Illustrating the Use of Agile Software Development Process

ZISHAN H. MIRZA

Bachelor of Software Engineering and Management Thesis

Report No. 2010:020
ISSN: 1651-4769
ABSTRACT
A software development process can be a challenging and complex task to analyze and evaluate. It is important to have a process that will support the project in a successful way. There are lots of factors to consider when evaluating a software process that fits an organization’s needs. Agile software development process is a very common process that many organizations are using in the recent years. It includes various other methods, such as Scrum, extreme programming (XP), Lean, etc. This research will illustrate how two organizations are using an agile-based software development process in their projects and if agile software development process can be recommended to organizations to use in their software development projects. Data have been collected from published articles and journals, as well as from interviews from two organizations.

1. INTRODUCTION
In today’s world, technology is very important. Most organizations are using some sort of technology to fit their business idea, whether they are in the software industry or not. Technology has also been changing dramatically in the past few years, especially in the software industry. There are new hardware and software and customers want these new hardware and/or software after the initial specification has been written down. A process is way of proceeding with a specific project. It is always recommended to have a process in any project. By having a process, it is possible to organize the project in order to better succeed. Change is an important concept in any project, since recently the customers’ are changing the requirements more frequently. Therefore, it is important to incorporate such a change during the process of a software project. There are a number of software processes that organizations can use, such as, a traditional plan-driven (also known as ‘Waterfall’), agile-based processes (such as Scrum, Extreme Programming (XP), Lean etc.), etc. Careful consideration should be made before using a process. There is no process that guarantees success for all projects. It depends on various factors in the individual project. A software process is needed because it is important to follow guidelines and have a plan over the entire project. It can also help to improve the way in which software is being developed (Chan & Thong 2009). By considering a process carefully, or choosing the positive aspects of different process in agile, it is possible to achieve a high quality product.

Agile has become a popular process (Ivček & Galinac 2009), especially in the software industry. Therefore, it is important to understand the use of such a process and analyze how successful the process is in projects. Thus, this research is a study on agile software development process with examples of two organizations of how they are using agile software development in their software development projects. A literature review has been done to get an understanding of the theoretical background of agile. This research will give an understanding of how actual organizations are using agile-based software development processes in their software development projects.

This paper will start off with an introduction where agile software development process is introduced to the reader and why a software development process is desired to have in a software development project. The Theoretical Background section contains the related theories that are available in published articles, journals, and conference papers. In the section named Research Methodology, this paper will state how the research was initiated and done. The Data Collection section contains the answers to the interview questions. This section is complemented by Appendix A, where all of the interview questions that were asked are stated. An analysis of the data collected is made in the Result and Analysis section. This is where the data collected is related back to the theoretical background. Later on, a conclusion is drawn based on the analysis made in the previous section.

2. THEORETICAL BACKGROUND
In this section, theories will be presented about agile software development, which includes Scrum and Lean as well. The theories will be based on published research articles and journals.

2.1 Agile Software Development Process
Agile software development incorporates various software development processes, such as Scrum, Extreme Programming (XP), etc. This is a rather relatively new phenomenon that came about in the software industry. Agile-based software development processes are iterative processes where
the different phases in the development cycle are repeated a number of times. The agile methodology actually originated from a few software engineers who wrote the Agile Manifesto. This is a set of guidelines for using this type of methodology (Wikipedia 2010a). The manifesto consists of four main principles (Wikipedia 2010a):

- “Individuals and interactions over processes and tools”
- “Working software over comprehensive documentation”
- “Customer collaboration over contract negotiation”
- “Responding to change over following a plan”

In recent years, more and more organizations are favoring towards an agile methodology, rather than a traditional plan-driven ‘Waterfall’ process due to the a lower success rate with a plan-driven process (Berger 2007). One of the reasons for this low success is that the requirements need to be well understood in the beginning of the project (Berger 2007) (Williams & Cockburn 2003). It is not always the case that all of the requirements are perfectly understood in the beginning of the project. An agile-based software development process provides flexibility for various changes that can occur during a project.

There are many organizations that have implemented an agile-based software development processes and have had faster deliverables and higher quality of the product (Dybå & Dingsøyr 2008). This can be true because if there are shorter iterations, then there can be more feedback from the different people working in the team, such as the project managers, developers, testers, and customers. This, in turn, can mean that the communication in the project increases and, thus, have a potential capability of increased satisfaction.

Dybå & Dingsøyr (2008) made a study that showed that organizations using agile software development will deliver faster, the quality of the product will be higher and the product will result in lower costs for the customer (Dybå & Dingsøyr 2008).

Misra et al. (2009) wrote in their paper that there are various factors that affect the success of software development projects using an agile-based software development process. These factors are very important in any software development process. A pure agile software development process does incorporate these factors.

One of the factors is how the customers are involved in the project and if the customers are satisfied with the product itself (Misra et al. 2009). Customer involvement in the projects can be beneficial, since the customer can see what is being developed and reduce any misunderstandings between the customers and the developing teams.

Another important factor is the location of the people working in the team. Distribution of team members can hinder the project in moving forward faster (Misra et al. 2009). There are solutions to solve distributed communication, such as virtual teams, videoconferencing, etc.

Management is a very important factor. It is very important to have a good project and risk management plan and to follow the project’s progress in order to succeed in a project.

Recruiting the right person for the right job is an important task. It is the competency of the people that can decide if the people are able to solve a specific problem. The competency of the people is an important part of recruitment. It is important to employ the people with the correct level of competency. Both knowledge and previous experience can come into the word ‘competency’. The level of competency can also lead to the success of projects. Therefore, if the employees have the correct level of knowledge and understanding in a specific project, there is a higher success rate. If the employees lack the prerequisite knowledge, it can lead to additional training in the organization and will, eventually, increase the cost of the project (Cockburn & Highsmith 2001). If the costs do increase beyond the limit specified by the management team, the project might be required to shut down. It is also important the people working in the project will be able to work in groups, or smaller groups. This is because it is often easier to work in small groups rather than larger groups. Therefore, the behavior of the people in smaller groups is important and can be used as a technique in the recruitment process by different organizations.

Communication is a very important concept in an agile-based software development process. The Agile Manifesto states that “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation” (Engineers website) [http://agilemanifesto.org/principles.html].

Training and learning can be considered as one of the most important factors. In any organization, workers need to learn the new technology (or the current technology for that matter) so that they can be productive in their work. It is necessary to keep learning all the time. People do learn as they work in any organization.

### 2.2 Scrum

Scrum is an agile software development process. Many organizations are using Scrum nowadays. Its popularity is increasing (Moe & Dingsøyr n.d.) and because of this, it is important to introduce what Scrum is in this research. Organizations are using Scrum because the requirements from the customers can often be confusing and misunderstood, and therefore, the project should be progressing at all times (Rising & Janoff 2000).

In Scrum, there is a list of tasks that needs to be developed. This is, of course, according to the customer’s requirements. This is known as the product backlog. The features in the product backlog are then selected according to which features are needed for a specific release of the product. This is known as a release backlog. Each feature is estimated on how long it would take to implement and develop a certain feature. These estimates are then added to get an estimate of the time it would take to develop a certain release. These features are then put into different sprints and made sure that each sprint is completed before starting the next sprint. These sprints vary in length and can last for one week or up to one month on average. There are meetings every day
during each sprint to make sure the sprint is on track (Moe & Dingsoyr n.d.).

The Scrum Master makes sure that the project is progressing. The product owner makes sure that the features are according to the customer’s requirements. An agile way of working is to have someone that would take the customer’s side and is responsible for the customer, which is the product owner in Scrum.

### 2.3 Lean Software Development Process

Lean software development process can also be considered as an agile-based process (Chow, T and Cao, D. B., 2007). This is due to the fact that it is flexible for implementing changes during such a process. The principles of Lean was first implemented by Toyota and was known as Lean manufacturing and is a rather old phenomenon. This phenomenon has recently been applied into software development projects. The principles in Lean software development are the following (Poppendieck & Poppendieck 2003):

- **Eliminate Waste:** Eliminating waste is a technique that is used anything that cannot benefit the customer or the customer’s value is regarded as waste. Lean tries to reduce this.

- **Amplify learning:** This principle is about learning during the process. When developers are writing code, it should be tested immediately in order to reduce the amount of defects and bugs. More code should be written instead of more documentation.

- **Decide as late as possible:** It is better to delay decisions when something is unclear. It can be beneficial in a project because costs can increase a lot if a bad decision has been taken.

- **Deliver as fast as possible:** Faster deliverables will mean that there are more feedback from the customers. It can, therefore, increase customer satisfaction.

- **Empower the team:** The right person for a specific task will be beneficial for the whole project. This is because if the person knows best on what to do with the task will have a better chance in completing the task. In a traditional top-down management approach, this does not happen.

- **Build integrity in:** Integrity can be split up into two sections: perceived and conceptual integrity. Perceived integrity is the usability of the system and what the users think of the system. Conceptual integrity is how the system works as a whole. Both integrities are required in order to achieve a higher rate of quality of the product. This can also increase customer satisfaction since the system should be working as a whole and the users of the system are satisfied with the system.

- **See the whole:** It is important for the project manager to monitor the project as a whole and make sure that the project is on track.

A practical example of the successful use of Lean would be at Ericsson. According to an article in IDG (Computer Sweden) magazine (Computer Sweden 2010), Ericsson has decreased their development time by 50% by using a Lean development process (Computer Sweden 2010). The article mentioned that they are using a Streamline development process. This is a process inspired by a Lean/agile way of thinking that was an in-house development by Ericsson. They claim that they have been using this process as a framework (Computer Sweden 2010) (MKSE.com 2010) and is built on the same foundations of agile manifesto (MKSE.com 2010). They also state that changes have been reduced by 80% and there have been 75% more products that have been delivered. The main important benefit was that they have been developing the correct product to the customers, based on the customer’s needs (Computer Sweden 2010).

They state that they have also reduced the size of the different projects and have reduced the projects to three months in time (Computer Sweden 2010).

### 2.4 Similarities of Agile and Lean Software Development Process

Both agile and Lean software development have various similarities. One of the most important characteristics that both agile and Lean have in common is quick and early customer feedback. One of the principles of Lean software development is to deliver faster (Poppendieck & Poppendieck 2003). This would then result in more feedback from the customers. This is important in order to increase customer satisfaction, since the customer will be able to see the product quicker. The principles of Lean development can definitely be included in an agile process. Since there are more feedback to the customers, there are more deliverables. The agile manifesto actually states that the customers should be involved (Engineers website) [http://agilemanifesto.org/principles.html], and hence the customer satisfaction can be increased.

Another important similarity is that it can accommodate change easier than a plan-driven development process. This is actually stated in the agile manifesto, that the process should be able to handle change (Engineers website) [http://agilemanifesto.org/principles.html]. One of the principles of Lean software development is to delay decisions when concepts are unclear. When concepts are not clear and the project is very complex, then it is important to be able to incorporate change in the project (Wikipedia 2010a). This is an important feature of an agile/Lean software development. Handling change is very important. This is because, because of today’s rapidly changing technology, the customer’s are realizing new features that they would like to have in their product and, thus, changes the requirements. The change in requirements does not have to be the only reason. There are a lot of misunderstandings between the developing teams and the customers. The customers might think about something different than the developers and would not fit the customer’s needs. The developers might consider some feature obvious, while it might not be that obvious to the customer. Therefore, it is important to be able to handle this kind of change in any software development project.
3. RESEARCH METHODOLOGY

The research reported in this paper builds on a case study research where interviews have been conducted at two organizations: two interviews at Ericsson AB and two interviews at Erlang Solutions Ltd. The interview questions can be found in Appendix A.

The interviews at Ericsson were in the form of a semi-structured interview. During these interviews, notes were written down and were mostly short. These notes were written down as quickly as possible and then after the interview, all of the notes were rewritten neatly. There were two interviews, and each interview had approximately one page of notes.

One of the interviewees at Ericsson is responsible for processes, tools, etc., and the other interviewee is a project manager.

The interviews at Erlang Solutions Ltd. were conducted through e-mail and were more in the form of a structured interview. For example, it is possible to either ignore questions like ‘if yes’ ‘if no’ if either one or the other is answered.

One of the interviewees at Erlang Solutions Ltd. is a chief technical officer of the organization and the other interviewee is a program manager.

The data that has been collected from the interviews has been analyzed based on the theoretical background on how organizations are using agile software development process. The success of these projects that the organizations have also been analyzed as well as the meaning of the term ‘success’.

According to Reed (1998), a literature review is very important process in any research project (Reed 1998). There are different kinds of references that is involved in research (Reed 1998). This paper has a primary reference and a secondary reference. The primary reference are the interviews. The secondary reference is the literature review (Reed 1998). Tertiary references has been used only to get the basic information. In the beginning of a research project, it is important to understand how to find the data necessary for the research (Reed 1998). Since there will be a lot of articles on agile software development, it is important to understand the abstract of the article (Reed 1998) and, in this way, it will be possible to understand if the article is relevant to this research.

The literature review that has been made in this research has been done by researching and reading various relevant research articles, conference proceeding papers, journals, and books.

4. DATA COLLECTION

4.1 Case Study 1 - Ericsson

One of the interviewees at Ericsson mentioned that in one of the units looks a lot at the Lean principles as well as at other agile-based methodologies, such as Scrum. There are, however, some projects still running in a traditional plan-driven development. However, these projects are not as successful as the agile-based projects. The interviewee also mentioned that both agile and Lean has several similarities.

Ericsson has huge projects, with 1000 people working. It is very difficult to manage this many people in a project, so they split the project into smaller teams (approximately 7-8 people per team). This is common in many methods, especially those that are inspired by agile development.

Ericsson uses metrics for performance measurements. Some examples were given by the interviewee, such as to integrate all of the code and the classical performance measurements. It gets complex when a multi-national and disperse organization like Ericsson work with people around the globe. These difficulties occur due to the location and the time-zone difference. One of the interviewee pointed out that people had to work early mornings or late evenings in order to schedule meetings. These difficulties can occur in any process and organizations need to have a solution to this. Various collaboration tools are also used, such as IBM Rational Team Concert. They are also estimating the length of a project and comparing this length with the length of the actual project.

For future improvements at Ericsson, one of the interviewees emphasized that they would like to learn more about Lean.

The other interviewee at Ericsson mentioned that in another development unit, they are using Scrum as the development process. They are collaborating with different teams in China, where Scrum is used. They are using various communication techniques to communicate with people in other branches of the same organization, whether they are in the same country or not or in a different time-zone. Scrum is an agile process. Normally, agile-based processes have local communication. This means that all of the people working together are located at the same place. However, this is not possible for organizations that have offices in different parts of the world. Efficient and effective solutions need to be brought on in order to solve this problem. There are various techniques to solve this problem, such as, videoconferencing, phone, e-mail, etc. These solutions are also normal features of an agile-based software development process.

The interviewee also pointed out that during the small sprints in the Scrum process, a ‘Waterfall’ approach is used. Basically, this would mean that they have deadlines that they would follow in each sprint. This also mean that they have specific goals defined in each sprint, where they might have looked at a larger scope and split it up into smaller problems to solve. This is one of the normal features of an agile-based software development process.

According to the interviewee, they are using cross functional teams. The whole project consists of a lot of peo-
ple (can be around 100 people). However, they split up the project into smaller teams. Cross-functional teams is where people with specific expertise are working together, such as, project managers, developers, team leaders, etc. An agile-based process is known to have cross-functional teams, where the teams can decide on what they are developing. One of the benefits of having cross functional teams is that it is possible for everyone in the team to work towards the same ideas in the project. This is because there would be developers, project managers, testers, etc. in the same team. The team can also understand the project better.

4.2 Case Study 2 - Erlang Solutions Ltd.
Two interviews have been conducted at Erlang Solutions Ltd., which is based in the United Kingdom. Both of the interviewees mentioned that they are using a process called Dynamic Systems Development Method (DSDM). This is an agile-based software development process (Wikipedia 2010b). One of the interviewees mentioned that this agile process has been well established in the United Kingdom and they have been using it in the last two years. The interviewee stated that traditional plan-driven development process is slow and not as effective as DSDM. The other interviewee stated that in a plan-driven development can only be successful when the customer’s requirements are fully understood.

One of the interviewees mentioned that there are 1-3 teams in a typical project with 3-5 people working in each team. The other interviewee mentioned that there are 1-2 teams with 1-5 people working in each team. Having smaller teams is an agile approach. It makes it easier to work together in smaller groups, rather than larger groups. This is because everyone in the group can participate in discussions and work efficiently. Working in smaller teams a normal feature of an agile-based process. It makes it more easier to work in and the efficiency of the has the potential capability of increasing.

One of the interviewees mentioned that they are using “timebox planning” for performance measurement. This is basically that they would have a well defined scope in an increment. The interviewee mentioned that this makes it easier monitor the project’s progress quickly.

The other interviewee mentioned that they are using the following methods as metrics for performance measurement:

- How many bugs are found
- How long it takes to fix a bug
- How many lines of code there are in a project
- How long it takes test each timebox
- If it is possible to meet the deadlines for each timebox

One of the interviewees mentioned that they need to be much better at reporting progress. They are collecting raw data for this, but they need to summarize this regularly.

5. RESULTS AND ANALYSIS
This section will analyze the data collected and relate it to the theoretical background.

5.1 Case Study 1 - Ericsson
According to the data collected from the interviews at Ericsson, they are using agile-based methodologies in their software development projects.

One of the interviewees mentioned that agile-based software development requires a lot of management and planning. Misra et al. (2009) wrote in their paper that planning is one of the factors for success (Misra et al. 2009). It is important to have a detailed project and risk management plan so that the project can be monitored in order to make sure that the project is on track.

One of the interviewee stated that they are using an agile-based process called Scrum, while using a ‘Waterfall’ approach in each sprint. Since in Scrum, all of the sprints are iterated, it is feasible to use a ‘Waterfall’ approach in each sprint. By using this approach, it gives a sense of structure and order in each sprint, while iterating them.

According to the articles in Computer Sweden (2010) and MKSE.com (2010), Ericsson have been using a methodology known as Streamline development as a framework (MKSE.com 2010) in their software development projects. The result of this change was that their development time decreased by 50% (Computer Sweden 2010). This, in turn, can increase the customer satisfaction because they will be able to deliver faster to the customer and the customer can give constructive feedback to the developers on what to improve and what they customers are expecting in their system. They stated that more product’s have been delivered (Computer Sweden 2010), therefore, it is possible to get more feedback from the customers. This is an important aspect of any development process. The customer satisfaction needs to be achieved since the developing organizations are going to deliver the system to the customer.

Dybå & Dingsøyr (2008) state that by implementing an agile way, the project tend to be delivered faster and increase the quality of the project to the customer (Dybå & Dingsøyr 2008). Layman et al. (2004) state that using the extreme programming (XP) process, the productivity increases (Layman, Williams & Cunningham 2004). Extreme programming is an agile methodology and, therefore, it is possible to state that using an agile methodology can increase the productivity, quality and reduce the time of the project.

According to the article in Computer Sweden (2010), Ericsson have small teams with short project lengths (Computer Sweden 2010). Both of the interviewees mentioned that they split up larger project teams into
smaller teams (7-8 people per team). It is often more difficult to manage larger teams, such as huge project teams as one of the interviewees mentioned. Therefore productivity might increase by having smaller teams. These smaller teams can focus on a narrow scope and then integrate their work with the other teams.

Shorter projects (three months, in this case at Ericsson) can also increase the productivity since there will be a narrow scope as a goal of the short project and it makes it easier to manage the project and make sure that the project succeeds (for example, in each sprint when using Scrum) with the goals and requirements set forth by the customer. This is a characteristic of an agile-based methodology.

Since there are shorter iterations, there can be more feedback from different people, such as the developers, project managers, and the customers. This is a characteristic of agile software development process. This means that there is more communication between the developing teams and the customers. Therefore, it is possible to have a higher level of satisfaction from both the development team and the customers. However, this does not mean that the quality will be higher. There is a potential that the quality could be higher. The quality depends on how the system has been developed according to the specifications customer’s requirements.

The article in the magazine (Computer Sweden 2010) states that by using this specific agile process there have been shorter development times, fewer changes and more deliverables, and developing the correct software according to the customers’ requirements (Computer Sweden 2010).

One of the interviewee stated that working globally can be a challenging task. This is because the people are not working in the same location, for example, the team in Sweden are working together with the team in China. It is always much easier to work in the same location, where it is possible to have face-to-face communication. There are various techniques to overcome this challenge, for example, to have videoconferencing and use collaborating tools, such as IBM Rational Team Concert, Microsoft Groove, Google Apps, etc. Time-zone difference can be another hinder for scheduling meetings. If something cannot be discussed in e-mail, the only solution that would be suitable would be to work early mornings or late evenings. However, this is not always possible. With agile, it is possible to incorporate these techniques in order to solve these techniques.

Monitoring progress is an important and rather difficult task. Metrics can be used to achieve this. Misra et al. (2009) mentioned in their paper that planning is very important. If there is a good risk management plan, then it is easier to monitor the progress of the project. It is important to keep track of the project’s progress in order to make sure that the project is progressing. This makes sure that the project is on track and the deadline has not been exceeded. This can lead to the project failing.

It is also important to have future goals. Based on one of the interviewees, the future goal is to learn more about Lean. This is a very important goal because if they learn more about Lean, then they will be able to be even more productive.

Ericsson has been using both agile processes and Lean very successfully. It is especially stated in the article in Computer Sweden (2010) magazine that Ericsson has reduced their development time by 50% (Computer Sweden 2010). Ericsson are using Lean as a framework, while using other agile-based methodologies in smaller projects, which makes sense. It is important to monitor all of the projects.

5.2 Case Study 2 - Erlang Solutions Ltd.

According to the data collected from the interviews, both organizations are using an agile-based software development process.

The interviewees at Erlang Solutions Ltd. mentioned that they are using the Dynamic Systems Development Method (DSDM). This is an agile-based software development process and mainly has an advantage when cost and time are the big factors (Wikipedia 2010b). This process is also similar to another agile-based process called Scrum. The DSDM can be split up into three steps: “pre-project, project life-cycle, and post-project” (Wikipedia 2010b). This method has been well established in the United Kingdom and is often much faster than a plan-driven development process.

Williams (2010) state that by using an agile way of developing software can actually implement more changes during a specific project (Williams 2010). This can be true because the agile software development process is very flexible. In a traditional ‘Waterfall’ method, it is very important to have an accurate plan over the different parts of the project, whether it is analyzing requirements or starting to implement the system. Since the customers can change the requirements, or request to add other functionalities to the system, it is therefore important to implement the ability to change during the project.

In order to have the ability to change the requirements more frequently often means that the customer is involved in the development process. However, having the customer involved a lot in the development process can have both positive and negative effects (Misra, Kumar & Kumar 2009). If the customer is involved more in the project, the positive effect would be that the customer give feedback more frequently, thus, have the capability of increased satisfaction. However, the negative effect would be that the customer might get too involved in the project and might not let the developers work for themselves. The team might have a difficult time to drive the project in the correct way (Misra, Kumar & Kumar 2009).
One of the interviewees mentioned that they need to be better at reporting progress. This is important because in order to improve, it is crucial that the organization understands where they need to improve. The success of the projects can depend on the progress of the project. If they are able to report the progress of the project, then they will be able to see which phase is slowing down the progress. They did mention that they were using “timebox planning” to make sure that they have progress in the projects. This is a way to make sure that there is a narrow scope with a specific deadline. This is used in agile-based software development processes and can be regarded as a successful way of making sure that the project has progress. This is because it lets the team know that they have a task with a narrow scope that has a deadline approaching.

In conclusion, it is possible to state that Erlang Solutions Ltd. are using various features that characterizes an agile way of working in a project. They are aware that they need to improve in progress reporting, however, according to the interviews, they have stated that they have seen more progress in agile-based process rather than with plan-driven processes.

5.3 Analysis of this Research

Based on the data collected from the interviews, it is possible to state that both organizations are using some form of agile software development process. Both organizations are using various features that characterizes agile. Furthermore, they have been successful in their software development projects using an agile-based process, while being quite satisfied with the software development process they are using. They do have future plans for improvements, and this is normal for any organization. It is, however, possible to answer that agile-based software development can definitely be recommended to organizations to use in their software development projects, and that agile can potentially benefit other software development projects. It is important to understand that there is no process that will guarantee success to all projects that use a specific process. This paper has shown how two organizations are using agile-based processes and they are quite satisfied.

It is also important to be critical on why agile-based processes should be recommended to organizations. Is agile software development process more successful? It is important to define what the term ‘successful’ means, as it can vary in different organizations. Success can mean finishing the project and delivering to the customers, while it can also mean to have maximum customer satisfaction. Both are valid statements, however, one might lean towards having more customer satisfaction, since the customers are the ones buying the product.

It is possible to state that agile can be considered positive and can definitely be recommended for using in software projects. Ericsson have been using Streamline development as a framework while using other agile methodologies, such as Scrum, in smaller projects. Erlang Solutions Ltd. are using a DSDM approach, which is an agile methodology. This method has been around the United Kingdom for quite some time.

6. CONCLUSION

This research is about illustrating the use of an agile-based software development process and if agile-based process can be recommended to organizations for software development process. Interviews have been conducted at two organizations in order to understand how organizations are using agile-based software development process. Both organizations that have been studied in this research are using an agile methodology in their software projects and are successful. There are a lot of benefits of using agile in comparison to a traditional plan-driven development process.

The main benefits of using an agile software development process is that it is a flexible method that can incorporate various changes in a project. These changes can be that the customers are wanting to change the requirements. Another possible reason can be that it is not always possible that all of the customer’s requirements are fully understood and it might take some time for the team to understand it fully. Therefore, it is important to be able to change a project plan and incorporate this change in a project. One of the interviewees mentioned that a plan-driven development (‘Waterfall’) process can be slow and not as effective as the Dynamic Systems Development Method (DSDM), which is an agile methodology. A plan-driven development process can be successful when the requirements are fully understood. Therefore, it is possible to conclude that agile can be positive for software projects and it can be recommended to use in software projects.

References
APPENDIX

A. INTERVIEW QUESTIONS A

A.1 At Ericsson

- What is your role at Ericsson?
- Are you using an agile development process for the different projects?
- If yes, which agile development process are you using?
- Why are you using this specific agile development process?
- How many people are working in a typical team?
- What type of metrics are used for performance measurement?
- Are there any projects are use the classic ‘Waterfall’ development process?
- Are these projects more successful than agile-based projects?
- Have you ever changed the software process recently?
- If yes, why?
- Are there any future improvements for the projects?
- If yes, please elaborate.

B. INTERVIEW QUESTIONS B

B.1 At Ericsson

- What is your role at Ericsson?
- Are you using an agile development process for the different projects?
- If yes, which agile development process are you using?
- Why are you using this specific agile development process? Any specific reason? Why not a plan-driven development process?
- How many teams are there in a typical project?
- How many people are working in a typical project?
- How many people in a typical team?
- What roles are there in a typical team?
- What type of metrics are used for performance measurements?
- Have you ever change the software process recently?
- If yes, why?
- If no, why?
- Which projects have been more successful?
- Why?
- Which projects have been less successful?
- Why?
- According to literature reviews, an agile development approach has been seen to increase productivity and quality. Are there any projects that are running using a traditional approach, such as the ‘Waterfall’ development process?
- If yes, are these projects more successful than other agile-based projects?
In your opinion, what is the benefits of agile software development in comparison with traditional ‘Waterfall’ development?

Are there any future improvements to the projects?

If yes, what? Please elaborate.

C. INTERVIEW QUESTIONS C

C.1 At Erlang Solutions Ltd.

What is your role at Erlang Solutions Ltd.?

Are you using an agile development process for the different projects?

If yes, which agile development process are you using?

Why are you using this specific agile development process? Any specific reason? Why not a plan-driven development process ('Waterfall')?

Are there any project that are running a traditional approach, such as the ‘Waterfall’ development process?

If yes, are these projects more successful than other agile-based projects?

How many teams are there in a typical project?

How many people are working in a typical project?

What type of metrics are used for performance measurements?

Have you ever change the software process recently?

Which projects have been more successful?

Why?

Which projects have been less successful?

Why?

D. INTERVIEW QUESTIONS D

D.1 At Erlang Solutions Ltd.

What is your role at Erlang Solutions Ltd.?

Are you using an agile development process for the different projects?

If yes, which agile development process are you using?

Why are you using this specific agile development process? Any specific reason? Why not a plan-driven development process ('Waterfall')?

Are there any project that are running a traditional approach, such as the ‘Waterfall’ development process?

If yes, are these projects more successful than other agile-based projects?

How many teams are there in a typical project?

How many people are working in a typical project?

What type of metrics are used for performance measurements?

Have you ever change the software process recently?

Which projects have been more successful?

Why?

Which projects have been less successful?

Why?

Why?

Which projects have been less successful?

Why?