, “team”, “teams”, “work”, “think”, “time”, “project” and “projects” were filtered out and the result was limited to 30 words. For the questions part (around 2,500 words), similar procedures were performed, filtering out words: “team”, “Agile”, “agile”, “project”, “work”. The results can be seen in the previous page, Figure 17.

As a result, after removing some obvious words, we could see that there are some patterns in the picture. Some words were both in the question and in the answers and show the consistency of the dialogue between researchers and interviewees. These words are: “Scrum”, “development”, “different”, “company” and “software”.

The main **finding of word cloud** visual method can be interpreted in following – researchers asked about outsourcing (two words in cloud: outsourcing, outsourced – combined effect would result in larger font word), but interviewees did not use that word often (possibly avoided topic or terminology). This shows that there is a lower level of outsourcing activities than expected. And second, researchers were not asking questions regarding customers (one of the reasons is research ethics), however interviewees very often referred to their customers and products. Customer focus might be one of key reasons for steering organizations and firms into Agile method. This is an important finding, and will be helpful in answering research questions below. However, word cloud method has its own limitations – words are ‘retrieved out of context’ and its method is unable to distinguish ‘meaningful phrases’, hence a whole method is not recommended as a stand-alone analysis technique (McNaught & Lam, 2010). As a result, after removing some obvious words, we could see that there are some patterns in the picture. Some words were both in the question and in the answers and show the consistency of the dialogue between researchers and interviewees. These words are: “Scrum”, “development”, “different”, “company” and “software”. The rest of the words, excluding some common English words, were different and grouped according to area.

As we are moving to the answering the research questions, first the coding and grouping of the empirical data was performed. Key answers by interviewees were grouped in 5 areas: Information on the case (8); Agile benefits and challenges (14); Team, Scrum and Process (31); Outsourcing benefits and challenges (15); and Recommendations (9). While first group was already used in case description, and last one to appear in later in recommendations, the analysis, in term of empirical data will rest on: Agile issues; Team, Scrum and Process; and Outsourcing issues groups. Followed after the grouping, themes identification took place and research questions were partly uncovered. Apart from that theoretical part, observation and secondary sources will be included in the analysis.

5.2. Drivers for introducing Agile method

The first theme, which was quite obvious to spot, is about the reasons behind the fact that all three case firms switched to Agile methods to some extent. Charette (2003, p

19.) suggests that development in Agile set-up is based on five key principles: “support the customer; keep things as simple as possible; embrace change; account for behavior; and build virtuous cycles”. In practice, based on our interviewees’ responses, it is possible to see that four main drivers emerge as a reason for organization to turn into Agile software development; these are: (1) Popularity, trend; (2) Engaging employees, fun and socialization; (3) Mirroring big customers and (4) Pressures from the market for short time product delivery (see Table 5).

These four identified drivers are quite different in their nature to the drivers of cost advantage or large employee pool, which are often reasons for global software development (Carmel & Agarwal, 2001). On contrary, the case firms are rather embracing some of Agile Manifesto values and principles: “to satisfy the customer through early and continuous delivery of valuable software”; and “the most efficient and effective method of conveying information to and within a development team is face-to-face conversation”. (Agile Manifesto principles see page 27)

The popularity and trendiness of Agile method are very much linked with the ‘human’ part of engaging employees, making work fun and letting people to socialize, or even work in pairs. Kaj, Denis, Olle and Linda in a different way expressed the fact that employees become happier, more engaged into work, even learn from each other. It was quite easy to see during Scrum meeting observation, that people were happy, smiled often, learned from each other and the whole atmosphere was inviting for interaction.

Mirroring big customers work methods is an important driver, as report by the Economist Intelligence Unit Limited (2009) previously suggested that in current environment, “the ability to satisfy customer expectations is core to profitability. If you’re not agile, you can’t do it, because customer expectations are never static”. And that is exactly what researchers could see in analyzed cases: GGA Software Services are working close to big pharmaceutical firms, Zenterio is following the trend in big Telecom firms, and Ericsson is implementing Agile in order to meet demands by network operators and final consumers.

In case of Zenterio’s project management method change, the necessity to meet ‘never static’ expectations of customer is a complex task for project manager and has to be addressed by often meetings and physical presence, as Fredrik described:

“..but the fact is that even if you give what they need and not you don’t give them what they want, they are still going to be upset, because if they had an expectation... so, just a lot of what I do revolves around doing favours for the customer, staff that they ask for is technically outside of whatever the commercial agreement we have..” (Fredrik, Zenterio)

And this is exactly what Wysocki & McGary say about the hard work with clients’ needs: “Sometimes those needs can be very different than the original wants.” (2003, p.

xxiv) Moreover, as we will see in the second research question, Fredrik had a strong attitude that project work and software development should be close to the customer.

Table 5 Drivers for using Agile method in case firms' projects

Identified drivers for using Agile method
(1) Popularity, trend
(2) Engaging employees, fun and socialization
(3) Mirroring big customers
(4) Market pressures for short time product delivery

Finally, as we talk about Pressure form market for short time product delivery, the Agile method, as a concurrent type of project, has a big advantage of being able to deliver in short time (Tonnquist, 2008). It is reasonable to concentrate on fulfilling customers' key demands and do it as fast as possible, since the work normally expands "to fill the time available", as Parkinson's Law says. Even though, 'time-to-market speed' and 'round-the-clock development' are seen as an advantage which fuels the trend of global software development (and outsourcing as consequence) (Herbsleb & Moitra, 2001; and Karolak,1998), it was possible to see that based on the interviewees answers, our case firms prefer close team allocation, short communication distance and face-to-face communication as a tools for quick product delivery.

To sum up the first identified theme, is it possible to say that drivers for switching to Agile method act as a lock-in for not outsourcing tasks and work within software development projects? It might be too early to say, that is why we move to the next theme, were the Agile practices will be analyzed in the light of outsourcing trend and possible challenges.

5.3. Emphasized Agile values and practices compared to Outsourcing trend and challenges

Much has been said about change and necessity to respond to the change (Van de Ven & Sun, 2001; and Sharp et. al, 2000). And if seeing change, as an operational factor within project level, then taking in account Mohammed at al. (2008) that such 'operational' factors are both 'barriers as well as enablers' for effective outsourcing, the response to the client's and environment change in Agile projects can both increase or prevent outsourcing. Moreover, Gower Handbook of Project Management (2000) provides alternative notion of project novelty -"Thus projects are novel endeavors, undertaken to deliver new development objective". In such case, if Agile project is a ever changing novel endeavor, it is much more likely to be kept in-house and serve as a barrier to the outsourcing.

Talking about the Agile Manifesto Values (presented in page 26), it is possible to see from the empirical data that at least 3 of 4 values are placed within case firms.

Researchers could observe during the Scrum meeting that ‘Individuals and interactions’ were prioritized over ‘Processes and tools’; while talking to project managers Denis and Fredrik, it was obvious that ‘Customer collaboration’ is more important than ‘Contract negotiation’; and finally interviewees in higher positions, like Kaj and Linda, responded that ‘Responding to change’ is done, rather than ‘Following a plan’. The only value researchers could not check is ‘Working software’, since as said in the Introduction, the technical part of project work is outside the current study.

Meanwhile in many textbooks on Outsourcing, authors stress the importance to carefully design the process and tools implaced, negotiate on contract, demand well provided documentation and basically stick to the plan of executing outsourced activity. For example, Dalcher in Brudenall (2005), suggests on contract negotiation: “Establish a contract, including full scope, duration, type of relationship with the vendor, clear boundaries, clearly defined performance criteria,..” etc.

As a result, we can see some fundamental contradictions between values of Agile software development and possibility to outsource some parts of project work. Next, we are going to review identified practices, used in projects of our case firms, and see if they contradict with outsourcing concept.

In short, after analyzing the grouped and themed answers of interviewees, six key practices were identified: (1) stand-up meetings; (2) Burn-down charts and visualization; (3) Constant contact with customer; (4) Co-located teams; (5) Knowledge sharing and focus and (6) Fun and Developer involvement. As the umbrella word for these five practices, in case of our analyzed firms, we can say that is Productivity. The best example, of how productivity improved efficiency and control is the answer of Kaj regarding the transformation within a firm:

“My feeling is that we are getting more efficient, more productive; we have better control with sprints, because sprints are much shorter. A year ago or more than a year ago we didn’t have control, I would say. [Now] I think we have quite good control over where we are and where we are heading.”
(Kaj, Zenterio)

5.3.1. Stand-up meetings

The performed observation of Scrum meeting served as strong source of understanding how the team interacts for researchers of this paper. In general, the meeting was performed in same manner as described by Sutherland and Schwaber (2011) – questions on accomplished tasks, next day work and possible obstacles were raised. We could see that meeting was not stressful, people were dressed casually, could great each other with applauds and debrief shortly on work status. Moreover, engineers were interested in listening to each other, raised questions, and helped colleagues with valuable advice. The simple version of visualization was present, the tasks were on the board and progress was shown. A variety of technical tools in the room enabled

meeting participants to refer to actual software code in the TV system right away during a meeting.

And such powerful usage and beneficial results to the team can be observed in other cases, not only Zenterio. As Denis from GGA emphasized: “Stand-ups are fabulous thing!”. In Ericsson case, Linda stressed the importance of meetings as motivational and team-building issue in addition to work progress.

Keeping all these issues in mind, it hard to see how outsourced part of team or a whole team could use the stand-up meetings to the same extent. Both Kaj and Denis stressed, that project leader should be well aware of project progress, and in distributed manner it would be extremely hard to keep the same communication speed. Moreover, as Olle pointed out, the meeting should be brief and the additional information between project members or other team members is shared face-to-face, otherwise meetings would be long, boring and not productive.

5.3.2. Burn-down charts and visualization

The visualization is an important empowerment for team motivation and is a key part of ‘system anatomy’ (Taxen, 2011). As was mentioned previously the performance of Zenterio was improved since efficiency and control were regained since the introduction of Agile method. In that particular firm, differently from others (could be due to the different depth of analysis by researchers), burn-down charts, forecasting delays and delivery time were emphasized by employees of all level. Visualization and forecast were both present in daily Scrum meeting rooms, weekly conference meeting rooms and employees computers.

Using burn-down charts and visualization should not be a big problem in case of offshoring the project parts. However, in case of outsourcing to the third party vendors, they should allocate some teams solely on one project and go deeper in tracking progress and delivery times.

5.3.3. Constant contact with customer

The world cloud analysis results show that the focus on customer was raised by interviewees on their own initiative. Hence it shows what crucial role customer-focus plays in their industries. Moreover, the Agile method enabled firms to keep constant contact with clients with a mean of often meetings and work prioritization. The relationships were also improved, since the visualization of progress, and as a result early communication on not meeting deadlines gives clients a chance to steer the process and choose what should be accomplished.

We could also see that particularly in case of GGA and Zenterio, their customers offer software development due to the particular expertise these firm can offer and improve the internal development previously done by customers. Meanwhile, as Fredrik strongly insisted, the customer wish to have close contact and what is very important – fast response time. That is why involving a third party might be risky, since such

response time can prolong from current 45minutes to 3-7 days – and that ‘would not look good’ concludes Fredrik.

5.3.4. Co-located teams

As previously discussed in the Chapter 2, the proximity theory is important in dealing with projects and collaboration within team. The close location of employees allows team members to have real-time observation to project work progress (Hoegl & Proserpio, 2004). Moreover, according to a mentioned study conducted by Dybå and Dingsøy (2008) on ‘empirical studies of agile software development up to and including 2005’, it is not obvious to conclude that, Agile method could be efficiently used for large distributed projects. As an empirical evidence of challenges in distributing project teams, we can refer to Denis:

“they are over deadlines and keep on hold each other, one team can work efficiently on time, other is not that efficient and it will take more time – naturally you have to find what job should be done by the first team in meanwhile. The problem is that, they perception can be different, they will not ask each other how you did it, not see how others worked, it will be more time spend, more bugs will come out.” (Denis, GGA Software Services)

In general, information systems area employees attitude towards outsourcing might vary from negative to positive (Mol, 200). The reason is primarily that outsourcing can cause the rise of complexity in system in order to integrate data from third party. At the same time this might be a new and interesting challenge for technical people. However, if we take in account Olle’s view, his is quite negative towards outsourcing:

“I think it’s yeah [smiles]. No I thinks it’s good do things yourself, then you know what you’ve done. It’s easier to follow up, if something is missing, it’s easier just to continue and add those stuff otherwise you need to get into someone else’s code, of course it’s the same if it is internal, but at least you can go and ask that person who wrote the code.” (Olle, Zenterio)

And that is not a surprise, since Festinger et al. (1950) argued that the more geographically team members are dispersed the less social integration they will have and that will lead to few opportunities for knowledge acquisition from team. On the opposite, the strong resistance to outsourcing might be seen as two syndromes – “not invented here” (NIH) and “not sold here” (NSH) (Chesbrough, 2006). NIH is an internal resistance based on xenophobia, it blocks internalization of ideas from elsewhere due to the fact that it is not from us, different and can not be trusted. NSH syndrome is based on mentality if we do not find any use out of technology, no one can make it either. In longer run, if such syndromes are present, it could result in lost opportunities to learn and co-operate with other firms in the industry.

5.3.5. Knowledge sharing and focus

Distribution of knowledge within teams and visibility of upcoming activities were voiced by most of the participants. There is a greater opportunity for learning among team members through face-to-face communication (Meyer, 1993) as it provides quick feedback and enables understanding complex information (Katz & Tushman, 1979). In case of Zenterio, it seemed very consistent since the knowledge sharing and visible projects were identified by all participants, as Kaj concluded:

“I would say it makes it more visible and see what status in all projects and look into the future little bit better and than other methods.” (Kaj, Zenterio)

The additional and surprising finding was that knowledge management within developers is not the only knowledge which is being transferred. It was obvious to see that some transformation in managerial approach and techniques happened in all cases. That is very much in line with Nerur et al. (2005) identified challenge of changing the role of leadership from command and control to support and collaboration. For example, Linda in Ericsson was emphasizing the importance of democratic and team oriented leadership style. She was very clear on the fact that she would not decompose the team in order to meet the ‘burning’ project, since it would not be proper and whole idea of Agile would suffer. Meantime, at Zenterio, such ‘firefighting’ is still happening from time to time, while at GGA it seems to be an often used practice. It seems that, the more the firm is using Agile method, the more management is supportive and teams are self-aware of being committed to the task. This is an interesting finding, and could be developed by future research.

On the outsourcing part, some of the challenges identified by participants such as longer communication distance, cultural and language barriers and inability to interact face-to-face – most likely would not contribute to the knowledge sharing within team or management in cases where project parts are outsourced.

5.3.6. Fun and Developer involvement

Agile method is in many ways about fun, even Scrum three groups of phases called as Pre-game, Game and Post-game (Schwaber 1995; and Schwaber & Beedle 2002) give a hint that this work method is close to game, joy and fun. In most cases, we could see that employees were happy and glad to work in Agile method; often compared it with tradition project management and picked up on positive change towards Agile. Moreover, during observation and interviews, researchers could see that claimed practices are in fact used in regular ‘developer’s life’.

In addition to that, Denis, raised the idea that despite the engagement and socialization, it is more fun to work in Agile set-up for fresh graduates and younger people:

“..Agile allows, different from Waterfall, to emphasize best practices and use it in life, in life of developer. Waterfall was giving big steps, what to do and when. Agile is oriented towards students and post-students. These guys

will to work a lot, to go further. Mature people are not that good in Agile set-up, they don't have that much drive. Young guys need a game, that is why for them Agile works. They want to communicate, they want to gather, socialize, to share same work then one develops the other double checks it. It should be Fun!" (Denis, GGA Software Services)

Kaj, then asked on the 'fun' part of the Agile projects, had nothing but to agree:

"I feel the same [that scrum is fun], that is what I have experienced, under this quite short period from March last year until now, that we have changed into Scrum method, working like this and people get much more engaged. I think they look happier – what they say is anyway it is more fun to work this way. In some parts this internal education, they educate each other in the teams anyway. .. What I heard, they are very positive not moving around all days, being kept together in teams and work like this." (Kaj, Zenterio)

Table 6 Identified Agile practices and following contradictions with outsourcing trend

Identified Practice of Agile team	Contradiction with Outsourcing trend
Stand-up meetings	Physical distance from other teams; Long communication distance; Necessity for excellent Project Leader
Burn-down charts and visualization	Possible to use practice in case of allocating a team working solely on one project
Constant contact with customer	Long response time for customer questions; Difficulties in reaching people over Skype, phone, email etc.; Risk of losing product range and control of development
Co-located teams	Lack of communication and less interactions; Difficulties in finding compromise between developers; Lack of trust and collaboration
Knowledge sharing and focus	Differences in work perceptions; More time spent and more bugs to fix; Limited option for learning and sharing knowledge
Fun and Developer involvement	Cultural and language differences; Inability to interact face-to-face; Missing informal interactions

To sum up, the identified practices of Agile team, in the analyzed cases, to great extent contradict with Outsourcing trend (Table 6 above). Since both introduction of Agile method in all three cases was linked with will to increase productivity, while at the same time Outsourcing is also described as productivity improvement technique, we could conclude that they are rather contrary in the nature. Both Agile and Outsourcing can be seen as a choice of between two strategies. At this point a notion of trade-offs of outsourcing (Berggren et al., 2011) can be introduced. As for example if Agile and Outsourcing both improve performance, but the firm would like to keep tight contact with customer or constant knowledge sharing, choosing Outsourcing would diminish these two benefits.

Despite the fact that all three cases are not fully engaged in Outsourcing and as we found it contradicts to large extent to Agile set-up of projects, the Outsourcing has a high value due to its potential identified by our interviewees: involve external expertise and competence, utilizing the time difference and working ‘round-the-clock’ and improve quality and productivity. Additionally we have also observed that case firms we transformed into Agile method to different extent – some were rather selectively choosing single practices, while others almost fully followed the designed method. Keeping these two issues in mind, at this point we would like to introduce the framework in the form of visual 2x2 matrix (Figure 20).

Towards framework

To start with, the developed framework/model rests on two principle findings: first empirical – we were observe notable differences among case firms, second theoretical – based on the concept of Fuzzy logic developed by Lotfi Zadeh (1965) the theoretical understanding that firms cannot be purely Agile or not Agile at all (same as example with water, which is not only cold or boiling – it can be worm); as well as company is never totally outsourcing relying or not involved in outsourcing at all. Contrary, based on the Fuzzy logic and real observations, firms are Agile and Outsourcing to some level: not just yes and no, but rather from low to high level.

Now before placing particular firms into matrix, it is important to discuss what the criteria the model is based on are. First,

LOW	LEVEL OF OUTSOURCING (CRITERIA BELOW)	HIGH
FEW	Nr. of ACTIVITIES (Mol, 2007)	MANY
COST	REASON (Berggren et al., 2011)	SUPPLIER KNOWLEDGE
NON-CORE	ACTIVITY (number of Authors)	CORE

Figure 18 Criteria determining Outsourcing level

outsourcing level in term of number of activities outsourced vary depending on the industry and company strategy for the search of ‘optimal performance’ as Mol (2007) suggests in his ‘outsorceability’ dimension model (see Figure 4 p.17). In addition to that, previously mentioned 3 waives of outsourcing (Berggren et al., 2011) include three types of outsourcing: ‘manufacturing outsourcing’ (reason – cost), ‘component outsourcing’ (reason – supplier knowledge) and ‘capacity outsourcing’ (reason – mix of previous two). We assume that in project set-up, outsourcing for cost, or as some interviewees identified – due to shortage of time is representing lower levels of being involved in outsourcing on project level work. Meantime, deliberately outsourcing for the supplier knowledge, or as interviewees called competences and particular expertise, would stand for higher levels of outsourcing. Finally, Gilley and Rasheed (2000); Dhar and Balakrishnan (2006); and Mol (2007) discuss on core versus non-core status of activities being outsourced. Even though, Mol is able to conclude that such perception can be dangerous for the firm, in practice we still observe managers to use this rhetoric.

We place non-core activities as low outsourcing level, while outsourcing core strands for high level. The summary of outsourcing dimension criteria can be seen in Figure 18.

As for the Agile method, the level to which firm is using it can be determined first by the consistency of how well the method is defined (Turner & Simister, 2000). As for the project type, as presented earlier (Table 3, see p.25), it is believed that organizations should use Agile practices for technology development projects, while classic methods are more suitable for operational projects (Chen, 2004). Product/process development projects can choose between two types. Moreover, the Agile Manifesto values (Beck et al., 2001) followed by the firm to a large extent represent the level of consistency of Agile set-up projects. Finally, as suggested by Nerur et al. (2005) switching to Agile Method requires a change of managers' leadership style from command and control to support and collaborate. Hence, the supportive and collaborating leadership style stands for high engagement to Agile method. The summary of mentioned four criteria can be found in Figure 19.

LOW	LEVEL OF AGILE SET-UP (CRITERIA BELOW)	HIGH
POORLY	METHOD DEF. (Turner&Simister, 2000)	WELL
OPERATIONAL	PROJECT TYPE (Chen, 2004)	TECHNOLOGY
OTHER VALUES	Beck et al., 2001	AGILE VALUES
COMMAND&C.	LEADERSHIP (Nerur et al., 2005)	SUPPORT&C.

Figure 19 Criteria determining Agile set-up level

As we have determined the criteria for both dimensions, it is possible to place our case firms in the particular area on the framework model (Figure 20). The model is developed inspired and partly based on previously mentioned 'goals and methods matrix' (Turner & Simister, 2000) and 'outsourcability model' (Mol, 2007). The legend on the right of main matrix explains the depth of study in each case firm, the time spent during interviews and observation is not the single criterion, but can represent the depth and probability of correctly putting the firm on the matrix are (the larger the bubble, the higher the precision is).

It has to be mentioned that all firms ARE using Agile methods and Outsourcing, so under no meaning LOW stands for NONE, neither it can be seen as POOR. As mentioned previously, it depends on each firms strategy and industry to which extend they want to utilize both options. However, based on the broad primary and secondary data on each case presented in Chapter 4, we place each firm accordingly: GGA – top left square (LOW, LOW), due to few activities outsourced mainly for cost reason and using 'selective' practices with control management style; Zenterio – lower left square (LOW, HIGH), due to outsourcing few non-core activities and using well-defined method, practicing Agile values with supportive management; Ericsson – in between

two lower squares (see model), due to outsourcing many core and non-core activities with mix of cost and knowledge reasoning and using quite well defined method, some non-Agile values and supportive management style.

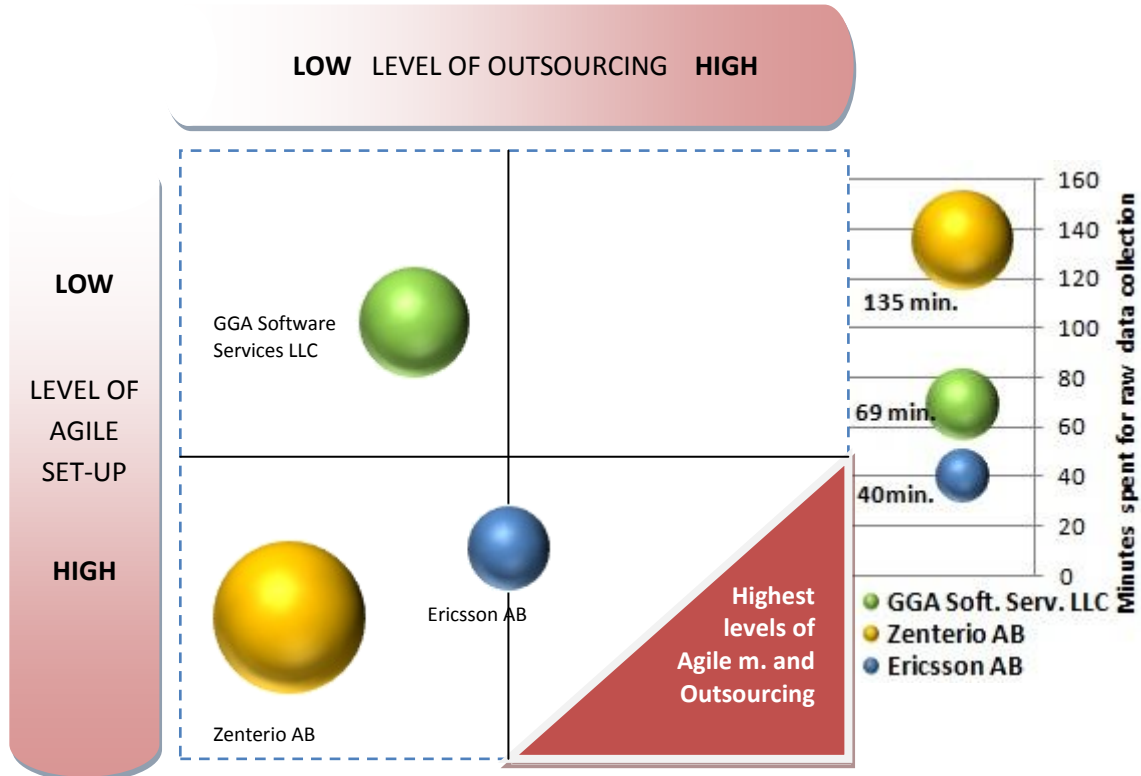


Figure 20 A two dimensional Model of Agile-Outsourcing levels

When analyzing the overall model including each case firm's placement, it is obvious to spot that two of four squares are almost empty. First, the top right square is empty, possibly due to the fact that according to research method purposive sampling was used and firms with Agile projects were selected. Lower-right square standing for High Agile set-up and High Outsourcing level is empty as well. That is not surprising, since the previous analysis identified that using both options is complicated due to number of contradictions. However, the most extreme part, represented as a red triangle in the model, should be seen as a point of interest for many firms, not only limited to our case firms.

The interest is fueled-up due to the number of benefits both practices can bring to the firms and project work; on part of Agile – ability to adopt to change flexibly and fast (O'Riely, 2009; and Cockburn & Highsmith, 2001), ensure regular collaboration with customers (Bohem, 2002), embrace human role, relationships and community among developers (Abrahamsson et al., 2002), and keep things simple in general (Charette, 2003); and on the side of Outsourcing - improve competitiveness (Gilley & Rasheed, 2000), higher quality, customer satisfaction and time to market (Dhar & Balakrishnan, 2006), reduction of assets and costs (Mol, 2007), specialization across supply chain (Berggren et al., 2011), sharing and re-using external ideas (Chesbrough, 2006), and playing strategic importance (Mohammed et al., 2008).

Since the motives of being in ‘red triangle’ are uncovered, it is time to move to second research question, which addresses the identified pre-conditions on how to design project work in order to enjoy benefits of combining Agile methods and Outsourcing simultaneously.

5.4. Outsourcing implications on Agile development Method

In agile software development, teams are considered to be globally distributed when interdependent workgroups consists of culturally diverse members and reside in more than one country during the development process (Cohen & Mankin, 1999). The cases under investigation have at least one development team each located in a distributed manner either in the same country or a different country.

The importance of outsourcing software development is increasing ever than before; in all the cases that we have investigated there has been considerable increase and positive attitude towards outsourcing. Distributed software development has become inevitable in the software development due to several reasons. Bartelt et al. (2009) claimed the global software development promises three main opportunities : Economic- as means of improving efficiency; Organizational- referring to the natural structure for firms which operate in a global network to have a globally distributed teams working for the same project; and Strategic- emphasizing on local presence to different markets to improve time-to-market.

In fact the drivers, for the different companies and their projects, to go into outsourcing vary from one another- for example: in Zenterio’s case as explained by Kaj ‘One of the reasons is that we have so many customers that we are not able to handle everyone in-house, we need help from outside’. Mol (2007) argued that outsourcing and international trades can be related as ‘two sides of the same coins’ (p. 14); and this argument well confirm Zenterio’s recent companywide expansion reflected in an increase of employee from 25 in 2011 to 60 along with its globally distributed customers.

For Ericsson as Linda put it - ‘competence matter a lot! One of the reasons why we outsource is to seek for particular a competence’; According to Berggren et al. (2011) ‘Firms are rarely self-sufficient and therefore need to integrate complementary knowledge of other firms in order to fully exploit their own resources and capabilities’ (p. 148). Linda has also affirmed that Ericsson cannot possibly possess all skills and competences in-house, thus outsourcing to a third party for a particular part of development is common practice. Similar rationale has also been presented by GGA with respect to outsourcing; Denis emphasized on the importance of quality in the software development and when the competence required to deliver such quality is lacking then outsourcing become a viable option.

Though all the above mentioned reasons force companies to work in an outsourced set-up or involve in projects with distributed teams, there still need to put in consideration the challenges associated with it. Karolak (1998) argued that globally

distributed projects are more challenging than even complex in-house projects. Hence, recognizing challenges that projects may face in a distributed team set-up help the participant to devise better mechanisms to deal with the problems.

The distributed development, according to Ågerfalk et al. (2005) has three implications in terms of distance: geographical distance, socio-cultural distance and temporal distance. Three major challenges are identified due to above aspect of distributed team: communication, coordination and control (ibid). Both Kaj and Denis pointed-out that the cultural and temporal distance implication on communication, language and time lag.

“There should be a constant communication between people, to update them on all changes. ... time lag, it is difficult to communicate with Americans at midnight, you want to sleep; documentation writing, in Agile you don't need it, but then should have a 24h reachable person all the time to explain details; some language issues...” (Denis, GGA)

However, Kaj emphasized the importance of time difference in some cases; one of the benefits that globally distributed development as mentioned by Herbsleb and Moitra (2001) was ‘round-the-clock’ development:

“... time difference could be good sometimes, when it comes to test development it is very good. Having development at one place and testing at nights somewhere else; and then start developing again the day after ... I had some good experience of putting some test staff outside in other time zone... round the clock working.” (Kaj, Zenterio)

Our study shows that distributed software development is inevitable for the companies under research: Zenterio, Ericsson and GGA. In addition, Agile development seems to be the preferred software development methods by all; Hence, the result of the study as shown above indicate there is both positive and negative implication of outsourcing on Agile development.

5.5. Preconditions for Outsourcing Agile Projects

Most of the literatures are written on what agile methods are and what challenges may affect the smooth development process of agile projects in outsourced and or distributed set-up, but less attention is given to what needs to be considered while thinking to outsource part or the whole process of the software development. One of our research questions was about identifying the preconditions of outsourcing agile projects. Here below are the different factors that have to be put in mind while deciding to outsource agile projects:

5.5.1. Not Product Related Development

The first criterion that we identified during this study was related to core versus non-core issues in outsourcing software development. According to Berggren et al. (2011) outsourcing enables to focus on core competences. Even though Mol (2007) was able

to come up with a persuasive discussion into the idea ‘those non-core activities should be outsourced’ as “fundamentally incorrect” (p. 148), in reality still companies seem to be using the logic in outsourcing decisions. Kaj stated that the main criterion for involving in outsourcing is whether it is related to the core product or not; if the part to be outsourced is not related with the core product or of less interest to the company then it is alright to outsource it. Kaj described it as follows-

“... some kind of customization that is not really product related, the thing that we don’t need in our own product into the platform, that will be outsourced as much as possible..., if it is really product related then we do it in-house, and when it comes to some other application stuff and some other stuff that we are not interested in, and the customer requires it then we outsource it.” (Kaj, Zenterio)

The main explanation that Kaj gave for keeping some things which are considered as core in-house is ‘to get the product hold together’. The other aspect that Fredrik also put as determinant factor for outsourcing was when the customer requests are many and related to application or functionality of the product?

“If the customer brings us a big bag, we want this, this and this and I look at this, and see there is a big junk of functionality, we don’t have this at all. We can give this to a third party company.” (Fredrik, Zenterio)

Olle, scrum team member, sees outsourcing from a different angle; in his opinion to our question of - having more tasks be outsourced versus in-housed? He pointed out technical issues as a factor that may create with outsourcing; for example in cases like adding to or continuing a code that has been developed by third parties creates difficulty; thus, he preferred to in-house over outsource- here is how he put it in his own words –

“I think it’s yeah [smiles]. No I think its good do things yourself, then you know what you’ve done. It’s easier to follow up, if something is missing, it’s easier just to continue and add those stuff otherwise you need to get into someone else’s code, of course it’s the same if it is internal, but at least you can go and ask that person who wrote the code.” (Olle, Zenterio)

We observed that employees have the desire to keep the entire development process under their own periphery and such tendency as portrayed by Herbsleb and Moitra (2001) emanate from individuals threat of job loss, control loss, fear of relocation and extensive travel.

5.5.2. Complete Team along with a Project Manger

It is also advised by almost all interviewee to put the project team to co-locate with their leader. In Ericsson Linda had an experience of working on a development project that has partly run in china and she believes that every team needs to be co-located and working with the same backlog and sprint; putting one or few member in a different

country creates a problem. Distance damage the feeling of teamness; hence, Linda strongly recommended that in thinking of working in a distributed development projects, at least complete team should be sitting together in the same place.

Fredrik in his response to our question of; what would happen, if one of the individual from team is located in different office or town? - He replied as follows:

“The problem would be that when you would really have to give a person a job that he could do in parallel with us... A lot of the time, the cooperation and being able to work in pairs, being able to say “Well now I am stuck, I am going to talk to Lars about this”, or “Uhh, I found a problem I am going to tell him about this”. All that stuff, sort of, vanishes. Ideally you could pick up a phone and call him, but that’s a much much much harder threshold than just walking over to him and telling “I found a problem, let me show you”. So, Skype and phones and emails, are great, but they are much easier to ignore than a person knocking on your door basically.” (Fredrik, Zenterio)

In agile development it is the so-called ‘soft skills’ which are critical denominator; the ‘soft skills’ which involve creating and maintain team relationships; better interaction; demonstrate flexibility, adaptability and being a team player (Chin, 2003). Hence such ‘soft skills’ require the team to be close to each other, because limited occasions of face-to-face communication and lack of common experience prone significant challenge such as on trust among team members, thus reducing the willingness of team members to communicate and collaborate (Pyysiäinen, 2003). Denis also highlighted the relevance of co-location for speedy decision making and implementation of projects and the need to consider such distances in outsourcing scenarios:

“All decisions can be implemented faster if you are present at the place, an employee can turn off Skype or telephone – you can’t reach the person. If you are at the place you can effect a person in different means, even not the nicest ones like physiological or bullying. If you can’t reach a person, there is nothing you can do. These risks should be taken into account and allocate more time for implementation.” (Denis, GGA)

In the ideal scenario, communication is to be best when it is on its basic prototype form- ‘face-to-face’ (Clark, 1996; and Carmel, 1999). It is evident that using those technology-mediated communication tools over temporal distance creates significant delays in communication (Ågerfalk et al., 2005). Kaj had a very strong stand on teams being very close to the project manager; Let alone putting them in a different location he argued that team should be together all the time -

“We have these teams that put together and they are put together quite hard so it is difficult for somebody to take someone out of the team or out of the project.” (Kaj, Zenterio)

Denis also argued in a similar way that, when development is in an outsourced set-up a single manager located in head office cannot possibly handle all the teams in the development process; therefore team leaders must be assigned for each team to facilitate the different activities spread across different countries. One of the problems that Denis mentioned was:

“... communication issues – to make proper communication between different teams is the hardest thing. Software developers they are all technical people, these kind of people have problems speaking the same language and to find some compromises, insist, co-ordinate something that it would work. They don't know how to communicate.” (Denis, GGA)

According to Denis the sever communication problem that distributed team face can only be solved if a good team leader is appointed for each team to have a matrix network among themselves. All the three cases under study respondents seem to be familiar with how teams should work in a distributed set-up; most importantly all opinions were consistently showing the need to put each team with their leader together to create and sustain the teamness and strong bond among members.

5.5.3. Minimizing Interdependence among Teams

There is always a coordination problem when two individuals have a common goal to achieve and when the action of one depends on the action of the other (Clark, 1996). Software development due to its complex and less routine work nature makes coordination to be difficult and expensive (Espinosa & Carmel, 2004). Coordination problem amplify when the software development involves distributed team (Ågerfalk et al., 2005) Denis emphasized the significance of devising a way to minimize task interdependence among teams and activities in the development process, he said that:

“Parts should be independent, as much as possible, it's true. And also with good specification. You should also carefully select the team, it's very much depends on the team.” (Denis, GGA)

Distributed team members of projects create a barrier in quantity and quality of communication (Allen et al., 1980; Jain & Triandis, 1990), making the integration and coordination effort more difficult (Keller, 1994; and Meyer, 1993). Hence, as reflected in all the interviews, while dividing assignments among the different teams, whether outsourced or distributed team in a different floor, due attention must be given to split it in manner that minimize once dependent over the other. Documentation happened to be more important due to the limited options in which developers and customers have to interact, thus the agile assumption of on minimizing documentation becomes a challenge (Turk et al., 2002). Denis in a similar manner argued that activities the need to be specified and planed beforehand; which consequently force to deviate the pure form of agile development mechanism and adapting to specific customized form of development.

5.5.4. Availability of Ample Time before Delivering Product to Customers

Agile projects are highly praised by customers because they allow the customers to have constant control over development processes (Dybå & Dingsøy, 2008). Kaj also stated in the same manner as:

“Yes, most of them [customers] are outside Sweden. Most of the meetings are phone meetings, sometimes the customers come here to our premises and sometimes we go to the customer and perform workshops and customer meetings. I would say this is quite important that you have a good relation and you are getting to know customer project managers and so on you have some kind of relationship together.” (Kaj, Zenterio)

Moore et al. (2004) also stated that outsourcing and agile developments respond better to customer requirement at a lower cost. However, in perusing outsourcing activity, Fredrik pointed out the precautions that must be made; for our question of ‘an opinion on outsourcing software development’, Fredrik replied in his own word as follows:

“Oh boy, more or less, emm... I have no problem with outsourcing jobs, outsourcing work to other companies. I think, when it comes to work that is directly customer related, I don’t want to do that, because if the customer going to call me, and say that we found a problem this morning, I don’t know if that is something we did or you did. I wanna be able to answer that of in a span of 45 minutes, to be able to say I will go and check it with someone, I will be right back. And if I then have to start a process with a third party company and take a response time of 3 days or a week. To get a simple question answer, I don’t think that would look good ... Fast response times, are probably thing more than anything else that customer value, to be able to get a quick answer to their question.” (Fredrik, Zenterio)

Even though it is difficult to distinguish what activities are directly related to customers and which are not, we can sense that the things which require fast response from customers must be handled in-house, and not outsourced. In addition, time to deliver the final product seems to affect the decision of outsourcing or in-housing. In agile software development responding to changes creates more value to customers and relevant in delivering quality software (Abrahamsson et al., 2002).

In order to create the value to customers the interviewees’ highlighted product quality as a major factor; and quality in outsourced set-up is improved when the outsourced part is delivered to them some time before final delivery, because such time will allow them to have a look at it before forwarding it to customers.

“... they [the outsourced parts] going to be done way before customer starts using and has questions – that will be fine. Because then by the time, we’ve got their delivery, there is some time down here, we have ownership of it,

and a customer calls we have a problem, we can come back with quick response time.” (Fredrik, Zenterio)

Therefore, before deciding to outsource a particular part of the software development process a special consideration must be given to factors related with relationship of the outsourced part with customer (if fast response to customer is likely); and if there is enough time to get the outsourced part and check before handing it over to customers.

6. Conclusion

The current language of software development industry is Agile; it is has become evident that in today's volatile business environment software development firms in order to survive the stiff competition, they need to be flexible and capable of embracing changes. In addition to that, the software development trend is showing that outsourcing partly or the whole process has become a common practice. The fact that, outsourcing became a common practice and the popularity of Agile methods in the software development projects has created a curiosity to research the area.

The findings of this research can be categorized into four main areas as follows: Drivers for Agile development; Drivers for Outsourcing; Contradiction between Outsourcing and Agile Methods; and Preconditions for Outsourcing Agile Methods. Even though our main study focuses on the last two areas, the need to have a clear understanding for the drivers of both outsourcing and Agile methods help in better elaborating our study.

Firstly, research was able to uncover four main drivers for organizations to turn into Agile method in software development, these drivers are: (1) Popularity, trend; (2) Engaging employees, fun and socialization; (3) Mirroring big customers and (4) Pressures from the market for short time product delivery. Having said this about the reasons for using Agile methods in software development projects, outsourcing also seem to be another aspect of the development, thus leading us to the second finding- The findings of this study also shows that firms go for outsourcing due to reasons such as: (1) the increasing number of customers which may not be possible to serve by only using internal capacity; (2) search for specialized competence from external parties as firms rarely are self-sufficient; and (3) the nature of their operation, meaning, it is natural for firms that are involved in an international trade to be involved in outsourced set-up, because the customers as well as resources are obviously spread all over the globe.

A discussion on drivers and reasons to combine both Agile method and Outsourcing resonates with benefits of 'red triangle' in the developed framework in the form of matrix (see Figure 20). This model shows reader the complexity and 'fuzziness' of firms' activities in area of Agile projects and Outsourcing. Moreover, it is very hard to be good at both practices for even strong market players. Hence, the notion of trade-offs between outsourcing and other activities (Berggren et. al., 2011) seem to be valid and apply to researched cases. Despite the fact, that model used 'study depth' for emphasizing size of bubble in the model, the real size of company matters as well – we can see that the larger the firm and the more it is involved in international trade, the higher it is placed on outsourcing axis. Moreover, Ericsson AB, as by far the largest firm among case firms, is much closer to 'red triangle' and it's benefits. We can see that such position is not accidental, and is shaped by management deliberately using

best of outsourcing opportunities – low cost manufacturing, market proximity and search for additional competence.

Even though the above reasons for being Agile in projects and outsourcing seem logical in a separate sense, combining both in the development projects holds different challenges. Software development projects in their own involve complex processes and outsourcing such type of complex processes brings several challenges. The three implications of distributed development identified by Ågerfalk et al. (2005), geographical distance, socio-cultural distance and temporal distance, also once again confirm to the challenges (communication, control and coordination) mentioned by all cases in our study except one where temporal distance seem to enable the project ‘round-the-clock’ development. Hence, the study shows that even though outsourcing and distributed software development seems to be inevitable and very important, there are inherent challenges associated with the specific practice.

The third area of the finding shows that, case firms’ projects preferred close team allocation, short communication distance and face-to-face interactions other the time-to-market speed and round-the-clock development possibilities offered by Outsourcing. Through talking to employees and observing Scrum meeting in one case, we can conclude that values proposed by Agile Manifesto in year 2001 were applied in the investigated projects. That means that ‘Individuals and interactions’, ‘Customer collaboration’ and ‘Responding to change’ were highly valued. On the theoretical analysis, this contradicts to the typical practices of designing Outsourcing process with clear contract, determined scope and duration, planned processes and tools. The study identified six most valued practices and this included: (1) stand-up meetings; (2) Burn-down charts and visualization; (3) Constant contact with customer; (4) Co-located teams; (5) Knowledge sharing and focus and (6) Fun and Developer involvement. In general, analysis of these practices showed that they contradict with Outsourcing to a big extent. Even more, they serve as an opposite mean of reaching productivity, which can be also reached by Outsourcing. Hence, we can conclude that these six practices for project teams act as a lock-in against implementing Outsourcing widely. As a result, the rhetoric question of “Is it possible to stay within red triangle?” can be raised.

Finally, the study emphasized the issue of what could be done in order to actually be able to involve Outsourcing the Agile set-up projects. The study illuminated the factors that facilitate to successfully outsource Agile projects. Hence, the criterions that participants put to involve in outsourcing agile developments are:

- The part which is going to be outsourced, must not be related to core product
- When outsourcing or involving in a distributed development a team should be co-located with its leader
- Minimize interdependence among distributed teams
- Outsource the part which gives enough time before delivering product to customers

Of course the above list are not the only criterion, but the respondents gave much emphasis to the above and may serve as a guide line while considering outsourcing in agile software development.

Apart from above four areas of main findings, we found couple of other fascinating observations. These observations are: the fun part of Agile set-up is important, it results in employees happiness, interactions and motivation; differences in managerial styles of project execution exist, there is a need for supportive and collaborative management in order to make project team committed to assigned tasks; role of project managers in Agile set-up is more complicated compared to traditional project management, it includes much more work in mediating between client and project team; and we could see that daily stand-up meetings are very important for these types of projects for multiple reasons as motivation, socializing, learning, knowledge transfer, debriefing, control and communication.

To sum up, the research was able to successfully investigate on the gap in the area of Agile set-up with Outsourcing. The overall findings present considerable number of contradictions in between the two issues (Outsourcing and Agile Method); however some pre-conditions have been identified while implementing outsourcing in Agile software development in order to increase success chance. Even though, the findings on both research questions are valid for our case firms, we would be careful to talk about generalization of these findings.

Closing Comments

This final part of paper will include authors suggested implications for practitioners, academics and future research; present overall limitations upon whole study; and finally provide a reader with less formal personal reflections.

Implications

Taking into account all the findings, it would be tempting to give conclusions and suggestions in the famous framework of “lessons learned” by E. G. Guba. However, we believe that findings could rather be transformed into a set of implications. First, based on our findings we would like to provide **implications for practitioners**.

As a most significant recommendation directly linked with second research question, we would suggest those practitioners who think about outsourcing their Agile projects to take considerations on four identified pre-conditions. In practice then designing the set-up of such outsourced projects outsourced part rather not be close related to core activity; the team should always be co-located and accompanied by experienced team leader; interdependences between distributed teams should be minimized; and proper amount of time should be allocated between delivery from outsourced team and delivery to the final customer.

In addition to that, developed 2x2 matrix framework can serve as a tool to illustrate company’s position on the field of outsourcing/agile dimension and help managers to design strategy of project work or serve as guideline for addressing the issues of working in distributed set-up. Moreover, we could clearly see that Agile enables face-to-face, interactions and co-location, which results in higher motivation, happiness, fun, team work, trust and collaboration – that is why managers should be careful on their decisions while considering outsourcing. However, it is up to those managers to decided on the trade-offs between both possibilities and as seen from case firms, different set-ups are possible and work in practice.

Then turning to the **implications for academics**, we consider that current research was able to improve understanding of Agile methods and Outsourcing nature; discuss on their contradictions and fill the identified gap in academic literature. Based on that achievement, the new path of exploring possibilities in outsourcing Agile projects was opened, since to our considerations it was not yet established. Authors of this paper brought new knowledge on what factors should be considered while thinking about outsourcing agile projects successfully. Furthermore, developed framework can be re-used for understanding and analyzing future research; it can be tested with different criteria for agile/outsourcing dimensions, hence other theories than those used in this research can be tested on this model.

The research on Agile software development that involves outsourcing still offers many **possibilities for future studies**. One avenue that opens up for future research is on

‘how to combine the good practices of Agile software development with the classical project management?’ in order to implement them in other project types. Another path is to research on ‘how to fulfill those four identified pre-conditions?’ - Such research could include cross study between business administration and IT field, since technological operational factors are important. Next, the notion of trade-offs (Berggren et. al., 2011) can be future tested with other variables, since level of Agile method is only one of possible trade-offs. Finally, the issue of ‘how to integrate distributed knowledge, which resides in outsourced agile projects?’ could be researched.

Limitations

Even though, as mentioned in previous chapters, a broad list of relative literature and good number of empirical data was gathered to make research findings valid, current and meaningful, there are some limitations to take into account before ending this study.

Firstly, the authors of this research come from business administration area, currently focusing on contemporary strategy, management, leadership and project management. The technological part of studied projects, as for example code writing, testing or software products, which is outside the scope of research, thus limiting technical knowledge in the area of study. As Mohammed et al. (2008) suggests ‘operational factors’ can be seen as barriers and enablers for outsourcing, meaning that if in future research would be steered towards these technical factors, some additional important findings are likely to be achieved. This might be important particularly to practitioners in the area of Agile set-up projects.

Secondly, several process limitations have to be considered before drawing any generalizations of this study. Authors recognize the fact that different amount of time for each separate case, in term of empirical data gathering, was spent. This results in different ‘thickness’ of study into each case. Hence, the possibility of comparing cases (what was not the aim of study, see Methodology chapter) within developed framework in form of 2x2 matrix, is limited. In addition, the process of data gathering was based on iterative semi-structured interviews, meaning that some questions emerged or were deliberately abandoned during the study. If talking about observation method – it was used only once and in one case, in that respect it should be seen as additional source of data and opportunity to take insider perspective in the study. Therefore, as a ‘stand alone’ method it is not credible enough for relevant findings in our research and should not be taken apart from other methods used.

Thirdly, the obvious limitation of qualitative data has to be considered in this study as well. Although the study has quite valid and reliable base, the findings are limited to the specific case under study and cannot be generalized, but should rather be interpreted and re-used if necessary.

Personal Reflections

In this part of the paper, before closing the study, authors would like provide a reader with personal reflections in a less structured/academic way. Working on each chapter was different and required diverse skills. Co-authoring also required some adjustment to each other work habits and style. But in the end, the paper is higher quality due to co-authoring and benefits a reader with number of methods and wider look on the issue. After all, it is a pleasant feeling to see how chapters are linked, support each other. Having an option to look retrospectively, we would make contacts with industry sooner and narrow down the literature list. However, still believe that as authors we did a good work, learned a lot on teamwork and qualitative research, contrary to quantitative research used in Bachelor degree studies.

If turning to the particular issue under investigation – Agile methods and Outsourcing practice, even though a multiple case study method was selected, authors often felt themselves like being in grounded theory area. It was harder to formulate research area properly, than in fact investigate it; literature review ‘neither provided key concepts, nor suggested hypothesis’, instead showed gaps in existing knowledge as rationale for research; and research questions were broad and changing during data collection and analysis (May (1986) c/f Creswell, 2007).

On the practical level, we could observe some key contradictions between Agile projects and Outsourcing. At the same time, improvement and project adjustment through the identified pre-conditions would easy the path to successful Agile and outsourced projects. It was peculiar and exciting to face some issues which were not in the research area from the beginning: the importance of stand-up meetings and their social/motivational aspect; role of leadership style in the Agile projects; and fuzzy design of Agile projects, result in the fact that none of case firms were completely Agile method project firm. Additionally, we see that industry is not static and needs to be analyzed, supported with an academic research. That is why, we hope that developed model and identified contradictions with pre-conditions is a contribution to academic research, could be re-used or further developed in order to have an effect on real industry actions.

At this point we would like to thank readers of this paper and announce research finished. Lastly, authors would like to welcome readers for any constructive criticism, which can be sent directly to us:

Maksim Buslovič: maksim.buslovic@gmail.com

Samson Deribe: samson_deribe@yahoo.com

List of References

- Abrahamsson, E. (2000), Change without Pain, Harvard Business Review, July-August, pp. 75-79
- Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. Proceedings - International Conference on Software Engineering , pp. 244-254
- Ågerfalk, P. J. (2004). Investigating Actability Dimensions: A Language/Action Perspective on Criteria for Information Systems Evaluation, *Interacting with Computers*, Vol. 16, No. 5, pp. 957–988.
- Ågerfalk, P. J.; Fitzgerald, B.; Holmstrom, H.; Lings, B.; Lundell, B. & Conchuir, E. Ó. (2005). A framework for considering opportunities and threats in distributed software development. Proc. International Workshop on Distributed Software Development, Paris, France: Austrian Computer Society, pp. 47-61.
- AgileAlliance (2001). Manifesto for Agile Software Development.
- Allen, T. J., Lee, D. M. S. & Tushman, M. L (1980). R&D performance as a function of internal communication, project management, and the nature of the work. *IEEE Transactions on Engineering Management* 27:2–12.
- Anderson, D.J., (2004). *Agile Management for Software Engineering*. Prentice-Hall, Upper Saddle River, NJ, USA.
- Bartelt, C., Broy, M., Herrmann, C., Knauss, E., Kuhrmann, M., Rausch, A., Rumpe, B. & Schneider, K. (2009). Orchestration of global software engineering projects - Position paper Proceedings - 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009, art. no. 5196956 , pp. 332-337.
- Barthelemy J. (2001). The Hidden Costs of IT Outsourcing. *MIT Sloan Management Review* 42 (3) , pp. 60-69
- Battin, R.D., Crocker, R., Kreidler, J. & Subramanian, K. (2001) Leveraging resources in global software development, *IEEE Software*, Vol. 18, No. 2, pp. 70-77.
- Beck, K. (1999a). Embracing Change with Extreme Programming. *IEEE, Computer* 32(10): 70-77.
- Beck, K. (1999b). *Extreme Programming Explained: Embrace Change*. Reading, Mass., Addison-Wesley.
- Berggren, C., Bergek, A., Bengtsson, L., Hobday, M., & Söderlund, J. (2011). *Knowledge Integration and Innovation: Critical Challenges Facing International Technology-based Firms*. New York, United States: Oxford University Press Inc.
- Blumberg, B., Cooper, D. R., & Schindler, P. S. (2005). *Business research methods*. Berkshire, UK: McGraw-Hill Education.
- Blumberg, B; Cooper, D.R & Schindler, P.S. (2011). *Business Research Methods*. McGraw-Hill Education Inc. 3rd edition
- Boehm B. (2002), Get Ready for Agile Methods, with Care. *Computer*, 35 (1): 64-69.

- Boehm, B. & Turner, R. (2004). *Balancing Agility and Discipline-A Guide for the Perplexed*. Addison-Wesley, Boston, MA, USA
- Braun, A., Dutoit, A.H. & Brugge, B. (2003) A Software Architecture for Knowledge Acquisition and Retrieval for Global Distributed Teams, In International Workshop on Global Software Development, (collocated with ICSE 2003), pp. 24-29.
- Brinklow, T. (2006). *Agility*. Brighton Business School.University of Brighton. Occasional / Working Paper Series. Copyright© Tony Brinklow. ISSN: 1 905593-15-5.
- Brudenall, Peter [Editor]. (2005). *Technology and Offshore Outsourcing Strategies*. Hampshire, New York, N.Y., USA: PALGRAVE MACMILLAN.
- Bryman, A., & Bell, E. 2007. *Business research methods*. 2nd Edition. Oxford university press, U.K.
- Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*. Prentice-Hall, Upper Saddle River.
- Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE Software*. Volume 18, Issue 2, Pages 22-29.
- Casey, V. and Richardson, I. (2004) Practical Experience of Virtual Team Software Development, In European Software Process Improvement (EUROSPI) 2004, Trondheim, Norway.
- Charette, R.N. (2003). Challenging the fundamental notions of software development. Cutter Consort. Exe. Rep. 4 (6).
- Chen, G. (2004), *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*, Saranac Lake, NY, U.S. AMACOM Published on, eISBN : 9780814427361, Project Management.
- Chesbrough, H. (2006). *Open Business Models: How to thrive in the new innovation landscape*. Boston, Massachusetts, USA: Harvard Business School Press.
- Chin, G. (2003). *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. Saranac Lake, NY, USA: AMACOM Books.
- Clark, H.H. (1996) *Using Language*, Cambridge University Press, Cambridge.
- Cockburn, A. (2001). *Crystal Clear: A human-powered software development methodology for small teams*. Reading, MA, Addison-Wesley.
- Cockburn, A. (2002a). *Agile Software Development*. Boston, Addison-Wesley.
- Cockburn, A., & Highsmith, J. (2001). Agile Software Development: The People Factor. *IEEE Computer*, 34(11), 131–133.
- Cockburn, A., (2007). *Agile software development: the cooperative game*. Addison Wesley.
- Cohen, D., Lindvall, M. & Costa, P. (2004). An Introduction to Agile Methods. *ADVANCES IN COMPUTERS*, VOL. 62. Copyright © 2004 Elsevier Inc.
- Cohen, S. & Mankin, D. (1999), Collaboration in the virtual organization, *Trends in organizational Behaviour*, 6, 105-120.

- Cohn, M. & Schwaber, K. (2003). The Need for Agile Project Management, Agile Times, vol. 1.
- Conboy, K. & Fitzgerald, B. (2004). Toward a conceptual framework for agile methods: a study of agility in different disciplines. WISER 2004 - ACM Workshop on Interdisciplinary Software Engineering Research, pp. 37-44
- Conboy, K. (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. Information Systems Research 20 (3), pp. 329-354
- Conchuir, E.O., Holmstrom, H., Agerfalk, P.J. & Fitzgerald, B. (2006). Exploring the Assumed Benefits of Global Software Development. IEEE International Conference on Global Software Engineering (ICGSE'06), pp 159 – 168
- Cooke, J. L., (2010). Agile Principles Unleashed: Proven Approaches for Achieving Real Productivity in Any Organisation. Ely, U.K.: IT Governance Pub
- Creswell, J. W. (1998). Qualitative inquiry and research design: Choosing among five traditions. Thousand Oaks, CA: Sage.
- Creswell, J. W. (2007). *Qualitative Inquiry & Research Desing*. Thousand Oaks, California, USA: Sage Publications, Inc.
- Damian, D.E. & Zowghi, D. (2002) The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, In Proceedings IEEE Joint International Conference on Requirements Engineering, IEEE Computer Society, Los Alamitos, pp. 319-328.
- DeVor, R., Graves, R. & Mills, J.J., (1997). Agile Manufacturing research: accomplishments and opportunities. IIE Transaction, 29, pp. 813 – 823.
- Dhar S. & Balakrishnan B (2006), Risks, Benefits, and Challenges in Global IT Outsourcing: Perspectives and Practices, Journal of Global Information Management, vol. 14, issue 3, Idea Group Inc.
- Dybå, T. & Dingsøyr, T. (2008), Empirical studies of agile software development: A systematic review, Inform. Softw. Technol. (2008), doi:10.1016/j.infsof.2008.01.006
- Eisenhardt, K.M. (1989). Building Theories from Case Study Research. Academy of Management Review, 14(14). PP 532-550.
- Enberg, C., Lindkvist, L. & Tell, F. (2010). Knowledge integration at the edge of technology: On teamwork and complexity in new turbine development. International Journal of Project Management 28 (8), pp. 756-765
- Erickson, J., Lyytinen, K. & Siau, A. (2005). Agile software development and extreme programming: the state of research. Journal of Database Management, Vol. 16 No. 4, pp. 88-100.
- Espinosa, J. A. & Carmel, E. (2004) The Effect of Time Separation on Coordination Costs in Global Software Teams: A Dyad Model, In Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1, IEEE Computer Society, pp. 1-10.

- Festinger, L., Schachter, S. & Back, S (1950). *Social Pressures in Informal Groups: A Study of Human Factors in Housing*. Palo Alto: Stanford University Press.
- Florida, R. (2005). The World is Spiky, *The Atlantic Monthly*, October: 48-51.
- Friedman, T.L. (2005). The World is Flat, After All, *New York Times*, April 3 2005.
- Ghauri, P. & Grønhaug, K. (2005). *Research Methods in Business Studies: Practical Guide*. Pearson Ed. 3rd edition.
- Gilley, K. M. & Rasheed, A. (2000), Making More by Doing Less: An Analysis of Outsourcing and its Effects on Firm Performance. *Journal of Management*, Vol. 26, No. 4, 763–790
- Glass R. (2001). Agile versus traditional: Make love, not war, *Cutter IT Journal*, 12–18.
- Grant, R. (2010). *Contemporary Strategy Analysis*. 7th ed. Chinchester: JohnWiley & Sons, Ltd.
- Grover V., Cheon M. J. & Teng J.T.C. (1996), The Effect of Service Quality and Partnership on the Outsourcing of Information Systems Functions. *Journal of Management Information Systems* Vol. 12, No. 4, pp. 89-116
- Gunasekaran, A., (2001). *Agile manufacturing: The 21st century competitive strategy*. 1st ed. Oxford; New York: Elsevier
- Halvey, M.J. & Keane, M.T. (2007). An assessment of tag presentation techniques. *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*, pages 1313–1314
- Hanssen, G.K. & Faegri, T.E. (2007). Process fusion: An industrial case study on agile software product line engineering. *The Journal of Systems and Software* 81, pp 843-854.
- Herbsleb, J. D. & Grinter, R. E. (1999). Architectures, Coordination, and Distance: Conway's Law and Beyond. *1999 IEEE Software* 16 (5), pp. 63-70
- Herbsleb, J. D. & Mockus, A. (2003) An Empirical Study of Speed and Communication in Globally Distributed Software Development, *IEEE Transactions on Software Engineering*, Vol. 29, No. 6, pp. 481-494.
- Herbsleb, J. D. & Moitra, D. (2001). Global software development. *IEEE Software* Volume 18, Issue 2, Pages 16-20
- Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R.E. (2001). An Empirical Study of Global Software Development: Distance and Speed. *Proceedings - International Conference on Software Engineering 2001*, Pages 81-90.
- Highsmith J. (2002). *Agile Software Development Ecosystems*, Addison–Wesley, Boston, MA.
- Highsmith, J. (2004). *Agile Project Management: Creating Innovative Products*. Addison-Wesley/ Pearson, Boston, MA, USA.
- Hoegl, M. & Proserpio, L. (2004). Team member proximity and teamwork in innovative projects, www.elsevier.com/locate/econbase, Elsevier B.V. All rights reserved

- Hossain, E., Bannerman, P.L. & Jeffery, D.R. (2011). Scrum practices in global software development: A research framework. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6759 LNCS , pp. 88-102
- Jain, R. K. & Triandis, H. C (1990). *Management of Research and Development Organizations*. New York: John Wiley and Sons.
- Karolak D.W. (1998), *Global software development: managing virtual teams and environments*. IEEE Computer Society Press, Los Alamitos, Calif.
- Katz, R. & Tushman, M. L (1979). Communication patterns, project performance, and task characteristics: An empirical evaluation and integration in an R&D setting. *Organizational Behavior and Human Performance* 23:139–162.
- Kedia B. L. & Lahiri S. (2007), International outsourcing of services: A partnership model. *Journal of International Management*, 13 (1), 22 – 37
- Keil, M., & Mähring, M. (2011). Is your project turning into a black hole? *California Management Review* 53 (1) , 6-31.
- Keller, R. T (1994). Technology-information processing fit and the performance of R&D project groups: A test of contingency theory. *Academy of Management Journal* 37:167–179.
- Kessler.E.H & Alok K. Chakrabarti (1999), *Speeding Up the Pace of New Product Development*. Elsevier Science Inc.
- Kiel, L. (2003) *Experiences in Distributed Development: A Case Study*, In *International Workshop on Global Software Development: GSD 2003*, (co-located with ICSE 2003), pp. 44-47.
- Krishna, S., Sahay, S. & Walsham, G. (2004) *Managing Cross-Cultural Issues in Global Software Outsourcing*, *Communications of the ACM*, Vol. 47, No. 4, pp. 62-66.
- Lee, J.N., Huynh, M.Q., Chi-wai, K.R & Pi, S.M. (2000). The evolution of outsourcing research: what is the next issue? In: *Proceedings of the 33rd Hawaii International Conference on Systems Sciences*, Hawaii, pp 1–10.
- Lenfle, S., & Loch, C. (2011). Lost roots: How project management came to emphasize control over flexibility and novelty. *California Management Review* 53 (1) , pp. 32-55.
- Lientz, B. P., & Rea, K. P. (1999). *Breakthrough technology project management*. San Diego, California, USA: Academic Press.
- Lindvall, M., Basili, V., Boehm, B, Costa, P. & Dangle, K. (2002). Empirical Findings in Agile Methods. *Lecture Notes in Computer Science*, 2002, Volume 2418, *Extreme Programming and Agile Methods — XP/Agile Universe 2002*, Pages 81-92.
- Lundin, R.A. & Söderholm, A. (1995) *A theory of the temporary organization*. *Scandinavian journal of management* [0956-5221] Lundin, R A år:1995 vol

- Malone, T., Laubacher, R., & Johns, T. (2011). The big idea: The age of hyperspecialization. *Harvard Business Review* 89 (7-8) , 58-65.
- Marlon, L., Daniel, G. & Mauro, T. (2009). Challenges on Adopting Scrum for Distributed Teams in Home Office Environments. *World Academy of Science, Engineering and Technology* 59.
- Martin C. R. & Martin M. (2006). *Agile Principles, Patterns and Practices in C#*. Prentice Hall, Pearson Education, Inc.
- Martino, R. L. (1964). *Project management and control. Vol. 1, Finding the critical path and Vol. 2, Applied Operational Planning*. New York, USA: American Management Association.
- Maylor, H., Brady, T., Cooke-Davies, T., & Hodgson., D. (2006). From projectification to programmification. *International Journal of Project Management* , Volume 24, Issue 8, Pages 663-674.
- Melnik, G. & Maurer, F. (2002), Perceptions of agile practices: a student survey, in: *Proceedings, extreme Programming/ Agile Universe 2002, Lecture Notes in Computer Science*, vol. 2418, 2002, pp. 241–250.
- Meredith, J. R., & Mantel, S. J. (2010). *Project Management: A Managerial Approach, 7th Edition*. John Wiley & Sons (Asia) Pte Ltd.
- Meso, P. & Jain, R. (2006). Agile software development: Adaptive systems principles and rest practices. *Information Systems Management* 23 (3), pp. 19-30
- Meyer, C. (1993). *Fast Cycle Time: How to Align Purpose, Strategy, and Structure for Speed*. New York: Free Press, 1993.
- Meyerson, D.E. (2001), Radical change the quiet way, *Harvard Business Review*, No.October, pp. 92-100.
- Midler, C. (1995). “Projectification” of the firm: The renault case. *Scandinavian Journal of Management* , Volume 11, Issue 4, 363–375.
- Miles,M.B.,& Huberman, A.M. (1994). *An Expanded Sourcebook, Qualitative Data Analysis*. London, Sage.
- Mohammed I. R., Shankar R. & Banwet D. K. (2008), Creating flex-lean-agile value chain by outsourcing. An ISM-based interventional roadmap. *Business Process Management Journal* Vol. 14 No. 3, pp. 338-389
- Mol, M. J. (2007). *Outsourcing: Design, Process, and Performance*. Cambridge, UK: Cambridge University Press.
- Moore, S., Barnet L. & Dowling K.Q, (2004). *Offshore Outsourcing and Agile Development: Improving Customer Satisfaction, Not just Cost Savings*, @2004, Forester Research, Inc.
- Moreira, M.E. (2010). Adapting configuration management for agile teams [electronic resource]: balancing sustainability and speed. John Wiley & Sons,
- Nerur,S., Mahapatra, R.. & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, Vol. 48, No. 5, PP 72– 78.

- O'Reilly III, C.A., Harreld, J.B. & Tushman, M.L. (2009), *Organizational Ambidexterity: IBM and Emerging Business Opportunities*, California Management Review, Vol. 51, No. 4, pp. 75-99.
- Palmer, S.R. & Felsing, J.M. (2002). *A Practical Guide to Feature-Driven Development*. Upper Saddle River, NJ, Prentice-Hall.
- PMBOK-Guide. (2004). *A Guide to the Project Management Body of Knowledge*, Third Edition. Newtown Square, Pennsylvania, USA: Project Management Institute, Inc.
- Pope, C., Ziebland, S. & Mays, N. (2000). Analysing qualitative data. *British Medical Journal*. Volume 320, Issue 7227, 8 January 2000, Pages 114-116
- Poppendieck, M. (2001). Lean programming. *Software Development Magazine* 9 71–75.
- Prikladnicki, R., Audy, J. L.N. & Evaristo, R. (2004). *Global Software Development in Practice Lessons Learned*. John Wiley & Sons, Ltd. *Softw. Process Improve. Pract.* 2003; 8: 267–281 (DOI: 10.1002/spip.188).
- Qumer, A. & Henderson-Sellers, B. (2006). Comparative evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT), in Irani, Z., Sarikas, O.D., Llopis, J., Gonzalez, R. And Gasco, J. (Eds), *Proceedings of the European and Mediterranean Conference on Information Systems (EMCIS2006) CD*, Brunel University, West London.
- Ribeiro, F. L. & Fernandes, M. T (2009). Exploring agile methods in construction small and medium enterprises: a case study. *Journal of Enterprise Information Management*. Vol. 23 No. 2, 2010. pp. 161-180. Emerald Group Publishing Limited.
- Sahay, S. (2003). Global software alliances: the challenge of “standardization.” *Scandinavian Journal of Information Systems*, Vol. 15, pp. 3–21.
- Schwaber K. & Beedle M. (2002). *Agile Software Development with SCRUM*, Prentice Hall.
- Schwaber, K. (1995). *Scrum Development Process*. OOPSLA'95 Workshop on Business Object Design and Implementation. Springer-Verlag.
- Sengupta, B., Chandra, S. & Sinha, V., (2006). *A Research Agenda for Distributed Software Development*, Proceedings of the 28th ICSE, Shanghai.
- Sharp J.M., Hides M.T. & Bamber C.J. (2000), *Continuous Organizational Learning through the Development of High Performance Teams*, HPO Research Group, Business School, University of Oxford, UK.
- Sidhu, J.S. & Volberda, H.W. (2011). Coordination of globally distributed teams: A co-evolution perspective on offshoring. *International Business Review* 20 (3), pp. 278-290
- Smith, J.K. (1983). Quantitative Versus Qualitative Research: An Attempt to Clarify the Issue. *Educational Researcher*, 12(3), 6-13.
- Somekh, B., & Lewin, C. (2005). *Research Methods in the Social Sciences*. London, UK: SAGE Publications Ltd.

- Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: Sage
- Strode, D.E., Huff, S.L., Hope, B. & Link, S. (2012). Coordination in co-located agile software development projects. *J. Syst. Software* (2012), doi:10.1016/j.jss
- Takeuchi, H. & Nonaka, I. (1986). The New Product Development Game. *Harvard Business Review*, Jan. /Feb.: 137-146.
- Taxen, Lars [Editor]. (2011). *The System Anatomy: Enabling Agile Project Management*. Lund, Sweden: Studentlitteratur AB.
- Tonnquist, B. (2008). *Project Management*. Stockholm: Bonnier Utbildning AB.
- Turk, D., France, R. & Rumpe, B. (2002). *Agile Software Processes: Principles, Assumptions and Limitations*. Technical Report. Colorado State University.
- Turner, J. R., & Simister, S. J. (2000). *Gower Handbook of Project Management, Third Edition*. Aldershot, Hampshire, England: Gower Publishing Limited.
- Van de Ven, A. H. & Sun, K. (2011), Breakdowns in Implementing Models of Organization Change, *Academy of Management Perspectives*, August, pp. 58-74.
- Van de Ven, A. H. & Poole, M. S. (1995), Explaining development and change in organizations, *Academy of Management Review*, Vol. 20, No. 3, pp. 510-540.
- Verma, Vijay K., (1997) *Managing the project team*, Includes bibliographical references and index, Contents: v. 3. *Managing the Project Team*, ISBN: 1-880410-40-0
- Whittington, R., Pettigrew, A., Peck, S., Fenton, E., & Conyon, M. (1999), Change and complementarities in the new competitive landscape: A European panel study, 1992–1996. *Organization Science* 10(5) (September–October):583–600.
- Williams, L. & Cockburn, A. (2003). Agile software development: it's about feedback and change. *Computer* 36, 39-43
- Williamson, Oliver E. (1985). *The Economic Institutions of Capitalism*. New York: Free Press.
- Wysocki, R. K., & McGary, R. (2003). *Effective Project Management: Traditional, Adaptive, Extreme, Third Edition*. Indianapolis, Indiana, USA: Wiley Publishing, Inc.
- Youker R. (1977), Organizational Alternatives for Project Managers, *Project Management Quarterly*. Vol. VIII, No.1,
- Zadeh, L.A. (1965). Fuzzy sets. *Information and Control* 8 (3): 338–353.

Websites Reference

- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). *Agile software development methods-review and analysis*. Extracted on March 10, 2012. <<http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>>
- Beck, K., Breedle, M., van Bennekum, A. Cockburn, A. Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for Agile Software Development. Extracted on February 15, 2012 from <<http://agilemanifesto.org/>>.
- Bloomberg Businessweek Company Overview of GGA Software Services LLC. Extracted on May 14, 2012 from <<http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=102325212>>
- Carr, N.C. (2003). IT doesn't matter. Harvard Business School Publishing Corporation. Extracted on February 10, 2012, from <<http://www.simoes.com.br/mba/material/ebusiness/ITDOESNTMATTER.pdf>>
- Dingsøy, T., Sridhar, S., Balijepally, V. G. & Moe, N. B. (2012). A Decade of Agile Methodologies: Towards Explaining Agile Software Development . Journal of Systems and Software, In Press, Accepted Manuscript, Available online 6 March 2012. Extracted on March 13, 2012, from <<http://www.sciencedirect.com/lt.ltag.bibl.liu.se/science/article/pii/S0164121212000532?v=s5>>
- Economic Intelligence Unit (2009) Organisational agility: How business can survive and thrive in turbulent times, Extracted on January 12, 2012, from <<http://www.emc.com/collateral/leadership/organisational-agility-230309.pdf>>
- Ericsson Annual report (2011). Extracted on May 14, 2012 from <http://www.ericsson.com/thecompany/investors/financial_reports/2011/annual11/>
- GGA Software Services LLC official web-page (2012). Extracted on May 14, 2012 from <<http://ggasoftware.com/>>
- Hayes, S. & Andrews, M. (2001). An Introduction to Agile Methods. Extracted on March 16, 2012 from <www.wrytradesman.com/articles/IntroToAgileMethods.pdf>.
- Ivček, M., & Galinac, T. (n.d.). *Adapting Agile practices in globally distributed large scale software development*. Retrieved 03 15, 2012, from www.ericsson.com: http://www.ericsson.com/hr/etk/dogadjanja/mipro_2009/21_1337_F.pdf
- McNaught C and Lam P. (2010) Using Wordle as a Supplementary Research Tool. The Qualitative Report Volume 15 Number 3 May 2010 630-643. Extracted on May 28, 2012 <<http://www.nova.edu/ssss/QR/QR15-3/mcnaught.pdf>>
- Pyysiäinen, J. (2003) Building Trust in Global Inter-Organizational Software Development Projects: Problems and Practices, In International Workshop on Global Software Development, (co-located with ICSE 2003), pp. 69-74, Extracted on March 6, 2012 from <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

Scrum (2011) The Scrum Guide, Extracted on January 11, 2012, from
<<http://www.scrum.org/storage/scrumguides/ScrumGuide.pdf>>

Softhouse Education (2012.). Scrum in five minute, Extracted on January 14, 2012,
from <<http://softhouseeducation.com>>

Sutherland, J. & Schwaber, K. (2011). The Scrum Papers: Nuts, Bolts, and Origins of
an Agile Process Extracted on January 10, 2012 from <
<http://jeffsutherland.com/ScrumPapers.pdf>>

Thesis Handbook (1997). Qualitative Research. Module R14. James P. Key. Oklahoma
State University. Retrieved on March 29, 2012,
<<http://www.okstate.edu/ag/agedcm4h/academic/aged5980a/5980/newpage21.htm>>

Weaver, P. (2007) Trends in Modern Project management, Past, Present & Future,
Extracted on January 16, 2012 from
<http://www.mosaicprojects.com.au/PDF_Papers/P061_Trends_in_Modern_PM.pdf>

Zenterio AB official web-site (2012). Extracted on May 10, 2012 from
<www.zenterio.com>

Appendices

Appendix 1: Scrum meeting observation report, Zenterio AB

Organisation: **Zenterio AB**

Adress: R&D Sweden
Diskettgatan 11B
583 35 Linköping

Date: 2012 05 03

According to the methodology part of research, the observation was structured in 4 major groups: non verbal behaviour, extra-linguistic behaviour, special relation observation and physical traces. The observation lasted during whole meeting, which was around 15minutes. Total of 6 people participated in the meeting.

Non verbal behaviour

Body movements:

Most people had arms-crossed posture, otherwise caped their arms in the pockets. We could not see some special notes or documents brought to the meeting. People were dressed quite casually and informally (Figure 1).

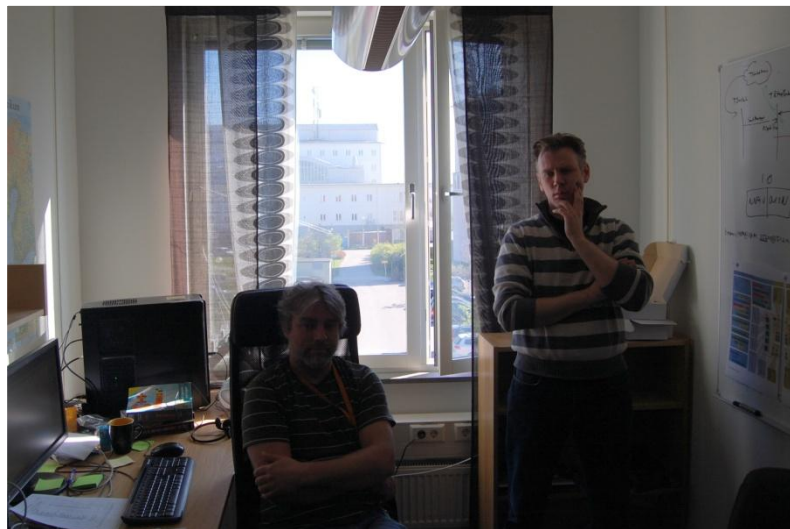


Figure 1 Position and posture of team members during scrum meeting.

It was quite obvious that the meeting was not very stressful; people were able to debrief themselves and see each other.

Group behaviour:

Generally people seemed happy, there were some smiles on the faces then they listened to their colleagues talking (Figure 2). They were also interested in listening and supportive to each other. In one separate case, the presenter on his own initiative raised a question regarding his part of work.



Figure 2 Meeting participants interacting and looking happy.

We observed that it was easy to interrupt each other during a meeting in order to ask question, share experience or clarify the issue. We could also see and feel that if someone from the team was stuck, other would raise the question. In that sense it was not possible for employees to do nothing at all.

Other:

We were surprised that a person, who seemed to be running the meeting, when asked about his own work was not confident in presenting it and went through tasks quite poorly. However, it was also quite notable that scrum master was more technically competent comparing to team members.

Extra-linguistic behaviour

Vocal loudness and duration:

Every member was greeted with applauds, which created the warm and motivational feeling. Generally the speech was clear and moderately loud, just enough for particular room. Each participant had almost equal amount of time to speak and discuss, roughly around 2-3 minutes per person.

Some people were singing in the corridor due to one employees birthday, the meeting was interrupted, the door was opened and agile team expressed their congratulations with applauds and quite informally.

Timbre and Temporal rate:

Voice timbre was getting much lower while employee would discuss his own work and achievements; generally people were much louder while asking questions or giving comments. The rate of speech was also slower during the own presentation, rather



Figure 3 Office room served as scrum meeting room.

than during group discussion or asking a question.

Spatial relation observation

How people organize space:

The room where meeting took place, was in fact the office room of two employees, those employees remained sitting, while other 4 people were standing. This could be also effect by our presence, since the room was quite limited in space (Figure 3).

How close people place themselves:

One the one hand, the room was small so it was engaging people to speak; they were quite close to each other. On the other hand, it was obvious that even in limited space everyone took maximum distance, around 1-1.5m, from each other, there were no hugs, standing close to each other, pairing-up (Figure 1).

Other:

The information on the board was quite limited, not much used and barely referred to, except one person. Tasks on the board were divided into three groups: queue, doing and done (Figure 4). Neither could we see the burn-down chart or any sort of time tracking on the board.



Figure 4 Part of the board used for scrum meeting; tasks divided into three groups depending on accomplishment.

Physical traces

Pictures, board, tools used:

Marker notes on the board are partly erased, the structure of board was quite simple (Figure 4); also part of the board was dedicated to some drawings not relevant to the meeting, neighed they were cleaned before the meeting (Figure 5).

Number of notes:

Stories on the board were represented by larger white paper with information on. These stories are being picked-up by meeting participants, in our case only one story was picked-up by one person, five others did not touch the board.

Oftenly used space, places:

We could see that the wall under the board was quite dirty because people aligned against the wall and touched it with their shoes (Figure 6). Standing in such way they were facing the board with the back, and probably not looking at it most of time.

Other:

Then one of the questions was raised during the meeting, instead of using notes, employees were able to use their monitors and TV's to recall the programming issue and display it right on the monitor during the meeting (Figure 7). At that moment, most of people approached the monitor and were standing closer, after – came back to initial places.



Figure 5 Notes, drawings and printed information on the board.



Figure 6 Shoe marks on the wall.



Figure 7 Equipment in the room allows team members to load and refer to problem areas during a meeting.

Appendix 2: Interviews Guide

The following table is an interview guide that will be used to collect data through semi-structured interviews; and it consists of Introduction, a set of questions and closing comments.

<p>Introduction key components:</p> <ul style="list-style-type: none"> • Thank you • Our Name • Purpose • Confidentiality • Duration • How interview will be conducted • Opportunity for questions • Asking for permission to record interview 	<p>We would like to thank you for taking the time to meet with us today. Our Name _____; we are students of Business Administration in Linköping University and currently we are writing our Master Thesis. The main purpose of our study is to understand how agile software development and outsourcing work in a combined manner. Hence our interest is to talk to you about your experience on ‘Agile Software Development’ and ‘Outsourcing practice’.</p> <p>The Interview should take around one hour and we would like to ask for your permission to tape the session so that we don’t miss any of your comments. All responses will be transcribed and send to you so that we can have your ‘Ok’ signal (thumb-up) in order to use it for our study. We would like to inform you that you don’t have to talk about anything you don’t want to and you may end the interview at any time.</p> <p>Are there any questions about what I have just explained? (verbal consent)</p>
<p>Questions:</p> <ul style="list-style-type: none"> • Open-ended Questions • Ask factual questions before opinion • Use probes as needed 	<ul style="list-style-type: none"> • Can tell us about yourself: your name and position (responsibilities)? • How big is your company? Follow-up questions -How would you define your Industry? And, who are your customers? • Are you using Agile Software development? (Yes or No)- If yes, when did you start using it? • What are the reasons (drivers) two switch from classical project management to Agile Methods? Follow-up question- What significant changes (improvements) have you noticed? • How many projects do you currently have? Follow-up questions- can you tell me about the size of the project and teams? • What is the team composition (Generalists or Specialist)? Follow-up. How long do the team members stay together? • How are activities coordinated among different teams within the same projects? Follow-up- how is inter-dependence handled among different teams? • What are the implications of co-located and distributed teams for Agile projects? Follow-up questions- Any challenges related to that? • Do you use outsourcing? (yes or no) If yes, what are the drivers (reasons) to outsource? • What challenges do you face while outsourcing part of agile software development? • What factors do you consider while deciding to outsource agile projects?
<p>Closing Key Components:</p> <ul style="list-style-type: none"> • Additional Comments • Thank you 	<ul style="list-style-type: none"> • What is your opinion on the long-run fate of Agile Methods? • Is there anything you would like to add? • We’ll be happy to send you a copy of the thesis, if you are interested. • Thank you for your time!