OULUN YLIOPISTO
UNIVERSITY of OULU

Agile Methods in Small Organization Environment

University of Oulu
Information Processing Science
Bachelor’s Thesis
Aleksi Värttö
21.7.2020

Abstract

Agile methods are a rising star and a proven concept in software development processes. They have revolutionized the ways of project management and are becoming more common than older methods day by day. This thesis studied in which way small organizations are different to large organizations, what kind of social and environmental variables there are in small organizations and how well most used agile methods fit into small organization environment. This thesis was conducted as a literature review. In this thesis, it was concluded that agile methods are suitable for small organizations when compared with differences between small and large organizations environmental and social variables. Major differences between small and large organizations being that small organizations are working in constantly changing and turbulent environment, which requires flexibility and ability to adapt. Also, small organizations are suitable for agile methods due to their social aspects, mostly for their need for effective communication and flexibility. In the future, the subject of this thesis should be headed more towards large organizations, because fitting of agile methods into large organization environment has been studied less, mostly since agile methods are naturally suitable for small organizations. It is also important to head the research towards large organizations, because it has been concluded that it is indeed the large organizations that have had problems with implementation of agile methods.

Keywords
Agile methods, Small Organization, Scrum, Extreme programming, Kanban

Supervisor
Jouni Lappalainen

Contents

1. Introduction

This study aimed to research what kind of environmental differences there are between small and large organizations and whether agile methods, such as Scrum, are suitable for use in small organization environment based on the differences in environments and social factors. First, it considered overall scalability factors of agile methods, then it presented differences between small and large organizations environments and finally it considered which agile methods would be suitable for small organizations based on these factors.

Agile methods provide more flexible alternative to older project management methods, such as waterfall method. They have made software development process more flexible and responsive to different scenarios and challenges by providing an approach that is designed to tackle uncertainty in software development. (Coram & Bohner, 2005). However, it may seem like agile methods are not being used to full extent in small organizations, as it was found out in a survey by Rodríguez, Markkula, Oivo and Turula (2012) that from respondents only, from which 60% were considered small organizations, 34% used only agile and 42% didn’t use agile methods. It seems like that small organizations tend to stick to non-agile software development methods or worse yet, to no software development method at all. Agile methods may seem unapproachable to small organizations, or small organizations may rather think that agile methods are not of value to them. This study contributed to these questions by studying possible factors why agile methods would be suitable and productive for small organizations, what are the environmental factors that possibly limit the ability of small organizations to implement agile methods and why small organizations have vastly different environment when compared to large organizations.

Small organization means an organization or project with less than 50 employees or project members. According to Richardson and von Wangenhein (2007), small organizations represented 85 percent of all software organizations in US, Brazil, Canada, China, India, Finland, Ireland, and Hungary in 2007. Therefore, small organizations are clearly important part of any country’s economic growth. Large and small organizations both need to manage and improve software processes, deal with rapid technology advances in information technology field, maintain products and sustain their organizations (Richardson & von Wangenhein, 2007). What makes small organizations different from large organizations is that small organizations tend to be more flexible and responsive than large organizations, because that is the competitive advantage they have over larger organizations. However, small organizations usually lack in resources available to implement, for example new methods. (Richardson & von Wangenhein, 2007.) Small organizations resources are usually tied to maintaining their products, rather than to performing secondary tasks. Also, small organizations suffer from tight finances, which constraints their capabilities of acquiring required expertise to perform new and more challenging tasks (Richardson & von Wangenhein, 2007).

In the time of writing the article by Cohen, Lindvall and Costa (2004), agile methods were still a quite new concept in the software development field. However, basic principles of agile methods have not changed much since. In the implementation of the traditional methods, the work begins with full documentation of the requirements, and then followed by high-level design, development, and inspection. Goal of agile methods is to allow organization to be agile. Small organizations have a need to be more agile than larger organizations due to their nature. Being agile means being able to "Deliver quickly. Change Quickly. Change often.". (Cohen et al. 2004).

In most agile methods, the software development is split into iterations, which allows the development team to adapt to varying situations, challenges and changing requirements in a more flexible way than traditional methods, such as waterfall method, would allow. Agile methods also have heavy emphasis on communication between developers in the organization or project, due to the flexible nature of agile methods.

Research question of this study is:

RQ: Are agile methods suitable for small organizations, and if so, how can we implement agile methods into the organization?

This study answers the research question by first considering environmental differences between small and large organizations, then researching social and PM (Project management) aspects of small organizations and then presenting four different possible implementations of agile methods by reflecting upon these factors.

This study was conducted as a literature review. Material for this study was found by using scientific databases such as Scopus and full text databases such as IEEE Xtreme. The search results were limited by using suitable search strings, such as "agile methods" and "small organizations", or a combination of these. Google Scholar was used for better material availability, if material was hard to find or unavailable otherwise.

The contribution of this study was, that it presented a probably suitable starting point for using agile methods in small organizations that do not yet use one. It also considers the possible reasons why agile methods can be suitable for small organizations and why research wise it could be more beneficial to rather study agile methods implementation in large organizations.

This study has five chapters, including introduction. Chapter two presents prior research on this subject and aims to present the current state of the research. Chapter three presents the findings of this study. Chapter four presents the discussion on the findings of this study. Chapter five is the last chapter and presents the conclusion of this study.

2. Agile Methods and Small Organizations

This chapter presents the prior research of agile methods in small organizations. First, it presents agile methods scalability, then agile methods to small organizations and finally presents four possible implementations of agile methods into small organizations.

2.1 Agile Software Development Scalability

Agile methods are most suitable for turbulent, constantly changing environment, therefore, agile methods are ideal for smaller teams (Boehm, 2002). Boehm (2002) stated that it is mostly the large organizations that have had problems with having success with agile methods. The tightly coordinated teamwork required by agile methods to be efficient is increasingly difficult when team size grows beyond 15 or more people. (Boehm, 2002.) This is to be considered in contrast with small organizations vastly different approaches to manage and improve software processes, deal with technology advances, to maintain products, to operate in a global software environment and sustain their organizations through growth (Richardson & von Wangenhein, 2007). Also, complexity factors when implementing agile at scale, mainly geographical distribution, and organizational distribution support the idea of small organizations being more suitable for agile methods than large organizations (Ambler, 2007).

Agile methods are hard to scale up and are trivial to implement into more traditional, top-down development organizations (Boehm & Turner, 2005). However, due to the nature of small organizations being more flexible and responsive than larger organizations (Richardson & von Wangenhein, 2007), agile methods are that way naturally more suitable to small organizations or development teams, whereas it is more so the large organizations that have troubles implementing agile methods when there are multiple teams of teams working on a project (Reifer, Maurer & Erdogmus, 2003). This view of environmental turbulence was supported in an article by Dyba (2000), in which it was recognized that small organizations require an improvement approach that recognizes the need for a shorter time frame between planning and action; planning an action does not provide all the details of its implementation; and that creativity is necessary to make sense of the environment.

When implementing agile at scale, it is likely to run into some form of complexity factor. These kind of complexity factors could be team size, entrenched culture, legacy systems, or organizational distribution. Agile is relative, different kind of environments require different agile approaches. Scaling agile to meet different complexities is possible, and it is likely that agile methods scale better than more traditional development methods. (Ambler, 2007). With agile state of mind, we are constantly looking for chances to increase the scope of a project, or we may want to decrease the scope of one project to increase the scope in other. Therefore, the projects arrive and are finished in stream, which involves constantly identifying deadlines, managing project scope and task-level management. (Nicholls, Lewis, & Eschenbach, 2015). This kind of resource management is described to be crucial to small organizations success by Richardson et al. (2007), as small organizations competitive edge over large organizations is their ability to be as flexible and responsive as possible, where large organizations tend to be more heavier in their maneuvering.

The simplified agile method for small teams described in the article by Nicholls et al. (2015) that is built around three key project management activities; Identifying hard and soft deadlines; Scoping, prioritizing and selecting projects and; Managing the tasks - to be done now, next task to start, current tasks and tasks completed. In simplified agile, the tasks should be small; no team member focuses on the same task unless the team is preparing, for example an conference call; In the project start, documentation for project should be around 80% complete and; Regular communications are critical to project success. (Nicholls et al. 2015.)

Like discussed by Richardson et al. (2007), key project management activities also are in line with key competitive differences between small and large organizations, such as being flexible and responsive. In the article by Nicholls et al. (2015) it is described that in more traditional approach to project management (PM) the project is scheduled using a defined scope, estimated task times, a network of task dependencies, and estimated availability of resources. This is not in line with key differences between the competitiveness of small organizations versus large organizations discussed by Richardson et al. (2007). Agile methods fit small organizations better, than for large organizations (Reifer, Maurer & Erdogmus, 2003), this view is supported by the views of competitiveness of small organizations (Richardson et al. 2007) and the difficulty to scale agile methods upwards, rather than to adopt them in a small organization scale (Boehm & Turner, 2005).

2.2 Small Organizations and Agile

Small organizations are generally responsive and flexible and usually are involved in many projects (Lee & Yong, 2013). This view is solidified in discussion by Richardson et al. (2007), as this is generally the very competitive edge of small organizations over large organizations. Small software projects are usually six months or less in length, has ten or fewer members, has a single solution, has a narrow scope and has less available funding.

What makes more traditional approaches ineffective to small projects are lack of planning, low priority, inexperienced team members, project manager being responsible of too many functions and use of standard project management tools in small project environment. (Lee & Yong, 2013.) Ineffectiveness of traditional project management methods in small organizations is also discussed by Nicholls et al. (2015). When compared with key competitive edge of small organizations by Richardson et al. (2007), agile methods are more effective in small organization environment.

In agile software development, learning is critical, and learning is not possible without communication. Small organizations or small teams are more effective in communication, collaboration, and coordination (Mishra, D. & Mishra, A. 2008). Formal communication, for example structured meetings or inspections, are useful for coordination, while informal communication, for example telephone calls, are needed when there is uncertainty or unexpected problems. This kind of agile and responsive communication is what makes small organizations environment suitable for agile methods, in a way described by Richardson et al. (2007) and Reifer et al. (2003).

Agile software development methods rely on effective collaboration and communication between project members. This kind of communication is divided into two forms; working together to reach common goals and the other is discussion between team members to resolve an issue. If there is more than 30 meters between, the movement to engage with someone in discussion is drastically reduced, as if being in different buildings. There are also risks in adoption of agile methods for small organizations. (Mishra & Mishra, 2008.) Complexity factors geographical distribution and organizational distribution (Ambler, 2007) are tackled in small organizations in a way that teams are usually located on-site and in smaller quarters than in larger organizations, therefore their communication is more effective (Mishra, D. & Mishra, A. 2008).

There are two groups of limitations in adoption of agile methods: personnel limitations and product limitations (Taylor, Greer, Sage, Coleman, McDaid, Lawthers & Corr, 2006). Personnel and product limitations are also discussed being present in small organizations by Richardson et al. (2007) in a form of resource constraints when compared to large organizations.

Agile methods work best when the environment has high amount of change and the other way around with plan-driven methods. In a survey conducted by Rodríguez et al. (2012) among Finnish software professionals by using an online survey tool. Target of this survey was software practitioners from Finnish association of ICT professionals, The Finnish Information Processing Association (FIPA). From the membership registry of FIPA, email addresses of said ICT professionals were gathered, suited for the survey (Rodríguez et al. 2012).

The survey was designed to describe the usage of agile and lean methods, principles and practices within the organizations and explore the reasons for adopting agile and lean methods into the organization. They received responses from professionals of 200 different organizations, from which approximately 60% would be considered small organizations. Roughly 34% of the respondents used only agile and roughly 42% did not use agile or lean methods. From agile methods, the clearly most common method was Scrum with roughly 83% usage among respondents. In the survey the most common agile methods used by software companies in Finland were found. The five most used agile methods were Scrum (83,1%), Extreme Programming (18,1%), Agile Modeling (11,4%), Feature-Driven Development (8,9%) and Kanban (4,7%). (Rodríguez et al. 2012.)

2.3 Implementation of Agile Methods in Small Organization

Main challenge for the small organization is to find a way to apply the most suitable agile practice into its development. In article by Lee and Yong (2013) an agile framework for small organizations is presented. The framework utilizes 4-Dimensional Analytical Tool (4-DAT) to assess degree of agility of agile methods. The framework is consisted of two main components: the AFSP (Agile Framework for Small Projects) process and AFSP practice pool. The following five steps are then performed; identify the five risk-based agility factors; determine the final set of agile practices; project build and deployment; assess an agile adoption or improvement level and evaluate the project success.

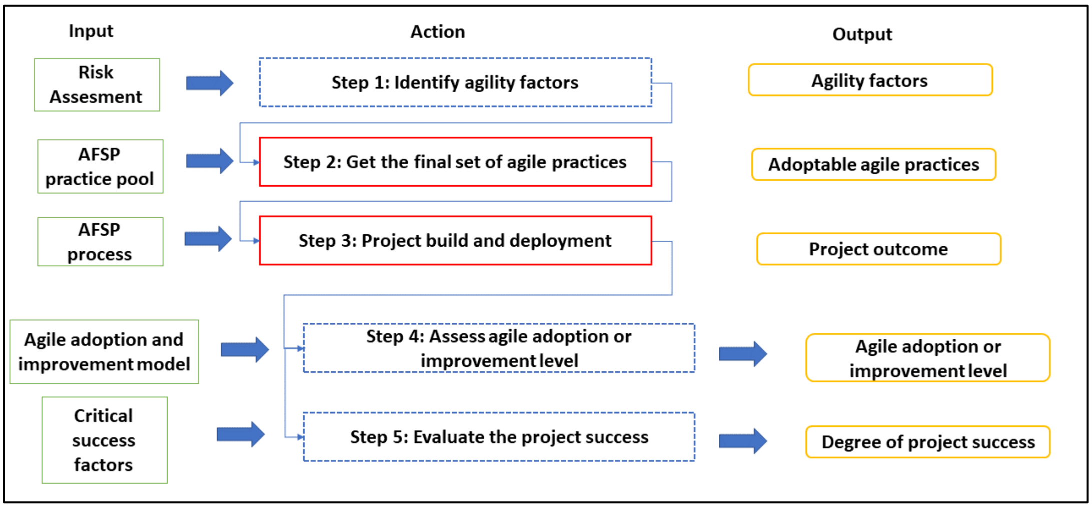
Input
Action
Output
Risk Assesment
Step 1: Identify agility factors
Agility factors
AFSP practice pool
Step 2: Get the final set of agile practices
Adoptable agile practices
AFSP process
Step 3: Project build and deployment
Project outcome
Agile adoption and improvement model
Step 4: Assess agile adoption or improvement level
Agile adoption or improvement level
Critical success factors
Step 5: Evaluate the project success
Degree of project success

Figure 1. Steps of the AFSP in a flowchart (Lee & Yong, 2013).

In the figure 1 above, we see the steps of the AFSP. In step 1, the team identifies the risk-based agility, these being size, criticality, dynamism, personnel, and culture (Lee & Yong, 2013). In step 2, the team determines the final set of agile practices where the team's strengths can be applied, and risks minimized. In step 3, the team focuses on the development and deployment of the project following the agile practices chosen in step 2. In step 4, the team assesses an AAIML (Agile Adoption and Improvement Model Level), which has six levels. This step can be used to assess the effectiveness of agile methods performance inside the organization. In step 5, the team evaluates the success of the project. (Lee & Yong, 2013.) Unlike in Scrum, Kanban, and XP, in AFSP set of agile practices are found for the project and therefore it is not a complete agile method itself. AFSP proved to be effective framework when compared to other agile methods, such as Scrum. The AFSP is described to be an effective and low effort start point for organizations that are looking to implement agile methods. (Lee & Yong, 2013.)

Scrum is an agile method that is described as a suitable development process for small teams (Dingsøyr, Hanssen, Dybå, Anker & Nygaard, 2006). Scrum was the most used agile method in organizations that use agile (83.1%), as found out by Rodríguez et al. (2012) in their survey. The theory behind Scrum is that software development is complex process, therefore it is either difficult or almost impossible to plan, like we would do in waterfall model (Dingsøyr et al. 2006). For this reason, it seems to be an attractive alternative for small organizations, which general environment is in a constant state of change (Richardson et al. 2007).

In Scrum, there is a team of eight persons and a Scrum master, who leads scrum meetings, identifies the initial backlog, and measures the progress toward the delivery goals empirically (Rising & Janoff, 2000), where in Kanban and XP, there are no specific team idea behind, rather in Kanban the workflow is visualized by the Kanban board (Rola, 2011) and in XP the new requirements are directly sent to the team (Erickson, Lyytinen & Siau 2005). In Scrum the development process is divided into sprints, which are variable in length. During sprints, the Scrum master holds scrum meetings with the team members, which should last 15 to 30 minutes. During a Scrum meeting, team members should only answer three questions; What have you completed? What problems did you face? and What will you do next? (Rising & Janoff, 2000). Scrum extends incremental software development to empirical process control, where the core element of Scrum is feedback loops (Dingsøyr et al. 2006). In the study by Dingsøyr et al. (2006), they found out that Scrum has yielded positive results in the cross-organizational project they researched. The developers and customers were satisfied with the Scrum method, as it was easier to see what was done on the project and what was still left to do (Dingsøyr et al. 2006).

In a case study conducted by Romano and Da Silva (2015), they studied a small organization which had difficulties in project management. They implemented the Scrum into this organization in four steps; set up infrastructure; train the team; deploy the Scrum agile and refine the deployment of Scrum. After three sprints of development, a questionnaire was conducted within the project team. Results of the questionnaire show an increase in satisfaction towards overall project management of the development team, from mostly dissatisfied to mostly satisfied. As positive results of the study, the development team recognized flaws in not having a development method, they also saw positives in the ways project goals and progress was visualized as a part of agile methods. Positive result was also that the team now has more motivation to face new projects with the support of the agile method, it specially built their confidence. As negative results, it was observed that the small organization environment causes external interrupts during a project. This is since a lack of employees leads to one person having multiple roles inside an organization, therefore a developer was often interrupted to resolve issues on other projects. (Romano & Da Silva, 2015.)

Extreme programming (XP) has four values in its core: communications, simplicity, feedback, and courage. Its laid around four main activities: coding, testing, listening, and debugging. (Erickson et al. 2005.) Small organizations require agility and communication due to their turbulent and constantly changing environment and need for flexibility (Lee & Yong, 2013). In short, XP means to “embrace change”. Basic ideas behind XP are also in line with the challenges small organizations face, mainly the turbulent environment (Richardson et al. 2007), constantly changing scope and, tighter budget restraints (Lee & Yong, 2013).

In XP, the process is to code what the customer specifies and test that code to ensure accomplishment of the developer’s desire in the prior steps of development process. In the process, there are no anticipations for tools or features, because XP is oriented toward providing a finished product in timely manner. Idea behind XP is that if any features or other needs regarding the product are required by the customer in the development process, the customer will notify the development team about these new needs, the development team do not worry about these needs for the present. (Erickson et al. 2005.)

Providing a finished product in timely manner is crucial to small organizations, who are working with tight schedules and budgets (Lee & Yong, 2013). This kind of approach is supported by views described in the article by Richardson et al. (2007): small organizations have limited resources for a project available at times when compared to large organizations, for this they have to be able to deliver on time. XP is more focused around the idea that an organization must push a product out, or to finish a project (Erickson et al. 2005). While in Scrum and Kanban, the basic ideas are more focused around either managing a team and its goals (Rising & Janoff, 2000) or visualizing workflow and progress (Rola, 2011).

Kanban is an agile method which main principles are visualizing the workflow, limiting work in progress (WIP) and measuring the lead time. In Kanban, visualizing workflow is mainly done through a tool called Kanban board. This board can be a whiteboard with post-its stickers on it. The board is split into different sections depending on the needs of project management, in these sections there are tasks which are designated to different team members. (Rola, 2011.) Kanban board is important part of Kanban method, because it contributes to the method’s overall transparency and waste reduction. There are seven wastes of software development: partially done work, extra processes, extra features, task switching, waiting, motion, defects (Rola, 2011). Reduction of this waste could be beneficial for small organizations when comparing various negative effects of waste on projects, such as waiting hindering the project teams ability to respond quickly to changing situation or partially done work later becoming obsolete or requiring reworks (Rola, 2011) with statements about small organizations limited capabilities regarding available resources, workforce limitations and budget described by Richardson et al. (2007).

3. Methodology

This research was conducted as a literature review. Literature review is a research method where the researcher identifies and organizes the concepts of relevant literature. The objective of literature review is to summarize the current state of the subject field. From this research, it is now possible to find and identify different areas in which it would be beneficial to study it further. According to Rowley and Slack (2004), literature reviews are important in supporting the identification of research topics, questions or hypothesis; identifying the literature to which the research will make a contribution; building an understanding of concepts and terminology; facilitating the building of a bibliography; suggesting research methods that might be useful; and analyzing and interpreting results. (Rowley & Slack, 2004.)

Material for this study was found using mainly the scientific databases, such as Scopus and full text databases such as IEEE Xplore. Google Scholar was used as a secondary platform due to it having better availability of full material in readable form (PDF). Finding material was first started by using general and common words regarding the subject, such as “agile methods” in conjunction with “small projects” or “small organizations”. Therefore, most common search strings used to find material were in line with “‘Small Organizations’ AND ‘Agile Methods’ “or “‘Small Project’ AND ‘Agile Methods’ “. These kinds of searches yielded multiple results and it was critical to limit the search result. Then, search results were limited by inserting more fitting search words, such as “project management”, into the search. If search results were still wide, the search was limited to certain subject. After limiting search results with keywords and subject, the articles were chosen by evaluating its usefulness to this thesis by first reading the title, if the title seemed to fit the criteria of the thesis, read the abstract, if abstract already indicates that the article fits the research, pick the article as a source. If after reading abstract it was still unsure whether the article fits the subject of this thesis, reading the article further would ensue. Picked articles sources were also evaluated with similar chain of evaluation as for sources found via searches to find additional sources (citation pearl growing).

4. Discussion

This chapter discusses the differences of overall environment differences between large and small organization, what makes small organizations different and what kind of key social and organizational environment elements small organizations have. It was then evaluated which of the agile methods or framework presented in earlier chapter would fit small organization environment based on the factors and overall differences between small and large organization environments.

Main differences between large and small organizations are the overall resources these instances hold. Small organizations tend to have much more limited resources in a context of project budget, employees, technical knowledge, process management knowledge, overall growth of the organization, project size, project length and project scope. For small organizations, managing the budget of projects is important, since the whole company's future might very well be hanging from the success of a single project. Therefore, it is crucial for the organization to allocate resources carefully between projects. Small organizations might see implementation of agile methods as an insurmountable weight and not of value to the organization, and not worth the required extra resources. The organization might lack the required knowledge or right employees to implement agile methods, which would lead into higher costs and time required. Small organizations tend to not have the extra resources or time to allocate into sizeable projects beside their main customer projects. Therefore, small organizations might see implementation of agile methods as an encumbrance, rather than longer term profit for the organization. In an article by Mohagheghi and Jørgensen (2017), it was stated that projects with agile development and fixed scope have a success rate of 58%, and projects with agile development and flexible scope had a success rate of 87%, with 12 and 15 projects studied respectively. It should be clear from this result alone, that implementing agile methods into a small organizations software development should be beneficial for the organization in a long run. The success rate of projects with flexible scope also reflect upon the key differences of small organizations; the need to be flexible and adaptive in constantly changing, turbulent and competitive environment small organizations face.

The two main limitation groups in implementing agile are product limitations and personnel limitations. Product limitations are limits to capability of producing reusable artifacts, developing safety-critical software, and developing large software. Personnel limitations are limits to distributed development environments, subcontracting and large teams. (Taylor et al. 2006.) In small organizations, both limitation groups are present. Personnel limitations are more clear, and are usually paired with budget constraints which are in most cases present within small organizations, as with constrained budget the organization lacks the ability to hire more experienced personnel or outsource projects or parts of a project, hindering the ability of organization to acquire knowledge. These limitations and the implications they present, that agile methods might be too risky to implement might push small organizations away from developing with agile methods. It is also a trend in small organizations not to use any agile method in their software development, even if it is shown to increase the overall project success rate.

Agile methods are based on excellent communication between team members. Communication is one of the key principles of agile development described by Mishra, D. and Mishra, A. (2008). In small project teams or organizations, communication comes more naturally due to the nature of small organizations environment, which is very turbulent and constantly changing, requiring excellent communication to keep up with the situation at hand. Small organizations also are in smaller office spaces, which further enhances their ability to communicate effectively. Small organizations usually have all their staff located on the same site, meaning there are no off-site communication required with different project teams, further increasing the effectiveness of communication in a small organization. Small organizations are generally quicker learners due to their smaller requirement or rather smaller threshold of starting a conversation with a co-worker inside office spaces, since when people are less away from each other, the lower threshold there is to start a conversation. This kind of conversion, which is not structured and is open, is an excellent platform for quick and effective learning. Also, this open structured conversation can easily lead to more structured meetings regarding the issues or findings talked about out in said conversations, making the whole process more effective in terms of learning process.

Presented as a potential agile method to implement into small organization in this thesis were four different methods. These were the AFSP (Agile Framework for Small Projects) described by Lee and Yong (2013), Scrum, Extreme programming (XP) and Kanban. These methods, much like any agile method available, are suitable to small organizations by their nature and that they have been used and are being used constantly in small projects or organizations. These agile methods are focused on effective communication and quick learning. The AFSP is the most leanness of the presented methods of agile implementation, as it is not itself a full agile method, rather it is used to find suitable agile practices for different projects or organizations. Scrum, Kanban, and XP are full agile methods, and they require more effort to implement resource wise from the organization.

From these, the AFSP is probably the most suitable entry point to start implementing agile methods for small organizations due to the fact that since AFSP only finds suitable practices to implement, it is not as heavy on the organization to start using agile practices. The AFSP is a framework, that allows small organizations to find and pick individual pieces, agile practices, rather than to implement a complete agile method, such as Scrum into the organization. This kind of partial implementation, or implementation of suitable practices has lower costs than implementing a full agile method, also it is more effective time cost wise, reducing the overall time of implementation. AFSP had proven its value in four different cases by being more effective than Scrum, Kanban, or XP. However, AFSP is more lean approach and is not as complete as full agile methods. This kind of approach of implementing agile is also a starting point for organization to start with agile methods and to later expand into actual agile method that utilizes or is center around the suitable agile practices found with AFSP. The clear strongest point of AFSP is its leanness when compared to complete agile methods, making it a quick and rather cost-effective resource wise alternative to full agile methods.

Scrum and Kanban are quite modern agile methods, which both revolve around efficient communication and learning. From these, Scrum method was the most used one in organizations, as was found out in a survey by Rodríguez et al. (2012), Kanban was at the time of writing the article the 5th most used agile method, behind Scrum and XP. In Scrum, the basic idea is that a development process is split into sprints, which vary in length. In these sprints are different kind of tasks, that can be anything from designing to testing. These tasks are handed to and supervised by a Scrum master.

In Kanban, the main idea is visualization of progress. Kanban uses a visualization board, which has different lanes representing phases of task completion. A task is then first assigned to a person and placed in backlog. Task is then moved between the lanes by the assignee depending on the level of completion of said task. Both Scrum and Kanban can be deemed suitable for small organization environment, due to their focus on effective communication and emphasis on learning. Both are also extremely flexible and are most suitable for turbulent small organization environment. However, both might be at start more tasking to implement when compared to AFSP and XP, since both have more complexities and therefore require more trained staff or guidance for successful implementation, which might be off putting to small organizations who might struggle resource wise.

Extreme programming has four core values: communications, simplicity, feedback, and courage (Erickson, Lyytinen, & Siau, 2005). Communications and simplicity are crucial for small organizations due to their lack of overall resources and time, which implies the need for simplicity of implementation. As XP is oriented towards providing a finished product, it fits the needs of a small organization, which wants to be able to close projects and provide finished products to their customers in timely manner, due to the constant need of finishing projects and beginning new projects. XP was the second most used agile method in 2012, therefore it has proven its place as an industry standard agile method, and, it is one of the oldest agile methods used today with Scrum. For a small organization, implementation of XP could be beneficial for its industry standard status, as this means that the method is well documented, which eases the threshold of implementation in terms of resources.

5. Conclusion

This chapter discusses the results of the literature review, answers the research question, presents results of the research, its limitations and possible future research.

Research question for this thesis was:

RQ1: Are agile methods suitable for small organizations, if so, how can we implement agile methods into the organization?

Agile methods, due to their nature of being more suitable for turbulent environments (Boehm, 2002), are to be considered suitable for small organizations, which also face challenges of operating in a constantly changing challenges (Richardson et al. 2007). Small organizations are also more reluctant on implementing agile methods into their development; in a survey by Rodríguez et al. (2012), it was concluded that from the responded organizations, only 60% were to be considered small, from which roughly only 34% used any kind of agile method. This indicates that there is a reluctance on implementing agile in small organizations. This reluctance could be caused by the risks caused by limitations of implementing agile methods; personnel limitations and product limitations (Taylor et al. 2006). From these two, personnel limitations are a major limitation in a small organization. Agile methods tend to fit smaller organizations workings better for their excellent scalability for smaller environments (Lee & Yong, 2013), however, that agile methods implementation faces challenges when scaling up to stiffer, top-down large organization environment, rather than scaling down to small and agile small environment (Boehm & Turner, 2005).

The three key project management activities described by Nicholls et al. (2002) reflect upon the basic concept of small organizations; Identifying hard and soft deadlines; Scoping, prioritizing, and selecting projects and Managing the tasks. In small organizations, communication is of utmost importance and it is much easier to have effective communication in small organization environment than in large organization, due to difference in organization culture and varying environments. In office environment, if two persons are more than 30 meters away from each other, they might as well be in different buildings. (Mishra, D. & Mishra, A., 2008.) Naturally, small organizations reside in smaller offices or spaces and therefore have more natural suitability with agile methods through effectiveness of communication between team members.

Small organizations have less employees than large organizations, therefore they may lack certain skills or one team member might face higher workload than others, mainly the project manager, who might also have to work on the project itself, rather than to focus solely on project management. Valid implementation of agile methods into small organizations is not a big challenge, considering that agile methods have a natural tendency to fit better into small organization environment. Main challenge for small organizations is finding an agile method that suits their needs. Lee and Yong (2013) presented a framework for small organizations, with which they may implement different agile practices into the development process. This framework does not directly find a suitable agile method for the organization, but rather it finds suitable practices for the organization. Other way to implement agile is to implement a complete method into the organization.

Romano and Da Silva studied usage of Scrum (2015) within a small organization that had proven to have problems with project management. The study yielded positive and negative results, as Scrum made the project team more confident in their work and more willing to accept new challenges, however, it was also observed that small organization environment causes interrupts during development, due to the smaller environments small organizations work in. This view is supported by Mishra, D. & Mishra, A. (2008), whom in their article discussed that small office spaces, in which small organizations tend to work, lead into more frequent discussions between colleagues.

In the time of the survey by Rodríguez et al. (2012) Kanban was the 5th most used agile method, behind Scrum (1st) and XP (2nd) with 4,7% used in IT organizations. In Kanban, the basic idea behind the method is visualization of workflow and progress. This is achieved by usage of a Kanban board, in which there are different lanes depending on the needs of project, and on these lanes there are different tasks which have been or will be assigned to different team members (Rola, 2011). Kanban is very lean agile method in terms of overall complexity as it mostly revolves around the usage of Kanban board, which is configurable to fill the needs of any project. The leanness of Kanban makes it suitable for any small organization, as it does not require huge investments in terms of resources.

Extreme Programming (XP) could be considered an industry standard at this point, as it was the second most used agile method in IT organizations only second to Scrum (Rodríguez et al. 2012). XP is focused around providing a finished product or finishing the project. In XP, the process has four main activities: coding, testing, listening, and debugging (Erickson et al. 2005). XP is focused towards providing a final product, this is the core. This makes XP an attractive option for small organizations, due to the overall need for small organization to push products out and produce new projects in their turbulent environment.

In this thesis the overall scalability of agile methods was considered. Agile methods are by nature scaled down to small environment setting. Agile methods are meant for turbulent and constantly changing environment, when the requirements of the project constantly change, integration is constant and new features are being designed. Small organizations also have these kinds of features, due to the natural requirement for smaller organizations to be more agile in their ways than large, top down organizations. Smaller organizations tend to have more suitable office environment and social atmosphere for agile methods. Due to smaller office environment, two people are more likely to engage in a conversation than they would in larger areas. Due to agile methods requiring active and effective communication, smaller office areas do have a role to play in their successful implementation into small organization. Therefore, it was concluded that agile methods are suitable for small organizations. Implementation of agile methods into a small organization could be done by using either the framework presented by Lee and Yong (2013) to find suitable practices for the organization or implementing a complete agile method. Scrum method seems to suit most of the needs and special requirements of a small organization. With these methods, it is possible to implement the agile methods into a small organization.

The limitations of the research were the lack of available material, as agile methods are naturally suitable for small organizations, the subject is not researched as much as their suitability for large organizations. In the future, this subject needs more case studies of successful and unsuccessful implementations of agile methods into small organizations, as the literature currently available on the subject is mostly theory. Also, in the future, it would be important to study possible implementations of agile methods into large organizations, as large organizations seem to have more problems when implementing agile methods. This thesis is useful for small organizations which are looking to improve their development process by implementing an agile method, as it presents them reasons of why and possible solutions.

As this research was conducted as a literature review, the results are limited to current literature on the subject. It was found out, that due to agile methods being by nature suitable to small organizations, the subject is not that well studied. Rather, implementation of agile methods into large organizations seemed to be the current perspective of discussion and important perspective in the future, as large organizations have had problems when implementing agile methods. It is likely that using qualitative methods and implementing an agile method into a small organization would be more useful than conducting literature reviews on the subject. However, it is possible that this research can be used as a starting point of implementation of agile methods into a small organization, as it presents why agile methods are suitable and presents possible agile methods to implement or an entry point to start implementing agile methods.

References

Ambler, S. W. (2007). Agile software development at scale. *IFIP Central and East European Conference on Software Engineering Techniques* (pp. 1-12). Springer, Berlin, Heidelberg.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64-69.

Boehm, B. & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, *22*(5), 30-39.

Cohen, D., Lindvall, M. & Costa, P. (2004). *An Introduction to Agile Methods.* Advances in Computers, Vol. 62, Issue C, 2004, pp 18-22.

Coram, M., & Bohner, S. (2005). The impact of agile methods on software project management. In *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)* (pp. 363-370). IEEE.

Dingsøyr, T., Hanssen, G. K., Dybå, T., Anker, G. & Nygaard, J. O. (2006). Developing software with scrum in a small cross-organizational project. *European Conference on Software Process Improvement* (pp. 5-15). Springer, Berlin, Heidelberg.

Dyba, T. (2000). Improvisation in small software organizations. *IEEE software*, *17*(5), 82-87.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: the state of research. *Journal of Database Management (JDM)*, *16*(4), 88-100.

Lee, S. & Yong, H. S. (2013). Agile software development framework in a small project environment. *Journal of Information Processing Systems*, *9*(1), 69-88.

Mishra, D. & Mishra, A. (2008). Workspace environment for collaboration in small software development organization. *International Conference on Cooperative Design, Visualization and Engineering* (pp. 196-203). Springer, Berlin, Heidelberg.

Mohagheghi, P., & Jørgensen, M. (2017). What contributes to the success of it projects? success factors, challenges and lessons learned from an empirical study of software projects in the norwegian public sector. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)* (pp. 371-373). IEEE.

Nicholls, G. M., Lewis, N. A. & Eschenbach, T. (2015). Determining when simplified agile project management is right for small teams. *Engineering Management Journal*, *27*(1), 3-10.

Reifer, D. J., Maurer, F. & Erdogmus, H. (2003). Scaling agile methods. *IEEE software*, *20*(4), 12-14.

Richardson, I. & von Wangenhein, C.G. (2007). *Why Are Small Software Organizations Different?* IEEE Software, Vol.24, Issue 1, January 2007, pp 18-22.

Rising, L. & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, *17*(4), 26-32.

Rodríguez, P., Markkula, J., Oivo, M. & Turula, K. (2012). Survey on agile and lean usage in finnish software industry. *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 139-148). IEEE.

Rola, L. D. (2011). Kanban for small software projects. *Project Background Report, The University of Manchester, Manchester.*

Romano, B. L. & Da Silva, A. D. (2015). Project management using the Scrum agile method: A case study within a small enterprise. *2015 12th International Conference on Information Technology-New Generations* (pp. 774-776). IEEE.

Rowley, J. & Slack, F. (2004). Conducting a Literature Review. *Management research news.* Vol. 27. Number 6.

Taylor, P. S., Greer, D., Sage, P., Coleman, G., McDaid, K., Lawthers, I. & Corr, R. (2006). Applying an agility/discipline assessment for a small software organisation. *International Conference on Product Focused Software Process Improvement* (pp. 290-304). Springer, Berlin, Heidelberg.