



Analysis of the Agile Deployment

A thesis work based on the literature study and empirical experience.

Yi-Lin Chen

Thesis work in Software Engineering and Management (Master Program)

Report No. 2009:073

ISSN: 1651-4769



ABSTRACT

Nowadays, software project management is becoming more and more important since a project needs an organized plan to follow through. There are two famous models in this area, which are Waterfall Process Model and Agile Software Development. The concept of the Waterfall Process Model is that the requirement analysis has to be done in the beginning phase, whereas, the Agile Software Development emphasizes that the requirements are changeable throughout the process. Thus, it seems that Agile Software Development is paid more attention to among software development companies because of the advantage of flexibility and the allowance of requirements change. However, the Agile Software Development is not suitable for every type of projects and it requires an experienced manager. Therefore, some questions about the Agile Software Development may occur. For example, is the Agile Software Development able to fulfill project managers' needs? Do project managers encounter any problems while executing the Agile Development Framework? In my master thesis, the usage of the Agile Software Development used among various companies will be revealed. Further, suggestions for those companies with the benefit of literature study would also be brought up since the companies may

face some unexpected problems during using the Agile Development Framework.

Acknowledgement

Firstly, I have to thank my supervisor, PhD Elisabeth Saalman (IT University, Chalmers university), who has given me many helpful suggestions during my writing of this thesis. Secondly, I would like to give my thanks to PhD Helena Holmstrom (IT University) and my mentor, Joakim Stolt (IBM, Denmark) who inspired me to have this topic. Joakim Stolt also suggested to me the direction of finding companies as well as helping me find contact information for my interviews. I also want to thank PhD Miroslaw Staron (IT University) for giving me many suggestions on my thesis writing improving the overall quality. In the end, I want to thank the interviewees. I wish to thank them for their help so that I could collect empirical data regarding the use of the Agile Software Development.

Keywords

Project Management, Agile Software Development, Iterative Process Model, Software Development Process, Global Software Development, Project Life Cycle, PDCA.



Contents

Definitions	3
1. Introduction	3
1.1 Problem Area	4
1.2 Purpose	4
1.3 Project Management Tool	5
2. Agile Software Development	5
2.1 What is Agile Software Development	5
2.1.1 History	5
2.1.2 General Understanding of the Agile Software Development	6
2.1.3 Why the Agile Software Development is important- see what Survey said	9
2.1.4 The Agile Project Lifecycle	11
2.2 Agile and Global Software Development	12
2.3 Methods of the Agile Development Framework	13
3. Research Method	15
4. Findings- Interview	16
4.1 Introduction	16
4.2 Interview Data Analysis	16
4.2.1 Volvo IT, Sweden	16
4.2.2 IBM, Taiwan	17
4.2.3 Ericsson, Netherlands	18
4.2.4 Keys Taken Out From Interview Data	18
5. Discussions	19
6. Conclusions	24
7. References	25
Appendix- Interview Question	28



Definitions

For facilitating reading this thesis, some terms are defined in advance to clarify understanding of this thesis.

Waterfall Process Model

Waterfall Process Model is a predictive method for software development project. The project process is based on documentation. For example, the activities consisting of requirement elicitation, analysis, design, code and test have to be well-defined in a document in the early phase. The project process must follow those activities step by step strictly. Thus, the Waterfall Process Model is said to fit large projects more since they are usually more complicated and it will be more organized to have a concrete and organized plan to follow through. However, the Waterfall Process Model does not allow the changing of requirements; thus, it is also seen as slow and illogical.

Backlog

In the Agile Development Framework, tasks are divided into several backlogs. Customers prioritized those backlogs and decide which backlogs will be processed in an iteration.

Sprint

Sprint is mostly used in SCRUM. A sprint is an iterative cycle consisting of a set of features comes from backlog. A sprint usually takes two to four weeks. A demonstration of work effort and the use of the software are resulted at the end of a Sprint.

Natural Language

Definition of a natural language is a language that is spoken or written which is understandable to everybody.

1. Introduction

Nowadays, software project management is becoming more and more important since a project needs an organized plan to follow through. There are two famous process models in this area, which are the Waterfall Process Model and the Agile Software

Development. The concept of the Waterfall Process Model is that the requirement analysis has to be done in the beginning phase, whereas, the Agile Software Development emphasizes that the requirement is changeable throughout the process.

Thus, it seems that using the Agile Software Development is becoming a trend for software development companies in order to improve the software process. The Agile Software Development was developed to solve the Waterfall's weakness. Differing from the Waterfall Process Model, the Agile Software Development emphasizes its flexibility. For example, requirements are allowed to change in an Agile project. A face to face communication also takes an important place in the Agile Software Development since it requires a quick feedback. However, some questions may emerge, such as, is every Agile team satisfied with the Agile Software Development? How do companies implement the Agile Software Development? Do project managers encounter any problems in using the Agile Software Development? Can we make it better?

This master thesis is carried out based on the literature survey and empirical information. The goal of this master thesis is to discover the way companies use the Agile Software Development, focusing specifically on the Global Software Development area by interviewing real companies and to generate suggestions based on the suggestions found in literature studies.

General understanding of the Agile Software Development such as its principle and concept is given in the Chapter 2. Some industrial survey of using the Agile Software Development to support why the Agile Software Development is the best process model for software development is also given in the same chapter. The most used Agile Development methods are also listed and defined briefly in the Chapter 2. Chapter 3 is devoted to the research method I have used for this thesis. This chapter elaborates on the approaches and strategies I used in this thesis. Furthermore, the interview can be found in Chapter 4, Findings. In the fourth Chapter, questions such as how the companies think about the Agile Software Development and the difficulties



in using the Agile Software Development are revealed. Chapter 5 is reserved for the discussions, which is to give solutions based on the interview results. For example, the answer of how to adapt the Agile Development Framework into Global Software Development, how to adopt the Agile Development Framework into a Waterfall project, and how to start up an Agile project with PDCA are addressed in the fifth chapter. The final chapter contains my own opinion and conclusions. In the final chapter, I also give the suggestions for Global Software Development based on my previous work experience as well as tips for better implementing the Agile Software Development for fresh project managers.

1.1 Problem Area

The thesis was carried out based on two questions. One was how companies use the Agile Software Development. The Agile Software Development highlights that a project should be developed incrementally and iteratively. Tasks in a project are divided into some backlogs and are prioritized. However, the way that companies use Agile Software Development may be different although the concepts of the Agile Software Development are the same. Thus, how companies use the Agile Development Framework is explored in the thesis. The other question was to see if literature studies can solve companies' problems while implementing the Agile Development Framework.

First of all, the goal of the Agile Software Development is to assist companies to take full advantage of the customer value of the delivered software product. However, the Agile Software Development is only suitable for a company working in single-project contexts because an Agile team must work closely with customer in order to produce high-value and high-quality software efficiently with rapid iterations and frequent feedback. On the other hand, the set-up in large organizational context is more complicated since companies developing large, complex systems have to control not only individual projects and products but also their interaction with various large customers who often experience long product life cycles (Kettunen, 2007).

As the first question mentioned, companies may experience some difficulties while implementing or executing the Agile Software Development. Thus, I present the suggestions for companies from the literature study in this master thesis. Moreover, actual experience is sometimes different from literature study; thus, the benefit of literature study may be helpful for them.

The problem area for this master thesis can be summarized into three core questions listed as below:

- Do project managers encounter any problem while using the Agile Software Development?
- Is there any difference in using the Agile Software Development among companies located in different countries?
- Could the literature study provide the companies some better suggestions or solutions?

1.2 Purpose

Currently, big companies such as Ericsson are attempting to implement the Agile Software Development into their software development projects. Thus, the goal of this thesis was been to sort out the data from companies' actual experience in using the Agile Software Development and compare those experiences with the literature study. Answers regarding empirical usage would be investigated by interviewing software development companies. The method used within this thesis was been to interview three companies located in different countries to discover the way they used the Agile Software Development. Those actual companies' experiences were also been compared with the expected results presented in the literature and to see if there were any suggestions arising from the benefit from the literature study.

Besides, the expected result for the topic is the ability to analyze the data from interviews and sort it to become useful information. Furthermore, how to apply knowledge gained from literature into future jobs is another expectation.

1.3 Project Management Tool

One of the Agile Software Development has been used during the project. SCRUMY is an online solution for a project management tool based off of SCRUM (see Figure 1.1). It takes the concept of “post it” for creating a task. The status of task can be changed by dragging it.

Furthermore, this master thesis was stored both in a local computer and a SVN repository. Due to the fact that the working environment can vary and this master thesis document would be modified constantly, SVN was the best solution for tracing the change of the documents.

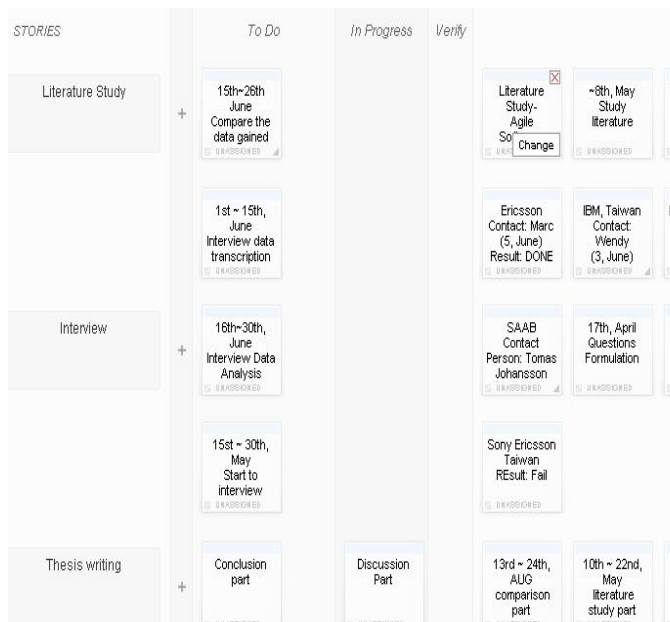


Figure 1.1 SCRUMY.
(Source: <http://www.SCRUMy.com/eline>)

2. Agile Software Development

The chapter talks about general understanding of the Agile Software Development.

Differing from other software engineering projects, a software project management focuses on the development phase. Some significant activities such as requirements¹, design, implementation, verification, and maintenance are paid more

¹ Requirement contains functional and non-functional requirements, user requirement and system requirement.

attention to. A software product is usually invisible (Ian 2007, p.93); thus, software development projects are highly unpredictable and that is why so many software development projects fail to meet expectations (Water, 2008).

The reason for why the Agile Software Development is needed for software development can be explained as below:

“If a predictive method like Waterfall is to avoid pain of problems, then we can say that Agile was developed to solve the pain of problems.”

-M. Scott Peck, The Road Less Traveled.

2.1 What is Agile Software Development

2.1.1 History

The Agile Software Development is a group of software process methodologies for small or medium organization. The Agile Software Development is also known as Agile Development and usually used in a software development project; it emerged in mid-1990s to react against “Heavyweight” methods such as the Waterfall Process Model (Highsmith, 2001). Initially, the Agile Software Development was called “Lightweight Methodology”. The Agile methods grew up separately for a number of years, until 2001. In February, 2001, a group of their leading proponents met at Snowbird, Utah. They agreed to the name "Agile Methodologies" and created the Agile Manifesto² and the principle. Later on, some members of this group formed The Agile Alliance, a non-profit organization that promotes the Agile Development (Highsmith, 2001).

The manifesto states (Agile Manifesto, 2001):

“We are uncovering better ways of development software by doing it and helping others to do it.

Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive

² Manifesto for the Agile Software Development <http://agilemanifesto.org/>

documentation

- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan”

“The Agile principle (Agile Manifesto, 2001):

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

adjusts its behavior accordingly.”

2.1.2 General Understanding of the Agile Software Development

Generally, agility is defined by ability which is able to be flexible and adaptable to change (Knoernschild, 2009). The idea of the Agile Development Framework is to create a pain-free working environment for those small, co-located, self-organized teams in order to assist companies to take full advantage of the customer value of the delivered software product (Kettunen, 2007).

On an Agile project, developers work closely with their customers to understand their needs, they are placed in a pair to implement and test their solution, and the solution is shown to the customers for quick feedback (Ambler, 2006). Therefore, the business contract will not become a barrier between customers and developers, but a platform to help customers and developers work together (see Figure 2.1).

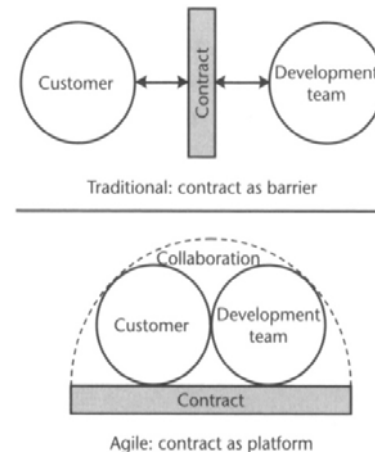


Figure 2.1 Comparison of contracts between traditional and the Agile Development Framework. (Source: Koch, 2005)³

Moreover, the book, *Managing Agile Projects*, defines that the Agile Software Development is the work of energizing, empowering, and enabling project teams to rapidly and reliably deliver business value by engaging customers and continuously learning and adapting to their changing needs and environments (Augustine, 2007).

³ Alan S. Koch cited the diagram in the report “*Agile Software Development: Evaluating the Methods for Your Organization*”, 2005



In the Agile Development Framework, user requirements (as known as User Story) are written from users' perspective, elaborating on what users want to do with a feature of the software. The composition of User Stories is usually considered by business value, the story point, and other factors such as risk. Thus, using User Stories is a simple and brief way to express those user requirements. A User Story is usually composed in natural language. The construct of a User Story is as follows: As a [user role], I want to [goal], so I can [reason].

A User Story should focus on Who, What and Why of a feature, but not How (Water, 2008). One of the examples is the Google Chrome project⁴.

The concept of the Agile Software Development can be summarized as below (Knoernschild, 2009):

Eliminate Waste: The Agile Development Framework advocates a "barely sufficient" approach to plan, process, and control software development process (Augustine 2007, p.23). "Barely sufficient", in other words, is to find the simplest things to meet needs of requestor instead of wasting unnecessary resources.

Sustainable Pace: The Agile Development Framework requires a daily meeting. All team members have to report what they have accomplished and what they are going to do in the meeting so that the progress of the project can be traced.

Intense Collaboration: Unlike the traditional approach, which relies on documents, the Agile Development Framework requires a daily face to face communication with customers and co-workers to understand and fulfill their requirements.

Frequent Delivery: Frequent delivery offers incremental business value to customers. Customers experience the growth of the system and obtain additional insight to how requirements are planned by interacting with the system early. Thus, frequent delivery is one important way to seek feedback on the quality of the application.

Continuous Feedback: The Agile Development Framework was invented to achieve customer's value. Thus, continuous feedback is demanded in order to inspect if the development team is well aligned with business objectives.

Include Change: Differing from the traditional approach, change of requirements is considered in the Agile Development Framework since remaining adaptable is a key to building a trusting relationship with customers. Furthermore, prioritizing features, exploring and explaining risk help both of team members and customers understand the consequence of change.

Empowerment: The Agile Development Framework also pays attention to the team working atmosphere. It is very important to set-up a well-organized and positive team. Therefore, encouraging team members is highly required.

Features of the Agile Development Framework are described as below (Augustine 2007, p.21):

Iterative and Incremental development: As Figure 2.2 shows, plans, requirements, design, implementation, deployment and testing are developed incrementally through iterations. Each iteration usually takes two to four weeks. Moreover, problem hunting, scope solving, feedback collection should be done at the end of each iteration. In addition, features and tasks are also inspected and tracked within each iteration. (Augustine 2007, p.21)

⁴ The Google chrome is a new web browser developed by Google.
<http://chrome.blogspot.com/>

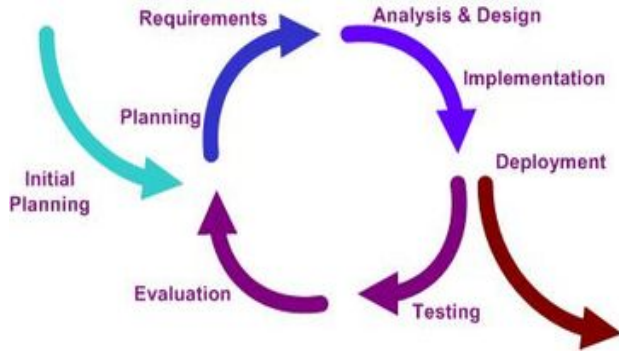


Figure 2.2 Iteration in the Agile Development Framework. (Source: Unknown, Wikipedia)⁵

Establish and Changing Requirements: In traditional software development process model, identification and analysis of requirements for the system are documented within an agreement for customers and the development team in an early phase of a project. Once this agreement has been reached, the requirements are not allowed to change. In contrast, the Agile Development Framework allows both customers and developers to change the requirements throughout the project, but only the customers have the authority to approve, disapprove and prioritize the ever-changing requirements (Koch, 2005), see Figure 2.3.

In addition, the requirements can be re-prioritized anytime. Once the priority changed, the new higher requirement will be pulled up to the top of the stack (Ambler 2004, ch.4), see Figure 2.4.

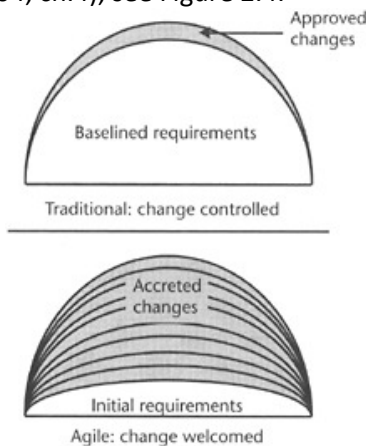


Figure 2.3 Difference of requirement change between traditional process model and the Agile Software Development. (Source: Koch, 2005)³

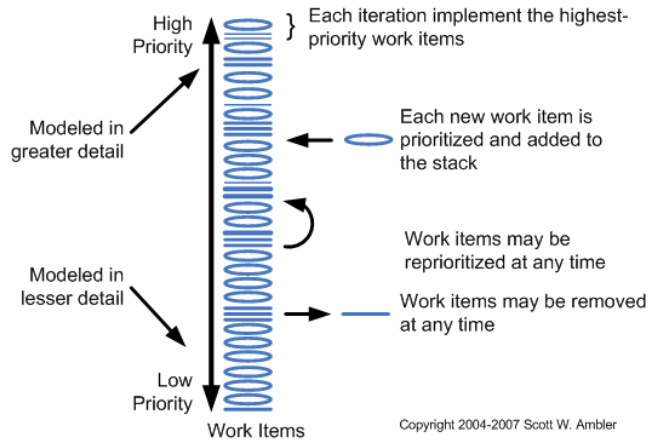


Figure 2.4 Process of Agile requirements change management. (Source: Ambler, 2004)⁶

Use Backlog: In the Agile Development Framework, tasks are divided into small chunks (also known as backlogs) to manage complexity and to get quick feedback.

Prioritize Backlog: In an Agile team, backlogs are prioritized by customers and only higher prioritized backlogs (top 2 or 3) will be processed in an iteration (Figure 2.5). Once the backlog prioritizing works are done, the prerequisites for calling the first Sprint Planning Meeting will be embraced (Stevens, 2008). Moreover, unfinished backlogs are inspected and re-prioritized at the end of each iteration in order to decide which backlogs will be processed in the next iteration.

⁵ The diagram is cited on Wikipedia website: http://en.wikipedia.org/wiki/Iterative_and_incremental_development. [Accessed 10, May, 2009]

⁶ Scott W. Ambler cited in the book "The Object Primer", Ch4. 2004.

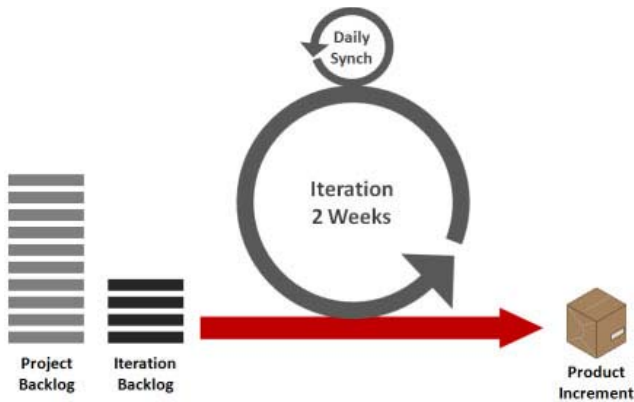


Figure 2.5 Agile Backlog Interface. (Source: dtsagile, <http://www.dtsagile.com/Agile>)

Pairing Programmer & Self-Organization: In an Agile team, a less experienced programmer is paired with an experienced one in order to share knowledge and teach each other. Moreover, each Agile team is self-organized. Team members are self-organized by accomplishing tasks with their co-workers from backlogs.

Face to Face Communication: The Agile Development Framework emphasizes face to face communication. It promotes holding a short daily meeting. In the meeting, team members have to report their project progress. Questions such as what they have done, what they are working on, and what they will do tomorrow should be answered in daily meeting.

2.1.3 Why the Agile Software Development is important- see what Survey said

Differing from the traditional process model emphasizing the measurement of success of conformity to predictive plans, the Agile Software Development emphasizes responsiveness to change. For example, the delivery of working software is the most important factor to lead a software development successful since in the Agile Development's view, metrics such as cost variance, schedule variance, requirements variance and task variance is virtually meaningless (Ambler, 2008).

According to the result of the DDJ⁷ 2007 Agile Adoption Survey, the Agile Development Framework has become a mainstream for software development. The survey indicates that 69% of respondents said that organizations were doing one or more Agile projects and 85% of them were even doing more than two.

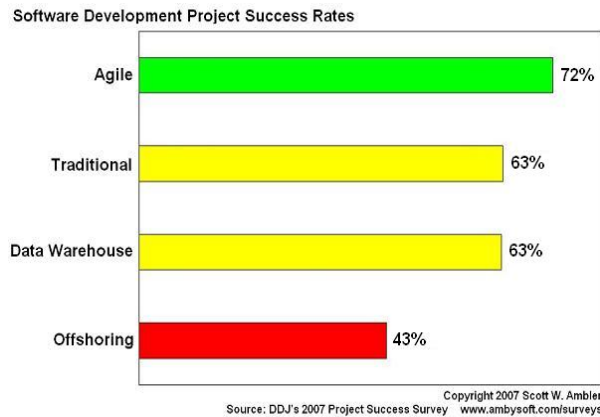


Figure 2.6 DDJ 2007 Agile Adoption Survey Result: Rate of Successful Agile projects. (Source: Ambler, 2008)⁸

Additionally, the 3rd Annual Survey 2008, The State of Agile Development, shows that the users using the Agile Development Framework thought that "Accelerate time-to-make" and "Enhanced ability to manage changing priorities" were the top two main reasons that they were concerned about adopting the Agile Development Framework. Moreover, both of DDJ 2007 Agile adoption Survey (see Figure 2.6) and the 3rd Annual Survey 2008, The State of Agile Development (see Figure 2.7), indicate that more than 50% respondents thought that they have had 90% to 100% of successful Agile projects.

Table 2.1 shows that almost half of respondents thought that the Agile methods have improved their projects from many aspects such as project visibility, productivity, software quality, and development process...etc.

⁷ Dr. Dobb's Journal is an organization leading the computer press in covering practical technology since 1976. <http://www.ddj.com/>

⁸ Scott W. Ambler cited the diagram in "Answering the 'where is the Proof That Agile Methods Work' Question", 2008

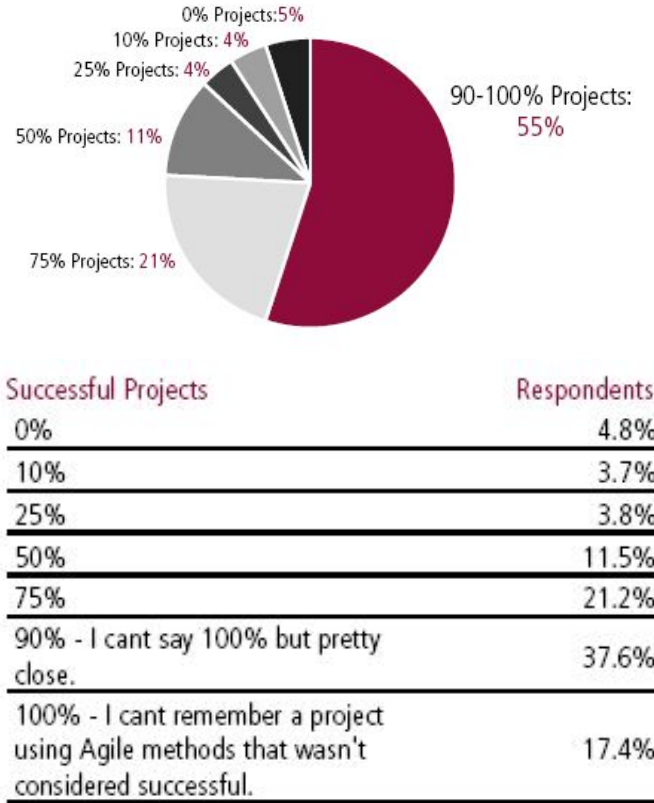
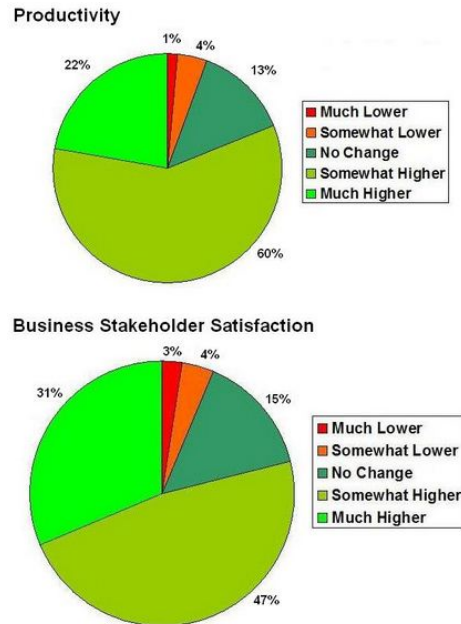


Figure 2.7 3rd Annual Survey 2008 Survey result: Rate of successful Agile projects. (Source: VersionOne Inc, 2008)⁹

	Significantly Improved	Improved	No Benefit	Worse	Much Worse
Improve Project Visibility	41.5%	41.8%	15.0%	1.2%	0.4%
Increase Productivity	23.6%	50.5%	15.6%	2.1%	0.3%
Enhance Software Quality	24.0%	44.3%	19.9%	2.8%	0.4%
Enhance Ability to Manage Changing Priorities	50.5%	42.1%	6.6%	0.6%	0.2%
Accelerate Time-To-Market	23.6%	41.3%	21.6%	2.4%	0.2%
Reduce Risk	16.6%	48.0%	23.6%	2.1%	0.3%

Table 2.1 3rd Annual Survey 2008 Survey Result: Value of implementing Agile Practices. (Source: VersionOne Inc, 2008)⁸

Furthermore, Figure 2.8 indicates that compared with traditional approaches, most of respondents thought that the Agile methods are more efficient.



⁹ The diagram is cited in the report "3rd Annual Survey 2008, 'The state of Agile Development'" by VersionOne Inc, 2008.

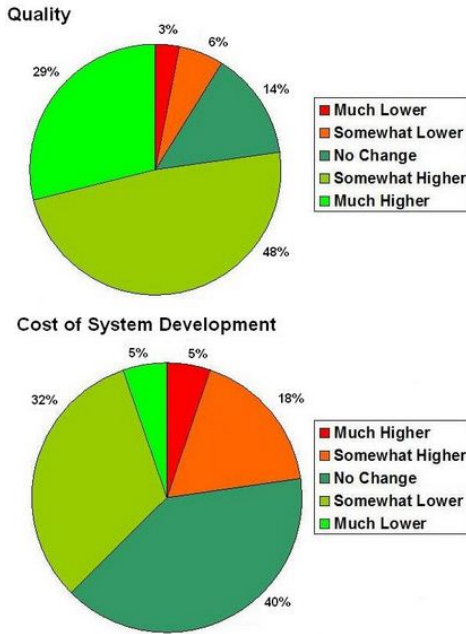


Figure 2.8 DDJ 2008 Agile Adoption Survey Result: Comparison of effectiveness. (Source: Ambler, 2008)⁷

According to the survey result shown above, we know that using the Agile Development Framework is becoming a trend within software development companies. Most of respondents gave a positive view to the Agile Development Framework. Moreover, comparing with traditional approaches, an Agile project has more possibilities to success.

Consequently, the Agile Development Framework brings software projects a great success because of the shorter feedback cycles.

For instance, Figure 2.9 shows the relationship between the cost and feedback cycle. Due to the fact that the Agile Development Framework requires a daily meeting for a frequent feedback and delivery, defects are detected in the early stage; thus, the cost curve stays in a lower position, whereas, defects are usually found late in traditional approaches since they rely on a predictive plan.

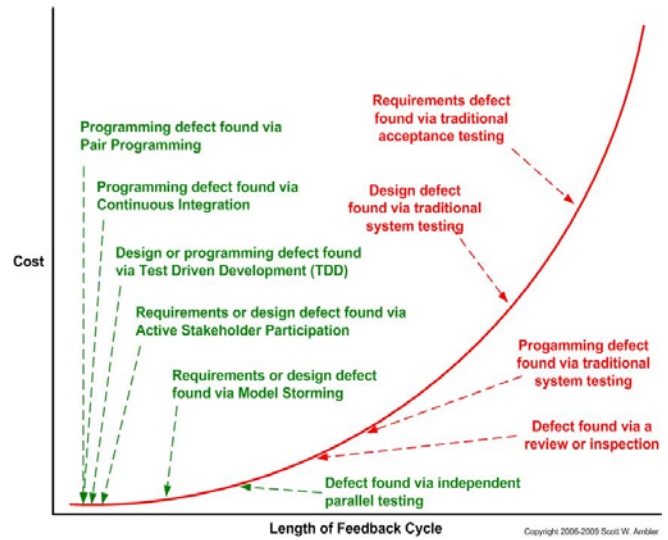


Figure 2.9 Comparison of Feedback cycles with traditional approaches. (Source: Ambler, 2008)⁷

2.1.4 The Agile Project Lifecycle

As I mentioned above, the Agile Development Framework is an iterative, incremental, and collaborative methodology for software development project. The Figure 2.10 shows that the Agile Software Development Lifecycle (ASDL) starts from an initial plan and ends with deployment. Each iteration consists of planning, requirements, analysis and design, implementation, deployment, testing, and evaluation (see Figure 2.2).

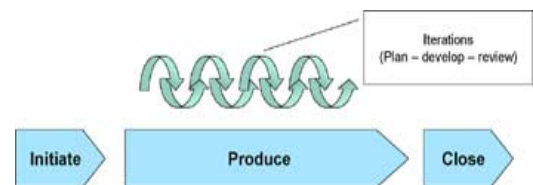


Figure 2.10 Agile Project Life Cycle. (Source: Oktaba & Piattini, 2008)¹⁰

If we look into the ASDL in detail, the ASDL is comprised of six phases: Iteration -1, Iteration 0 (Warm up), Construction, Release (End Game), Production, and Retirement (Ambler, 2006), see Figure 2.11.

¹⁰ Hanna Oktaba & Mario Piattini cited the diagram in the book "Software Process Improvement for Small and Medium Enterprises" ch.4, 2008.

Agile System Development Lifecycle

Copyright 2005-2009
Scott W. Ambler

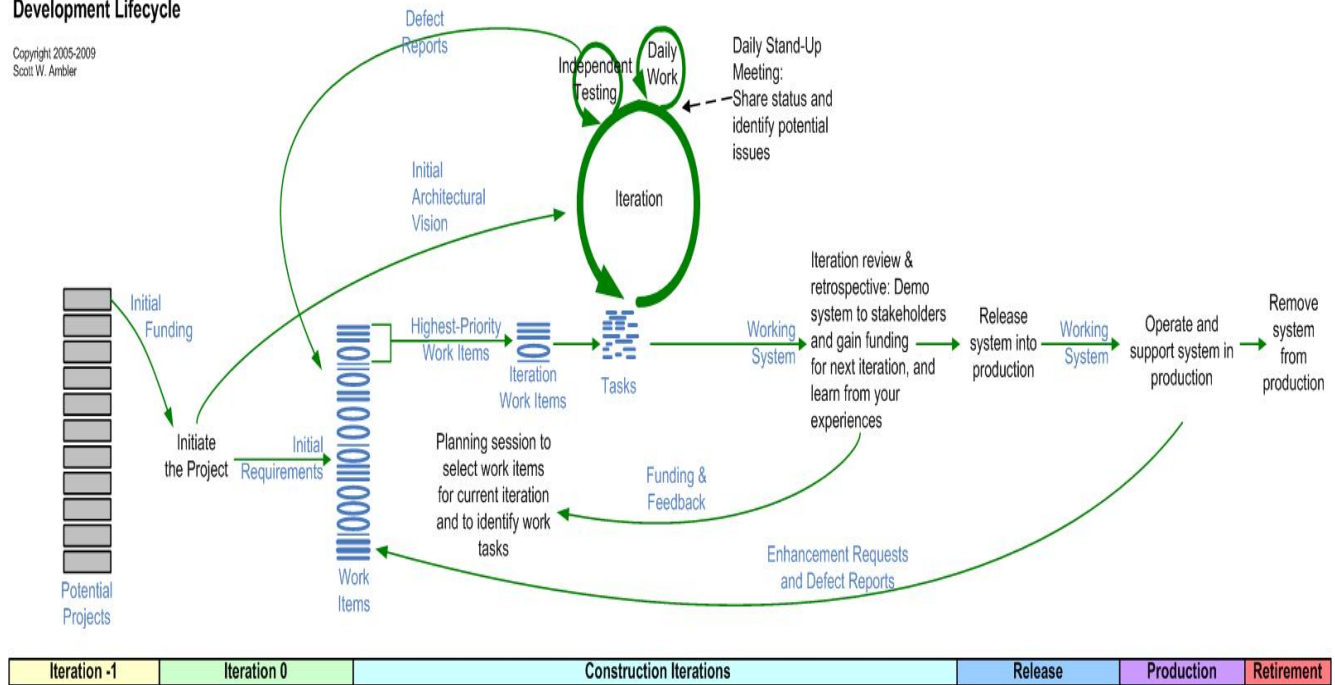


Figure 2.11 ASDL Diagram. (Source: Ambler, 2006)¹¹

Each phase is described as:

Iteration -1: Iteration -1 is devoted to pre-project planning. Activities such as defining the business opportunity, identifying and assessing the possibility for the project are involved into this phase.

Iteration 0: After completing the iteration -1, the environment setup, team formulation, support and funding obtainment, and so on taken in the iteration 0, also known as project initiation.

Construction iteration: During construction iteration, a highly collaboration between team members and customers is required. Moreover, works regarding prioritized functionality implementation, system analysis and design, regular working software delivery, and verification are also embraced into this phase.

Release: In this phase, activities such as testing, documentation finalization, and system deployment into production have to be done in this phase.

Production: When a project turns into production phase, the goal is set as keeping systems useful and productive after deploying them to the user community. In other words, the system should be kept running and help users to use it.

Retirement: In the end of the life cycle, the iteration goes to retirement phase which is to update enterprise models and remove the final version of the system-data conversion if the system has become obsolete or can be complete replaced (Ambler, 2006).

2.2 Agile and Global Software Development

The section talks the relationship between the Agile Development Framework and Global Software Development.

The relationship between the Agile Development Framework and Global Software Development is well defined in the work of Fox (2007):

“Many software development organizations are intending to implement the Agile Development

¹¹ Scott W. Ambler cited the diagram in “The Agile System Development Life Cycle”, 2006.



Methodologies in order to take advantage of the cost, quality, and time-to-market benefits commonly achieved with this approach. Meanwhile, these organizations are moving software development offshore to take advantage of greater scalability and “round the clock” development cycles. However, in combining these two efforts, the highly collaborative nature of the Agile model is tested as teams are faced with cultural challenges and necessary work habit shifts.”

Global Software Development (GSD), including outsourcing, subcontracting, and partnerships has quickly become a common strategy for software development organizations today (Passivaara, et al., 2008). GSD means establishing development teams in multiple countries (also known as onsite team and offshore team) in order to fulfill the concept of working round the clock. An advantage is that the onsite team codes all day, updates the offshore team and then goes home. Meanwhile, the offshore team takes over and works on continuing programming (Bhandari & Veeramuthumoni 2009). Thus, the project is able to be worked on twenty-four hours per day.

Implementing GSD also helps companies accumulate experience in developing software globally. Moreover, it also brings the significant benefits such as cost savings, flexibility, access to key skills and experience, and quicker time to market to organizations (Fryer & Gothe 2008).

Nevertheless, it seems there are some conflicts between the Agile Development Framework and Global Software Development. For example, the primary communication mechanism the Agile Development Framework emphasizes is to meet face to face with the aim of getting frequent feedback. However, this seems difficult to achieve in a globally distributed project because of distance problem. Moreover, due to the fact that the development teams are set up in multiple locations in a GSD project and this does not seem to match the concept of the Agile Development Framework which is to work co-located.

In contrast, the Water Process Model will not suffer the communication difficulties with offshore teams because it uses documents as the primary communication method. In other word, it means that the communication has already taken all the damage from lack of direct contact, so the offshore effect is less noticeable. On the other hand, the Agile Development Framework tries to re-establish the direct contact in order to improve communication (Fowler, 2006). Moreover, comparing to the Waterfall Process Model, the Agile Development Framework provides the visibility into the team’s progress and the high level of communication and collaboration with the offshore team, and this could be considered as the better solution for the globally distributed projects (Fox, 2007). Thus, even if the Agile Development Framework has these communication obscurities, it is still said a better than a documentation-driven approach by Fowler (2006).

However, although the Agile Development Framework has been promoted in literatures, actual software development organizations still seem to suffer from an agile offshore software development implementation. In the following chapter, a real relationship between the Agile Development Framework and Global Software Development will be explored by interviewing actual software development companies and solutions will be addressed in chapter 5.

2.3 Methods of the Agile Development Framework

Some of the most used Agile Methods are introduced briefly in this section.

Adaptive Software Development (ASD)

ASD is a software development process that grew out of rapid application development work by Jim Highsmith and Sam Bayer. ASD focuses on continuous learning and adaption to the emergent state of the project provided by a repeating series of speculate, collaborate and learn cycles. An ASD life cycle is divided into three parts: Speculate, Collaborate, and Learn and based its six characteristics which are Mission Focused,



Component Based, Iterative, Timeboxed, risk Driven, and Change Tolerant (Highsmith, 2000).

Dynamic System Development Method (DSDM)

DSDM was created in the mid-1990s. The DSDM has its roots in an iterative-incremental process model that uses prototyping at each stage of development, also known as Rapid Application Development (RAD). A DSDM project is worked based on its nine principles (Krebs 2008, ch.2):

- Active user involvement
- Addressing business needs
- Base-lining of high-level scope
- Communication and collaboration among all stakeholders
- Frequent delivery
- Team decision making
- Integrated testing
- Iterative-incremental development
- Reversible changes throughout development

Extreme Programming (XP)

XP was developed by Ken Beck, Ward Cunningham and Rom Jeffries during the 1990s. XP is one of the most adapted Agile Methods in the high-technology industry. The XP highlights the feature of pair programming and test driven development. Moreover, it is often seen as an Agile engineering process although it provides planning practices for project management (Krebs 2008, ch.2).

Feature-Driven Development (FDD)

FDD is a client-centric, architecture-centric, and pragmatic software process. It was first introduced in 1999 via the book Java Modeling in Color with UML. Differing from other the Agile Methods, the FDD was first applied in an 18 month long middle-large project containing 50 persons. A FDD life cycle includes the following phases: Develop an Overall Model, Build a Features List, Plan by Feature, Design by Feature, Build by Feature (Ambler, 2003).

Lean Software Development (LSD)

LSD, created by Bob Charette (Krebs 2008, ch.2), is the application of lean principles to the craft of software development (Norton, 2005).

LSD is a strategy-oriented approach that considers the expenditure of resources for any goal. Software development is based on the goals of one-third the time, one-third the budget, and one-third the defect rate (Highsmith 2002, p.285).

The LSD principles stated by Norton (2005) are:

- Eliminate Waste
- Amplify Learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity in
- See the whole

SCRUM

Same as XP, SCRUM is one of the most used Agile Methods. It is a lean approach to software development. In SCRUM, development is structured in iterative cycles of work called Sprint. An Iteration of work usually takes 2 to 4 weeks. Team works on prioritized tasks by customers during each iteration and at the end of each Sprint, a potentially shippable product should be delivered (*SCRUM Alliance, n.d.*).

SCRUM was created by Ken Schwaber and Jeff Sutherland in the 1990s. SCRUM is usually used as an Agile project management method rather than an Agile process (Krebs 2008, ch.2). A short daily meeting in order to give team members a quick update is required in the SCRUM framework.

IBM Rational Unified Process (RUP)

RUP is also listed as one of the Agile Methods in the work of Krebs (2008, ch.2). RUP was invented by IBM in 2003. It provides industry-tested practices a comprehensive process framework for software and systems delivery and implementation for effective project management. In RUP, users can customize the project process to meet their unique demands by



selecting and deploying the process elements they need for their projects (IBM, n.d).

3. Research Method

The focus of this thesis is to interview companies to obtain the empirical data of their use of the Agile Software Development Framework. The structure of this thesis follows the structure proposed by Berg (2007), which are: the title, the abstract, the introduction, literature review, research method, findings, discussion, and conclusion. This section is composed by following the structure found in the book: *Qualitative Research Methods for the Social Science* (Berg, 2007). The reference citation follows the "Harvard System of Referencing Guide".

The term "Waterfall Process Model" is as it was defined in the work of Ian (2007), the term of "Agile Software Development" came from the Agile Manifesto (2001), and the "Global Software Development" term can be found in the work of Conchúir, et al (2005). The research method focuses on two parts, literature survey and interview.

First of all, for understanding the Agile Development Framework, I conducted literature surveys. The source of literature came both from books and papers found on internet. The related books were obtained from the library at Chalmers University as well as from an online library system called Book24x7. Moreover, due to the fact that a book usually takes two years to prepare in order to be published, the information might be old. Thus, the focus on literature survey was based on the papers found via the internet since the information on papers was probably issued in the same year as it was published. The source of papers was from the internet and the Academic Research Library database, which was also in the library system at Chalmers. Due to the fact that there are too many general findings within literature regarding the Agile Development Framework, the choosing of literature focused on three aspects: the Agile Development Framework practices and implementation, the relationship between the Agile Development Framework and Global Software Development, and survey of the use of the Agile Development Framework.

After accomplishing the literature study, interviews were conducted. Interviewing companies is the best way to get data of actual experience in using the Agile Development Framework since I needed to collect some information containing the practical experiences to compare with. The selection of companies was targeted at companies located in different countries because different cultures may affect the way that companies use the Agile Development Framework. As the work of Berg (2007) specified within the research methodology framework, I identified the subjects to be members in software development teams. The members have used the Agile Development Framework in their teams and worked in different countries. The approach of contacting those members (interviewees) was to find companies using the Agile Development Framework and the members were contacted by E-mail and Telephone. Company searches were based on internet search by typing the key words as well as via recommendation. It was very difficult to find companies willing to be interviewed so some interviewees were recommended and introduced by my mentor, Joakim Stolt, who works at IBM.

The data were collected by interviewing the members. All interviewees also gave me consent to use the information I gained from them into this thesis. The interviews were organized by face to face and telephone interviews. A mobile phone was utilized while holding a face to face interview in order to record the content of the interview. An IM tool, Skype, was also used for holding a telephone interview. The interview question was divided into five themes (see Chapter 4 and Appendix) and the interviews were conducted based on those five themes. Upon in the interview chapter, issues of using the Agile Development Framework addressed by the members were elaborated and analyzed in the section 4.2.4.

As the work of Berg(2007) describes that the discussion section is to present and elaborate key points and suggestions about how the findings fit into the literature survey. Hence, solutions and suggestions according to the interview result were presented and elaborated after finishing the interview part. The analysis techniques used in this



thesis were to analyze the interview data and correlate it with the information obtained from the literature study to come up with some solutions or suggestions. The method of literature survey was used again in the discussion section in order to come up with solutions for the issues. The approach was to find literature related to the members' concerns regarding the Agile Development Framework implementation and come up with solutions or suggestions for those issues. For example, an Agile project could be started up easily with the PDCA framework would be introduced explicitly in the same chapter.

My own opinion of the research result reflections are found in the last chapter, the discussion.

4. Findings- Interview

Analysis and elaboration of interview data are presented in this section.

4.1 Introduction

There are three organizations selected for the interview, which are Volvo-IT, Sweden, IBM, Taiwan, and Ericsson, Netherlands. Actually, I have contacted over 15 companies around the world, however, only 3 of them accepted the interview request.

Besides, most of companies in Taiwan did not follow any project management method. For example, two project managers I tried to interview figured out that they had their own process models for software development developed by their departments. Thus, the interview data are not included in my master thesis since the focus in this master thesis is the Agile Development Framework.

The interview themes were divided into several aspects: the use of the Agile Development Framework, within team, with different department, with customers, and overall.

A face to face interview was conducted while interviewing with Volvo-IT and telephone interview was used for both IBM and Ericsson.

Tools used in the interviews were mobile phones and Skype in order to record the conversation.

4.2 Interview Data Analysis

4.2.1 Volvo IT, Sweden

Volvo IT is a global company and a part of Volvo Group. They provide reliable industrial IT solutions, competitive telemetric services and insightful consulting services to their customers such as Volvo Group, SCA, Skandia, and GE Healthcare.

Place: Volvo-IT Office (face to face)

Date: 11:00 AM. 18th May, 2009

Duration: 60 Minutes

Interviewee: Project Manager, Volvo-IT

The interviewee has been working at Volvo IT as a Subject Matter Expert in Project Management for several years. The interviewee took the responsibility to train project managers with the general knowledge of project management and also with their specific steering models for projects. The project process overall followed the Waterfall Process Model. At Volvo-IT, the project process was divided into seven phases which are Pre-study Phase, Concept Study Phase, Development Phase, Final Development Phase, Industrialization Phase, Deployment Phase, and Follow-up Phase (see Figure 4.1). Two of the Agile Methods, SCRUM and Extreme Program (XP), were only used in the software development phase but not in the whole IT project.

Implementation of the Agile Development Framework, however, was considered for the whole IT projects in order to take the advantage of iterative process model. The interviewee indicated some difficulties of implementing the Agile Development Framework into their project procedure.

First of all, the customers did not understand the concept of the Agile Development Framework so it was difficult to ask them to prioritize tasks. Moreover, the customers felt unsecured without reading a concrete plan at the beginning since they wanted to make sure that everything was carried on by following the plan/schedule. The Volvo-IT is a global organization, which means that some members are in other countries; however, the Agile Development Framework emphasizes a co-located collaboration, so

it is difficult to bring the Agile Development Framework into the whole team.

Finally, the Agile Development Framework does not define the beginning and end part of a project. For example, literatures regarding the Agile Development Framework only explain the software process part such as prioritizing the feature or holding a daily meeting, but in fact, there are many prerequisites before launching a project into the software development phase. For instance, how to solve the communication problem and how to convince customers to use the Agile Development Framework are not answered.

By large and all, the interviewee was satisfied with the Agile Methods running in their software development process, however, it would be better if the Agile Development Framework could suit their whole project process.

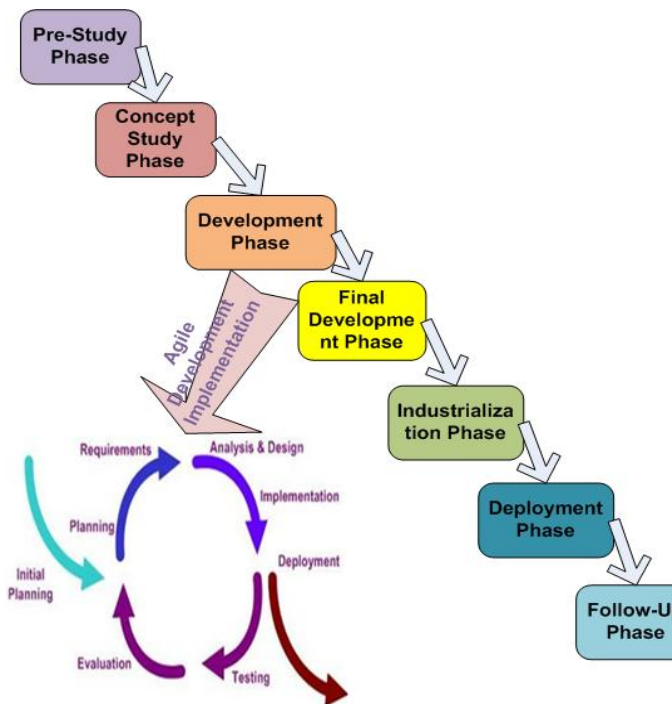


Figure 4.1 Scope of Project Process at Volvo IT.

4.2.2 IBM, Taiwan

Place: Home (Telephone Interview on Skype)

Date: 10:00 AM., 3rd June, 2009

Duration: 30 Minutes

Interviewee: Project Manager in Software Testing Department, IBM

The concept of the Agile Development Framework has been implemented into projects of the software testing team. The Agile Development Framework has been used in the department since 2007. Not only did the IBM invent their own Agile method (RUP) but it is also popular in software development industry. The interviewee indicated that the Agile Development Framework emphasized to divide tasks into backlogs so they could find problems in the early stage and control the test quality easily. Moreover, at the end of each iteration, a demonstration would be held so that team members knew the progress of the project and a frequent feedback could be also delivered to stakeholder.

The interviewee also has had some experience in using the Waterfall Process Model. Differing from the Agile Development Framework, the interviewee thought that the Waterfall Process Model was simpler; in particular for test plan and requirement. For example, a complete test plan and requirement were drawn up in the early stage so their team member could just follow the plan step by step. On the other hand, the coverage of the test plan in the Agile Development Framework was not as complete as the Waterfall Process Model. However, the Agile Development Framework helped the team find some problems in detail so they could focus on those problems.

Communication was the most important part for the interviewee. The main tools used in the interviewee's team were e-mail, discussion within team and IM application. An internal meeting was also held once a week.

Although the interviewee had to work with their offshore team, the Agile Development Framework has only been used in the team in Taipei, Taiwan. For example, they had a software development department located in the U.S.A using the Agile Development Framework as well, but because of time zone problem, they did not share the same framework.



The interviewee also revealed some difficulties of implementing the Agile Development Framework into their off-shore team. In the interviewee's view, the time zone problem and how to remote the teams were the most arduous part of the Agile Development Framework implementation.

4.2.3 Ericsson, Netherlands

Place: Home (Telephone Interview on Skype)

Date: 04:00 PM., 25th June, 2009

Duration: 10 Minutes

Interviewee: Streamline development & Agile driver, Ericsson

The interviewee sent me a clear answer for the interview questions by e-mail so the interview was only held for 10 minutes. The role that the interviewee took on is a streamline development and Agile driver. The interviewee gave support, training and coach solution of the Agile Development Framework implementation to their development team. The interviewee mainly worked in Netherlands but he needed to collaborate with different departments located in other countries such as in Sweden. They mixed the diversity of the concept of the Agile Methods such as XP, SCRUM, and Kanban. Moreover, they also follow the principle of LEAN. They chose the Agile Development Framework as their primary process improvement method because the Agile Development Framework focused on customer value, short feedback cycles and continuous improvement.

However, the Agile Development Framework oversimplifies a project process so it may slow down the progress of projects. The interviewee also encountered the same problem as Volvo-IT, which was that their customers required an extensive documentation and it differed from the concept of the Agile Development Framework. The Interviewee also suffered some difficulties while implementing the Agile Development Framework. For example, in large organizations, decisions were taken on multiple layers so the process was more complex and they had to convince people to adapt the Agile Development Framework since they might not be

willing to change, and the value for customers was also hard to find in multination.

Roles were defined well in their Agile project team. For instance, the roles were defined into such as account managers, key account managers, system integrators, strategic product management, R&D technical product management and developers.

The interviewee also thought that communication was the main factor of a project's success and failure since they stress customer's need and value.

4.2.4 Keys Taken Out From Interview Data

To sum up the three interviews data, we can see the all of interviewees were positive of the Agile Development Framework although they had some problems with it. Not only does the Agile Development Framework focus on the customer value but also its flexibility. For example, the interviewee at Volvo-IT pointed out that he wanted to implement the Agile Development Framework into their whole project process because the iterative process model would bring its benefit to shorten the lifecycle of the project. Moreover, the concept of the Agile Development Framework is to create frequent feedback within teams and for customers so that the progress of projects and customers' needs can be tracked during processing and be reviewed and coped with at the end of each iteration.

On the other hand, the Agile Development Framework seems only suitable for single units or teams as it was assumed. For example, all interviewees needed to cooperate with an oversea team; however, they did not bring the Agile Development Framework they were using in their local team to their off-shore team. Some difficulties of applying the Agile Development Framework into a global team were time zone problem and challenge of managing remote teams.

The interviewees also had problems in implementing the Agile Development Framework in the beginning. For example, as the interviewees indicated that they encountered some problems while starting deploying the Agile Development Framework. It might be caused by the lack of experience and related documentation as well as the Agile Development



Framework is specified over simply. Besides, most of literatures do not elaborate how to start and end up an Agile project and this could be seen a difficulty to implement the Agile Development Framework. In fact, the Agile Development Framework such as SCRUM, is not only for the software development process but also for the project management. Unfortunately, most of literatures only put the emphasis onto the software development process part more.

Furthermore, two interviewees said it was very difficult to express the concept of the Agile Development Framework to their customers. They showed that their customers preferred to have a well-organized plan that the project process should follow. Thus, the project process at Volvo-IT still followed the concept of the Waterfall Process Model. However, this betrayed the concept of the Agile Development Framework.

Finally, factors concerning the interviewees the most with the Agile Development Framework can be summarized as:

1. How to easily start up an Agile project? (It is not easy to start up an Agile project in the beginning.)
2. Can the Agile Development Framework really be adapted into a globally distributed project?
3. Do the Agile Development Framework and the Waterfall Process Model cooperate together?

Thus, the following chapter would be carried out based on following suggestions:

1. Solutions for a globally distributed agile project.
2. Adopting the Agile Development Framework into a Waterfall project by using the key points addressed in the work of Sliger (2006).
3. Starting up an Agile project with PDCA.

5. Discussions

This chapter addresses three points according to the interview section. These are: how to bring the Agile Development Framework into a globally distributed project, how to merge the Agile Development Framework with a Waterfall project and how to start up an Agile project.

Solution for a globally distributed agile project

As section 2.2 describes, the Agile Development Framework is the best methodology for GSD. The interview section, however, discovers the difficulties of a globally distributed agile team implementation. Communication can be thought as the main obstacle which impedes organizations to implement the Agile Development Framework into their offshore team. For example, the Agile Development Framework requires close communication in order to obtain the benefit of team interaction; however, this may not be applicable for teams using the Agile Development Framework because of time zone and distance problem. Apart from the communication problem, organizations may also suffer from challenges such as lack of team cohesion, lack of shared context and knowledge, and unavailability of team members (Passivaara, et al, 2008) while deploying a globally distributed agile team.

However, the companies are still seeking the way to adapt the Agile Development Framework into their offshore team since they realize that the Agile Development Framework can bring the values of continuous improvement, establishing team spirit with virtual teams, automate as much as possible, continuous communications, and sharing expectations into their offshore teams (Massol, 2004).

To diminish the pain of communication, the first step is to set up the project organization well. For instance, defining same roles on both sides as much as possible can be seen as a good solution. For example, as the Figure 5.1 shows, assigning a local project leader on the offshore team not only does prevent feeling of “superiority” from one site but also spread overall knowledge which helps improve productivity (Massol, 2004).

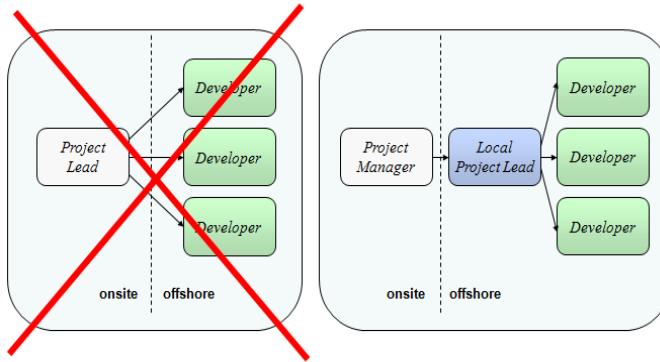


Figure 5.1 Scope of globally distributed agile team. (Source: Massol 2004)¹²

Moreover, team members in a globally distributed project must begin their collaboration early in the development cycle (see Figure 5.2), continuous involvement should be a requirement to help analyze and validate the globalization architecture (Hsieh & Wang 2007).

Thus, business trips are a must in order to maintain collaboration relationship between onsite and offshore teams as well as building trust. Both onsite and offshore teams should travel to each other's sites so that the remote communication can be conducted effectively. The point is to get people used to working together, so some joint tasks can be arranged (Fowler, 2006). There are two kinds of visits: seeding visits and maintaining visits. Seeding visits aim to build a relationship and create trust and they should be planned early in the project. Maintaining visits are shorter and they are supposed to maintain the collaboration relationship (Paasivaara, et al., 2008). However, this method, does not suit for a project with a small budget since a business trip requires a big amount of money.

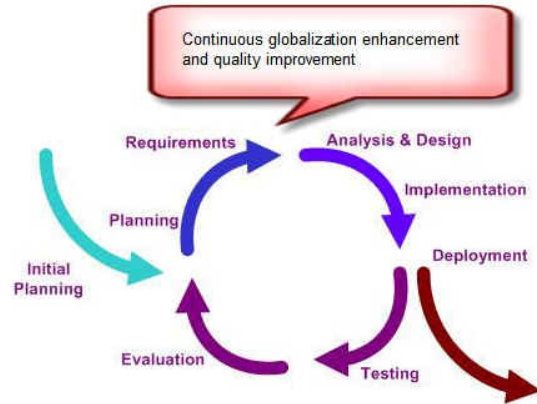


Figure 5.2 Continuous globalization enhancement and quality improvement should be started in the requirement eliciting stage. (Source: Hsieh & Wang, 2007)¹³

Furthermore, a daily meeting is required in order to get a frequent feedback in the Agile Development Framework. Instant Messages (IM) application therefore can be utilized as the primary tools for a daily meeting. Nowadays, IM applications such as Windows Live Messengers (MSN), Skype, e-mail, have become a main communication tool on internet. Those IM tools provide a variety of functions facilitating communication. For example, MSN and Skype allow users to chat online in real time. The chat can be held variously. Users, for instance, can chat by text, voice and video.

In a globally distributed project, IM tools not only smooth the progress of communication between onsite and offshore teams but also fulfill one of the principles within the Agile Manifesto, Customer collaboration over contract negotiation. For instance, Skype is said to be better than a phone because it allows users to write stuff when a verbal communication is obscure. Those IM applications make global communication as easy as yelling over the office partitions (Mawdsley, 2008).

A customer is a major player in a project so it is very important to give him/her the right to access project pages, evaluate the project's progress, request changes and contact the team leader and developer at any time by using those IM applications (Mawdsley, 2008).

¹² Vincent Massol cited the diagram in his report "AODS: Agile Offshore", 2004.

¹³ Joyce Hsieh & Wendy Wang cited in their report "Effective Agile Delivery Toward Globalization", 2007.

Moreover, a virtual private network (VPN) giving teams equal access for sharing resources should also be built for the IT infrastructure (see Figure 5.3). The VPN environment is less strict than a publishing network allowing team members to use such features of IM tools. For example, MSN provides features such as application sharing, video and voice chat, remote assistance, and whiteboard (Filev, 2006).

In addition, as the interview section reveals, time zone difference also troubles the interviewees' attempts to apply the Agile Development Framework into their globally distributed teams. The time zone difference may cause information delay, and globalization requirements or feedback may not be dealt with in real time (Hsieh & Wang 2007).

For this reason, using Web2.0 technology such as wikis may be a good solution to assist in sharing common knowledge. A wiki allows everyone editing privileges so information can be updated frequently. It not only speeds up collaborative communications for stakeholders (Hsieh & Wang 2007) but also provides a space containing common information such as project scope, project schedule, and project status for a globally distributed team. Wikis are also easy to set up and compatible with any browsers (Fowler, 2006).

From the team's perspective, the wiki space lets each team member get to the same page, share the relevant information and gather all the documentation in one, searchable space (Mawdsley, 2008). Moreover, from the customer's value point, project related documents can also be attached so that customers can track the project progress (see Figure 5.4). Therefore, a wiki is the best proposed practice for a globally distributed project as it makes a global communication and collaboration simpler and also reinforces the Agile values of responding to change and working software declared in the Agile Manifesto (Hsieh & Wang 2007).

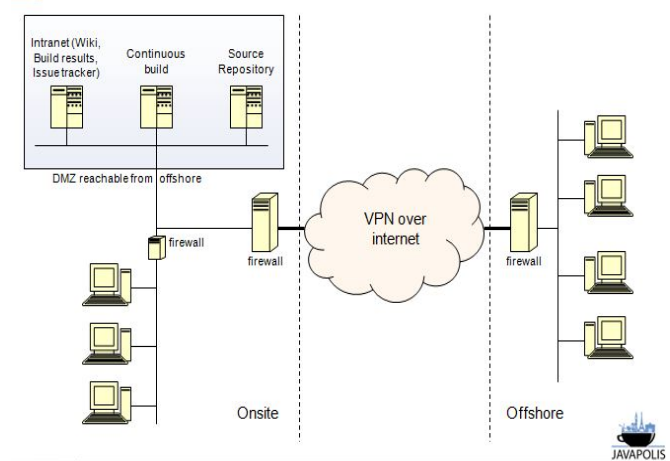


Figure 5.3 IT Infrastructure-VPN (Source: Massol, 2004)¹¹

Further, the team must choose the right practices and tools to apply to distributed teams. For example, common coding standards, a source-control server, one-click, build-and-deploy scripts, continuous integration, unit testing, bug tracking, design patterns, and application blocks are the well-known practices used by successful software-development team. These practices must be adhered to in the workflow of offshore teams more strictly than to local teams (Filev, 2006).



Figure 5.4 an example of wiki space. (Source: Hsieh & Wang, 2007)¹²

Using a short iteration should also be encouraged for globally distributed agile projects. As I mentioned in section 2.1.2, a regular meeting held at the end of each iteration is to get feedback and expound problems. An iteration generally takes two to four weeks in a regular Agile team. However, a regular meeting in a globally distributed project should be

held more frequently because of location consideration. The project goal should be defined and attached in a highly iterative working style for each week. Shortening iterations make project manager and stakeholders keep an eye on things and correct them before they go off into the gap. It also helps developers find bugs in each other's code (Mawdsley, 2008).

By large and all, the strategy for implementing the Agile Development Framework into a globally distributed project can be summarized in the following points (Massol, 2004):

- Appoint good mediators/ coordinators to solve communication issue.
- Wikis smooth the progress of sharing activities between onsite and offshore team.
- Having a frequent business trip between both sites to maintain the collaboration relationship and build trust.
- Write functional test cases before development starts help transfer business knowledge.
- Dedicate offshore support persons in each team to minimize question round trips.
- Shorten iterations with project goal attached to facilitate the track of project progress and bugs exploration.

Furthermore, the scope of a globally distributed agile project can be shaped by three aspects: project organization, collaboration tools, and best practices.

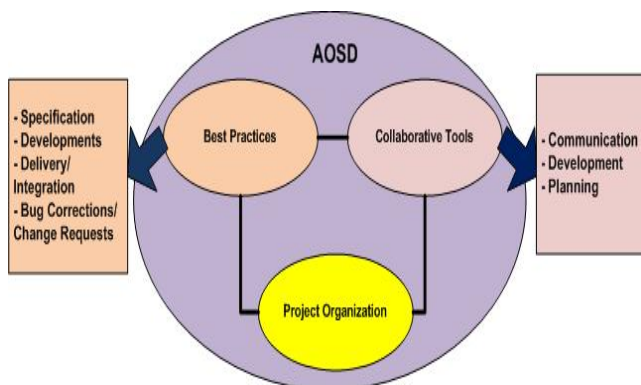


Figure 5.5 Scope of Agile Offshore Software Development (Source: Massol, 2004)¹¹

Adopt the Agile Development Framework into a Waterfall project

The Agile Development Framework seems totally different from the Waterfall Process Model from many aspects. For example, the Waterfall Process Model relies on a well-organized plan, but the Agile Development Framework focus on a face-to-face communication. The section 3.1.1 mentioned that unlike the Waterfall Process Model which is applicable for large projects, the Agile Development Framework was developed for small or medium ones. In the interview section, the interviewees also indicated that their project management method currently follows the Waterfall Process Model because their customers prefer to have well-planned documents in hand. Thus, I came out with an idea, which is to merge the Agile Development Framework into a Water project.

Michele Sliger (2006) addressed several key points for how to implement the Agile Development Framework into a Waterfall project successfully.

1. Finding an executive sponsor: First of all, the Agile team must find an executive sponsor who is in charge of facilitating and driving ongoing improvement in Agile teams. The sponsor also takes responsibility for informing problems exposed by the Agile team. The sponsor also needs to communicate with team members constantly.
2. Explaining the definition of the Agile Software Development to stakeholders: As the findings section reveals that problem of stakeholders not understanding the Agile Development Framework also concerns the interviewees. Thus, the definition of Agile Development Framework must be explained to their stakeholders, and ask for help in making a pain-free working environment.
3. Dividing requirements of the Waterfall project into backlogs: For implementing iteration reviews and incremental improvement to the Waterfall team, requirements of the Waterfall project should be divided into backlogs. Those backlogs

should be prioritized so that the teams can focus on only top two or three for the next iteration.

4. Applying the “barely sufficient” concept: “Barely sufficient” should be also applied in order to use resources efficiently. The “barely sufficient” concept is to bring the principle of “just enough” into the team to make sure that the budget/ resource is under control.
5. Including the Waterfall project’s manager in the Agile team meetings: The Waterfall project’s manager should also be included in the Agile team meetings in order to track the project process. In the second last stage, project review and retrospective should be held at the end of each iteration so that managers could know how to improve the quality of project in the next iteration.
6. Making sure that understanding the Agile Development Framework is everyone’s responsibility: The responsibility of the developers is to implement the Agile Development Framework and report the results up the hierarchy. Should any issue arise that a team cannot or is not authorized to handle, it is the upper management’s responsibility to assist the team in solving them. Thus, it is very important to let all team members understand the Agile Development Framework.
7. Including everyone in the retrospective meeting: The project manager should start preparing for the project retrospective presentation and invite everyone such as stakeholders and Waterfall team members in the meeting. The project’s results are presented in the meeting so the stakeholders or the sponsors can assess the outcomes and make appropriate changes in the next iteration.

Start up an Agile project

In my point of view, an Agile project can be started up easily by following the Plan-Do-Check-Act (PDCA) model which is an iterative four-step problem solving

process typically used in business process improvement. (see Figure 5.6).

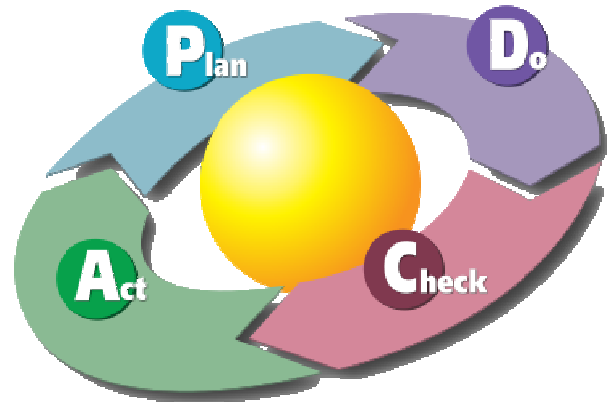


Figure 5.6 PDCA Life Cycle. (Source: Unknown, Wikipedia)¹⁴

Plan → Plan here means that results with the expected output have to be delivered by establishing the necessary objectives and processes. This plan must be a problem solving and goal oriented with a meaningful effort (Little, 2008) fed by limited resources and budgets. Thus some significant dimensions such as the goals of the project, identification of stakeholders and competitors, and the completeness and accuracy of the specification, have to be included in the plan. This plan must also contain the plan to change requirement and the target of the project.

Do → Implementing the new process. A small scale is considered the best. Moreover, the information collection and its procedure have to be considered in this step.

Team formulation and explanation of the Agile Development Framework definition to stakeholder should be carried out in this step as well. Consequently, it is important to assure that the team is on the same page, and agree on a definition of Done since it is often forgotten or neglected (Little, 2008).

Check → Measure the new processes and compare the result against the expected results to ascertain

¹⁴ The diagram is cited on Wikipedia website: <http://en.wikipedia.org/wiki/PDCA>. [Accessed 10, Oct, 2009]



any difference. The use of strategy is also judged in this step.

Activities of preparing a Product Backlog of user stories, work on Iteration zero which includes the preparation of the infrastructure that the team needs to do its work and start daily meetings, and do Sprint Planning are enclosed in this step (Little, 2008).

Act → Analysis of differences and solution implementation for improvement will be brought into the act step. For example, where to apply changes will be determined by reviewing all steps (Plan, Do and Check) and resulted in improvements (Lee, 2008).

The act phase includes several steps (Little, 2008):

1. **Do Sprint Planning:** Prioritization of the tasks and what tasks will be brought into the Sprint should be determined in this step.
2. **Do a daily Sprint and daily meetings:** This step includes completing the User Stories and refactoring the product backlog and preparing the Agile specifications for the stories in the next Sprint.
3. **Hold a Sprint Review Meeting:** The project progress should be arranged to demonstrate the meeting, proposing to get feedback and correct mistakes.
4. **Hold a Retrospective Meeting:** This meeting is to identify actions and determining whether to take actions with some impact before the next Retrospective. The actions should be addressing one or two top items on the Impediments List.
5. **Repeat steps 1 to 4 until effort is finished.**

6. Conclusions

From my point of view, although the communication between onsite and offshore team is the most important problem to be solved in a globally distributed agile project, the distance barrier seems easy to be overcome because of the revolution of IM tools. Compared with ordinary telephones, IM tools not only do provide a more convenient method of

communication but also decrease the cost of communication. The Agile Development Framework emphasizes face to face communication, those IM tools, therefore, seems to be able to fulfill this because of the video chat function.

Travel can also be seen as a good solution for creating trust between onsite and offshore team. From my prior working experience, setting up an onsite support team cannot only report bugs immediately but also ease communication. Onsite support, if needed, means sending one to two onsite team members who are familiar with other languages to the offshore team. The responsibilities are to discover problems existing in the distributed team and report the status of bugs to the onsite development team straight away once problems occur. Onsite support also mitigates the language barrier so communication between onsite and offshore team can be more fluent.

Furthermore, a VPN and wikis offer onsite and offshore team members a place to access and share common resource and information. They also make the project's progress and customers' rights transparent since it also provides a place to track and follow the project's progress for customers.

Creating a source control server is also an essential part in a software project. These configuration management tools such as SVN and CVS allow teams to work on the same document and code as well as keeping track of changes.

Secondly, the Agile Development Framework has to be brought into a Waterfall project gradually. The best manner is to build a bridge of communication between the Agile team and Waterfall team. An executive sponsor responsible for facilitating and driving ongoing improvement in the Agile team can be seen the bridge. Furthermore, it is very important to acquaint customers and the Waterfall team with the concept of the Agile Development Framework. The Agile Development Framework focuses on customers' value and software product's features are prioritized by them. Thus, the definition of Agile Development Framework has to be explained to customers as early as possible.



In the end, following the PDCA model could assist project managers to start up an Agile project easily. On the other hand, according to those interviewees' experiences, they have suffered problems while deploying the Agile Development Framework. Thus, I believe that a main reason to lead an Agile team to succeed is accumulation of experience. However, a fresh project manager is usually lacking in experience. Joining an Agile course, therefore, may help. SCRUM Alliance, for instance, provides some SCRUM certification courses combining literature and practice; this gives the ability of running an Agile project to fresh project managers.

My conclusion is that the Agile Development Framework is recommendable for software development process; nevertheless, it is not suitable for all software development projects. For example, a globally distributed agile project is not suited for a tight-budgeted team since it requires a lot of business trips between two countries. Moreover, the Agile Development Framework emphasizes that the change of requirements is allowable; however, change of requirements always involves lots of team members in a large software project team. Thus, using the Agile Development Framework may delay the project progress if a project manager does not assess the feasibility of implementing the Agile Development Framework in advance.

As a result, just as what the interviewees said, the concept and standard of the Agile Development Framework help you understand the basis of the Agile Development Framework. However, to gain the ability of leading an Agile team is based on the amount of effort you have made as well as how you use it.

7. References

Alan S., K., 2005. *Agile Software Development: Evaluating the Methods for Your Organization*. S.L.: Artech House.

Ajay, B. & Kumarasivan, V., 2009. *The GDM-Agile Paradox: Tips to Tap into the Capabilities of Agile in the Global Delivery Model*. [Online], Available at: <http://www.agilejournal.com/articles/column>

[/column-articles/1470-the-gdm-agile-paradox-tips-to-tap-into-the-capabilities-of-agile-in-the-global-delivery-model](http://www.agilejournal.com/articles/column-articles/1470-the-gdm-agile-paradox-tips-to-tap-into-the-capabilities-of-agile-in-the-global-delivery-model) [Accessed 15 September 2009].

Andrew, F., 2006. *Adapting and Benefiting from Agile Processes in Offshore Software Development*. [Online] Microsoft Inc., Available at: <http://msdn.microsoft.com/en-us/library/bb245671.aspx> [Accessed 20 September 2009].

Agile Manifesto, ca. 2001. *Principle behind the Agile Manifesto*. [Online], Available at: <http://agilemanifesto.org/principles.html> [Accessed 10 May 2009].

Bruce.L.B. 2007. *Qualitative Research Methods for the Social Science*, 6th ed. S.L.: Pearson Education, Inc.

Darrell, N., 2005. *Lean Software Development Overview*. [Online], Available at: <http://codebetter.com/blogs/darrell.norton/articles/50341.aspx> [Accessed 01 October 2009].

Eoin Ó, C.et al., 2005. *Global Software Development: Never Mind the Problems – Are There Really Any Benefits?* [Online], Available at: <http://www.itu.dk/~elisberg/Includes/Papers/1/1-2.pdf> [Accessed 22 January 2010].

Hanna, O. & Mario, P., 2008. *Software Process Improvement for Small and Medium Enterprise*. [E-book] IGI Publishing, Ch.4. Available at Books24x7 <http://books24x7.com> [Accessed 08 May 2009].

IBM Inc., N.D. *IBM Rational Unified Process (RUP)*. [Online], Available at: <http://www-01.ibm.com/software/awdtools/rup/> [Accessed 10 October 2009].

Jason, M., 2008. *Global Development VS The Agile Manifesto*. [Online], Available at: <http://www.agilejournal.com/whitepapers/798-global-development-vs-the-agile->



[manifesto?format=pdf](#) [Accessed 20 September 2009].

Joyce, H. & Wendy, W., 2007. "Effective Agile Delivery Toward Globalization". [Online] IBM Inc., Available at: http://www.ibm.com/developerworks/rational/library/oct07/hsieh_wang/ [Accessed 18 September 2009].

Joe, L., 2008. "How do I start an Agile Project". [Online], Available at: <http://agileconsortium.blogspot.com/2008/04/how-do-i-start-agile-project.html> [Accessed 06 October 2009].

Jochen, K., 2008. "Agile Portfolio Management". [E-book] Microsoft Press, Available at Books24x7 <http://library.books24x7.com/toc.asp?bokid=27540> [Accessed 06 October 2009].

Jim, H., 2000. "Retiring Lifecycle dinosaurs: Using Adaptive Software Development to meet the challenges of a high-speed, high-change environment". [Online], Available at: <http://www.jimhighsmith.com/articles/Dinosaurs.pdf> [Accessed 26, Oct., 2009]

Jim, H., 2001. "History: The Agile Manifesto". [Online], Available at : <http://agilemanifesto.org/history.html> [Accessed 10 May 2009].

Jim, H., 2002. "Agile Software Development Ecosystems". [E-book] Pearson Education. Inc., Available at Google Library <http://books.google.com/books?id=8Or-q-EXdgEC&printsec=frontcover&dq=Agile+Software+Development+Ecosystems&ei=meHmSt7IH5aWzgSKzZ3xCw#v=onepage&q=&f=false> [Accessed 26, October, 2009].

Jimson, L., 2008. "Plan-Do-Check-Act and the PDCA Deming Cycle". [Online], Available at : <http://speedendurance.com/2008/10/14/plan-do-check-act-and-the-pdca-deming-cycle/> [Accessed 10 October 2009].

Kathryn, F. & Mats, G., 2008. "Global Software Development and Delivery: Trends

and Challenges". [Online] IBM Inc., Available at:

http://www.ibm.com/developerworks/rational/library/edge/08/jan08/fryer_gothe/index.html [Accessed 20 September 2009].

Kelly, W., 2008. "User Stories. Published on Agile Software Development". [Online], Available at: <http://www.agile-software-development.com/2008/01/user-stories.html> [Accessed 20 May 2009].

Kirk, K., 2009. "Grass Roots Agile". [Online], Available at: <http://techdistrict.kirkk.com/2007/01/22/grass-roots-agile/> [Accessed 20 May 2009].

Michele, S., 2006. "The Agile/ Waterfall Cooperative". [Online], Available at: <http://www.rallydev.com/documents/AgileWaterfallCoop-Sliger.pdf> [Accessed 10 March 2009].

Martin, F., 2006. "Using an Agile Software Process with Offshore Development". [Online], Available at: <http://martinfowler.com/article/agileoffshore.html> [Accessed 17 September 2009].

Maria, P., Sandra, D. & Casper L., 2008. "Using SCRUM in a Globally distributed Project: A Case Study". [Online] Wiley InterScience, Available at: <http://www.interscience.wiley.com> [Accessed 17 September 2009].

Neil, F., 2007. "Global Agile Development: How investing in the Right Team Impacts Long-Term Rewards". [Online], Available at: <http://www.agilejournal.com/content/view/full/36/33/> [Accessed 12 May 2009].

Peter, S., 2008. "Prioritizing the Product Backlog". [Online], Available at: <http://agilesoftwaredevelopment.com/blog/peterstev/prioritizing-product-backlog> [Accessed 12 May 2009].

Petri, K., 2007. "Extending Software Project Agility with New Product Development Enterprise Agility". [Online], Available at:



<http://www.interscience.wiley.com>

Scott W., A., 2003. "Feature Driven Development and Agile Modeling". [Online], Available at:

<http://www.agilemodeling.com/essays/fdd.htm> [Accessed 12 May 2009].

Scott W., A., 2004. "Agile Modeling Driven Development (AMDD): The Key to Scaling Agile Software Development". [Online], Available at:

<http://www.agilemodeling.com/essays/amdd.htm> [Accessed 09 May 2009].

Scott W., A., 2006. "The Agile System Development Life Cycle (SDLC)". [Online], Available at:

<http://www.ambysoft.com/essays/agileLifecycle.html> [Accessed 09 May 2009].

Scott W., A., 2008. "Answering the "Where is the Proof That Agile Methods Work" Question". [Online], Available at:

<http://www.agilemodeling.com/essays/proof.htm> [Accessed 09 May 2009].

SCRUM Alliance, N.D. "What is SCRUM".

[Online] SCRUM Alliance, Available at:

http://www.SCRUMalliance.org/pages/what_is_SCRUM [Accessed 10 October 2009].

Sommerville, I., 2007. "Software Engineering", 8th ed. S.L.: Harlow-Addison-Wesley.

Sanjiv, A., 2007. "Managing Agile Projects".

S.L.: Pearson Education, Inc.

VersionOne Inc., 2008. "3rd Annual

Survey:2008, 'The State of Agile

Development". [Online] VersionOne Inc.,

Available at:

http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf [Accessed 12 May 2009].

Vincent, M., 2004. "AODS: Agile Offshore".

[Online], Available at:

<http://codehaus.org/~vmassol/blog/AOSD%20-%20Agile%20Offshore%20-%2020041217.ppt> [Accessed 20 September 2009].



Appendix- Interview Question

Description

This interview is for master thesis use only. The goal of the interview is to collect data in using the Agile Methods and compare with the benefit of literature study in order to come up some suggestion to improve it.

The interview will be carried out based on following questions:

Interviewee's Background

1. Which department do you work in?
2. What is your role in your department?
3. What do you do in your job?
4. How many members are in your group?

Agile Process Survey

Methodology

1. **Why did you choose the Agile Methods?**
2. **What kind of Agile Methods do you use?**
3. **Are there any other method used or in use within your team?**
4. **How long have you used it? How has it suited you?**
5. **What do you think about it? (Do you like it? Why? What are some of the strengths and weaknesses? Do you have any examples share?)**
6. **Did you work with other software process improvement methods before? (Ex. Waterfall)**
 - a. *(If yes) Which method did you use?*
 - b. *How do you think about it?*
 - c. *Which one is better? Why?*
7. **Could you tell me about the process of the project in your group?***(How does it work, how to communicate with members, how do you define the working role, how often are meetings held... etc)*
8. **Within the process of a project, what factors are you concerned about?** *(Within budget, on-time delivery, good communication...etc)*
9. **How long is an average project life cycle?**



10. Do you think it is difficult to implement\deploy the Agile Development Framework? (If so, what are the difficulties?)

11. The Agile model is based following principle, for which parts you think that your team is doing the best and which part needs to be improved?

- Eliminate Waste (specification are stable, architecture is easy understood)
- Sustainable Pace (The process has been tracking)
- Intense Collaboration (Good communication with client to understand their requirements)
- Frequent Delivery (Incremental business value to a client)
- Continuous Feedback (Are meetings efficient?)
- Change of Requirements are considered
- Empowerment (Is the team well-organized and team members are positive?)
- Other _____

Within Team

1. **How does your team collaborate? What do you think about it?***(What are the strengths and weaknesses in applying Agile model into your team collaboration?)*
2. **Which are the parts you think that need to be improved in your team?** *(Communication way, efficient meeting, delivery..etc)*
3. **What do you think about the way your team use the Agile Development Framework?**
 - Very good
 - Good
 - Acceptable
 - Bad
 - Very Bad

With different department

1. **Do you need to collaborate with different departments?**
2. **If so, could you tell me how you interact with them?***(What is the role you took on)*
3. **How do you communicate with them? How often?**
4. **Do you think that the Agile model helps with this part?**



With Customers

1. **How do you cooperate with your customers? How often is the meeting held? What do you think about it? Are there any issues that you feel Agile does not address in relation to the customers.**

Overall

1. **What are the factors to lead a project successful for you?**

- Pre Study
- Clear definition of working roles
- Communication with customer
- Communication in the group
- Clear System requirement
- Project Manager
- Project Plan
- Other _____

2. **What are the factors that cause a project to fail for you?**

- Pre Study is not enough
- Unclear definition of working roles
- Lack of communication with customer
- Lack of communication in the group
- Unclear System requirement
- Project Manager is not skillful
- Project Plan is not clear
- Budget
- Other _____

3. **By large and all, do you think the Agile Development Framework fits your project? Are you satisfied with it? (Yes or No)**

- *Why*
- *(If no) Why?*