

Australasian Journal of Computer Science

ISSN: 2251-3221



http://scialert.net/ajcs

Australasian Journal of Computer Science 2 (1): 1-17, 2015 ISSN 2251-3221 / DOI: 10.3923/aujcs.2015.1.17 © 2015 Science Alert



An Examination and Evaluation of Agile Methodologies for Systems Development

Kasra Madadipouya

Department of Computing and Science, Asia Pacific University of Technology and Innovation, Kuala Lumpur, Malaysia

ABSTRACT

In the rapidly growing world, the concept of the traditional development needs to evolve. When applied to the modern workplace, the traditional methods provide as much harm as they do good. Sadly, a lack of adaptability in the traditional methods has led to a rigid work structure that is really not compatible with today's industry. The agile methodology is, therefore, a more acceptable practice, based on producing software at a more rapid pace, while still maintaining efficiency. This being particularly useful for smaller software production firms with limited resources. The agile methodology emphasizes on the quality issue and provides a very stable backbone for today's software development. In this report we discuss the history of agile methodology, in a general context, with listing of the agile manifesto and the agile principles. Additionally, we explain four of the existing agile methodologies with more focuses on the famous and mostly known agile process: the Extreme Programming (XP). Furthermore, the report includes a discussion about the critical success factors, benefits and weakness of the agile methodologies based on a number of existing surveys with some real-world examples that shows some of Agile methodology's advantages and disadvantages.

Key words: Agile methodologies, software engineering, software development, scrum, extreme programming

INTRODUCTION

Now a days, the first role of the software market from the customer's perspective is the development of fast and low cost software with high quality features (Dyba and Dingsoyr, 2008). Customers need powerful software that covers all their requirements with high responsibility to deliver it on the exact time (within the time). The developers and software houses have different criteria for the intended software. The software product should return high and fast profit with less effort. Further, the software houses are usually looking to the successful accomplishment of a larger number of the projects in a shorter time. These and many other are the reasons for the use of agility.

Software usage is growing very fast. All businesses become more and more dependant on IT and software systems. Chow and Cao (2008) noted that, the software development process is not a perfect process. It is a very complicated process. Software development in the early years (from 1970s until 1990s) was very rigors (not flexible enough to accept the changes) and the methods used were very formal (Pfleeger and Joanne, 2006).

The term agility was first coined in 2001 by Beck (2000) and Pressman (2005). It uses a mix of the traditional and iterative process features. The new methodology comes with a number of new

concepts. Several studies have shown that most of the software projects that have used agile have met success with very little delay, failure, rejection, or on going expensive maintenance (Chow and Cao, 2008).

Agile has slowly been more accepted over the past few years. A lot of software development houses are using agile for a majority of projects. Implementing agile methodologies provides a large number of positive results. As noted in the famous agile case study "Shine technologies" (Shine Technology, 2002), the company achieved several success points, when they implemented agile methodology in a number of projects such as:

- Increasing quality of the applications by 88%
- Team productivity improved by 93%

It is interesting to note that there is a lack in research and literature that shows project that had agile weaknesses, when it is implemented (Cohen *et al.*, 2004). That is why agile is currently the most useful method. The reason behind that is if agile methodology is implemented, it will probably work (Cohen *et al.*, 2004).

The main goal of agility is to achieve customer satisfaction by "early and continuous delivery of valuable software". Customer needs change over time. That, sometimes, requires a change in the intended system. Agile also means responsibility for change (requirement or specification). Changing process much related to the customers needs. That will lead to the achievement of customer satisfaction (Jacobson, 2002). All agile goals are noted in three simple sentences by Highsmith (2004). The authors believe that the agile method means" Deliver quickly, Change quickly, Change often".

At the time of this writing, there are six different processes that use agile concepts. Some of which are very widely used in the software industry. That includes Feature Driven Development (FDD), Extreme Programming (XP), Adaptive Software Development (ASD), Crystal, SCRUM and Dynamic Software Development Method (DSDM).

This leads to the agile method becoming an important area of study. Many studies are running to examine the effectiveness of agile development. In this report, we are going to look at agile methodologies' history, a brief explanation of the six agile methods, a comparison of the degree of agility in all these methods when compared to traditional and incremental models, the key success factors of agile methodology and finally the advantages and disadvantages of implementing agile methodology in the modern software industry.

In 2001, a group of the industry's experts met to discuss some ways that might allow software engineers to work effectively, quickly and better respond to changes (faster). They named their group (Agile Alliance). After several months the Agile Alliance stated a manifesto called (The Manifesto of Agile Alliance).

This manifesto was the first step in agile development. The group added some principles that explain the manifesto. A list of 12 agile principles came out by the same group as the foundation of the all agile development methodologies (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 time-scale
- 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

The current agile methods are all implementing differently the concepts of software development, but all follow the same principles. Theses principles show how the right software development process can work without facing problems that are usually associated with the traditional methodologies. However, these principles can be changing over the time. New business or market strategy can result in some changes to these processes. But, it is still suitable and good enough for current industry and for a wide rang of projects.

The agile methodology is different than the traditional methods. The waterfall methodology, for example, represents the "formal" way of software development. As noted in the principles of agile methods, the customers play a very important role. The agile development team looks for customer satisfaction more than in traditional methods.

Hansson *et al.* (2006) noted that the old development methods were stopping the communication between the developer and customers when the requirement specifications were signed. The communication issue takes a higher priority in the agile methods. Now, agile welcomes requirement change at any time or stage. The customer normally changes the requirement and this requires a dynamic team and method that can go back (previous work and requirement) and re-do or re-organize their requirements.

In the traditional methods, customers are always required to wait to see their requirements implemented. But with the new development method (agile methods), the customer can see the progress of the development by continuous delivery of the software (frequently). This will give a higher possibility for discovering and correcting errors during the development process. This means that the agile development process is an iterative process that is intended to deliver a working software piece in a short period of time, while continuously developing the next piece of the software. The overall work should be straightforward (as simple as possible), not complex, build it in pieces (iterative), a fast and effective team and the overall goal is the customer's satisfaction.

AGILE METHODS

In the last few years, many agile methods have come out in order to establish an effective software development age. During the last 8 years, many authors and researchers have come-up

with different approaches that each represents agility in different way. Before we explain what some of the more useful agile methods are, we should explain first the word software agility or agile development.

Many definition of agility have been developed. Qumer and Henderson-Sellers (2007) note that one of the clear and inclusive definitions of agility based on their survey and other author's definitions. They define agility as follows: "A software development method is said to be an agile software development method, when a method is people focused, communications-oriented, flexible (ready to adapt to expected or unexpected change at any time), speedy (encourages rapid and iterative development of the product in small releases), lean (focuses on shortening time frame and cost and on improved quality), responsive (reacts appropriately to expected and unexpected changes) and learning (focuses on improvement during and after product development".

In this definition, there are four important areas that represent the core elements in agile methodology or agility. This includes, flexibility, rapid development, responsiveness and learning from the development environment.

Different methods represent different ways of implementing agile projects, but all are similar in implementing agile principles and roles. Each one might not have unique processes, but are implementing them in different ways than others. There are six agile processes that are already being used in the software development industry.

Feature Driven Development (FDD), eXtreme Programming (XP), Adaptive Software Development (ASD), Crystal, SCRUM and Dynamic Software Development Method (DSDM).

Please note that some agile methods might have a common idea with other ones (Cao *et al.*, 2004). For example, XP and SCRUM both aim to produce a simple software design. That is one of the agile developments roles. However, the use of agile on the industry will depend on the type of project. According to Hansson *et al.* (2006) the use of agile industrially will depend on the project's requirement and also depends on the characteristics of the project and the company itself.

Why agile methods implemented differently? Agile methods are actually implemented differently depending on the perspective of the implementer. Each one of these methods implemented differently using different characteristics. The reason behind that is that each one of these methods try to achieve the agility by solving different issues related to different factors in the development process. For example, the people factor was the main concern in the SCRUM methodology. The authors believe that the human aspect is the main problem with any development process. On the other side, DSDM focus on a dynamic method with faster response for other concerns. The main concern here is the management perspective. The ASD is targeting the development of large and complex projects in an agile manner. The founder of the methods believes that agile can be also into large and complex projects. Crystal method focuses on the people and their influence on the final quality of the product. So, it is looks mainly on the relationship between quality and people.

Although, agile methods have different views but still all implement or use the same manifesto and principles. Agility is still open for the future methods that may implement it differently but still use the same principles.

CURRENT AGILE METHODS

There are currently six agile methods which already implemented. Some authors considers Lean Development (LD) as the sevens one. In this section we will focus on fife agile methods as an example of using of agile in software industry (Appendix A). Table 1 is summarizing the main features, key points and other criteria of agile methods.

Appendix A: Software process in six agile methods						
Software process	XP	Scrum	FDD	ASD	DSDM	Crystal
Development	Short	Scrum	Domain	The project	Active user	Staging
process	releases	teams	object modeling	mission	involvement	
	Metaphor	Product	Developing	Developing by	Empowered teams	Holistic
		backlog	by feature	components		diversity and strategy
	Simple	Sprint	Individual	Collaborative	Frequent product	Parallelism
			class ownership	teams	delivery	and flux
	Testing	Sprint	Feature teams	Joint	Fitness for business	User viewings
		review		application development	purpose	
	Refactoring		Inspection	Customer focus	Iterative and	Revision and review
	0		1	group reviews	incremental development	
	Pair		Regular	Software	Reversible changes	
	programming		buikls	inspection	-	
	Collective				Requirements are	
	ownership				Baselined at high level	
	Continuous				Integrated testing	
	integration					
	On-site				Collaboration and	
	customer				cooperation among	
	Coding				stakenolders	
	counig					
Project	The	Somum	Reporting/	Adaptivo gvolo	Not specified	Monitoring of
managamant	nlenning	mester	visibility of	nlanning	Not specified	
process	game	master	results	plaining		a progress
r	0	Sprint		Adaptive		
		lanning		management		
		meeting		model		
		Daily				
		serum				
		meeting				
Software	Not specified	Not	Configuration	Not specified	Not specified	Not specified
configuration	-	specified	management	-	-	-
control process/						
support process						
Process	Not specified	Not	Not specified	Project	Not specified	Reflection
management		specified		postmortem		workshops
process						methodology
						A

Australasian J. Comp. Sci., 2 (1): 1-17, 2015

Adapted from Qumer and Henderson-Sellers (2007), XP: Extreme programming, ASD: Adaptive software development, FDD: Feature

SCRUM

driven development, DSDM: Dynamic system development method

The word scrum comes from Rugby when two teams scrimmage together to find the ball. SCRUM can be the best agile method, when it comes to project management concepts. The teams or managers can focus on the two potions of the project: management and development. The method focuses on developing an iterative development. Each iteration is called a "Sprint" (Pfleeger and Joanne, 2006). The development process of each release should be within the limit of 30 days. For this reason, SCRUM is more suitable to short term projects.

Additionally, there is a daily meeting of "15 min" that aims to check the progress of the work. The SCRUM focus on self organization of team members and also working in the iterative by prioritizing the requirements. Which one should we develop first or in other words, which one does customer want first? The main activities in SCRUM are: Requirements, analysis, design, evaluation and delivery. Refer to the Appendix B, SCRUM life cycle.

As noted by Schwaber and Beedle (2001), SCRUM is the most flexible, adaptable, empirical, productive and iterative method to meet the software industries needs. This method matches



Appendix B: SCRUM life cycle

Table 1: Summarizing and	comparison of	f different agile	method
--------------------------	---------------	-------------------	--------

Parameters	Scrum	DSDM	Crystal	ASD	XP
Process type	Several releases	Not specified	Incremental paces/iteration	Incremental paces/iteration	Several releases/iterative
Process period	30 days term/release	Not specified	Not specified	Not specified	Two weeks term/iterative
Focus	Self organization	Higher	People has the	Address issues	Communication
	of team members	acceptance	most influence	such as social,	Simplicity
	and prioritize the	probability	on software	cultural and	Courage
	requirement	of changes	quality	team skills	Feedback
		Use 80:20 roles	It looks to adjust		
			for every project separately		
Development speed	Fast	Very fast	Fast	Fast	Very fast
Type of projects	Short-terms	Small/medium	Not specified	Large/complex	Small/large/complex
Special feature	15 min daily	Dynamic	It allows	Human	It is suitable for a small
opecial leavare	meeting	development	agile team to	collaboration	team's workforce (5 to 15)
		Involve	select the	Team self	team members)
		prototyping	most suitable	organization	Maximizing
		method	method		communication which leads
					to enhancing team-work
					Pair programming
Additional	Flexible,	Flexible,	Fast,	Rapid,	Flexible, using coding
features	Adaptable,	fast,	collaborative	iterative	standard, simple design,
	Empirical	collaborative			planning game

DSDM: Dynamic system development method, XP: Extreme programming, ASD: Adaptive software development

between theories and real industry needs. Scrum has three phases: Pre-Game, Development and Post-Game (Schwaber and Beedle, 2001). Pre-Game can be further divided into two phases: planning and high level design (architecture design).

SCRUM meets the agile manifesto in the following way (Advance Development Methods, 1996):

- To minimize the communication, the scrum is organized by dividing the work into small working teams
- The process is adaptable to both business and technical changes. That will help to ensure that the intended product is the best
- Development works are partitioned into clean partitions
- Constant testing, documentation is performed as the product is built

Dynamic Systems Development Method (DSDM): As mentioned before, the agile method involves methods and structures of different and previous development methodologies; the DSDM involves prototyping methods as a way of developing the system. DSDM aims to provide a framework rather than a method that supports a fast, collaborative and iterative software development process for producing high quality business system solutions (Abrahamsson *et al.*, 2002). DSDM is more suitable for small and medium sized projects (Highsmith, 2002).

DSDM represents a dynamic way to develop the system. As a role of agility, the system in DSDM is more flexible and has a higher acceptance probability of changes. Abrahamsson *et al.* (2003) set that the basic concept of DSDM as the resources and time frame are adjusted and then the goals and the required functionality (that is not fixed) are adjusted accordingly. Pressman (2005), further explains that the DSDM "provide a framework for building and maintaining systems which meet tight time constraints through the use of incremental prototyping in a controlled project environment".

Iterative process is the main role and philosophy of developing the system in packages. Additionally, the 80:20 roles engaged with DSDM. Pressman (2005) further explains that 80% of the system can be delivered in 20% of the completion time. So, the development team will focus essentially on enough work require for each iterative package.

The DSDM methodology can be combined with other methods to provide a combination of features (Pressman, 2005). The DSDM can be combined with XP to provide a good method that provides bolts practices (XP) with a solid process model (DSDM).

Crystal: Crystal is actually a grouping of the existing development approaches. The method primarily looks to adjustment for every project separately. It works with every project by applying a different set of polices, conventions and methodologies (Pfleeger and Joanne, 2006). The name crystal comes from the characteristics of geological crystals, each crystal has it is own color, shape and hardness. A direct reference to the Crystal methodologies own nature (Pressman, 2005).

Crystal was originally proposed by Cockburn (2001). The inventor of the method was looking for the relation between quality of the final products and the working people. Mainly, Cockburn was writing a method based on his belief that people have a major influence on the quality of the software process and product. When we improve the quality of the involved people, the overall quality of the software will improve (Pfleeger and Joanne, 2006). The crystal methodology allows for the agile team to select the most suitable method (called crystal family) that is most appropriate with their environment (Pressman, 2005).

As mentioned by Fowler (2005), the crystal method is a people-centered method. Crystal method focuses on dividing work into incremental paces. Each increment may tack some iteration to complete (Qumer and Henderson-Sellers, 2007).

The most common life cycle for all Crystal methodologies projects is: envisioning, proposal, sales, setup, requirements, design and code, test, deploy, train, alter (Cockburn, 1999).

Adaptive Software Development (ASD): The ASD method comes to solve the issue of implementing agile for large and complex systems. The method focus essentially on three elements: rapid, incremental and iterative development that parts of the agile principles. The method focuses essentially on applying these elements on the large and complex systems (Highsmith, 1998).

The key aspects in ASD methods are: Human Collaboration and Team Self-organization. The ASD address issue such as social, cultural and team skills. The first goal of any development organization is to be able to respond quickly to a change that is to be adaptive (Riehle, 2000).

We can add ASD concepts of collaboration and self- organizing teams to the DSDM that can be adapted to a combined process model (Pressman, 2005).

Extreme Programming (XP): As noted by many authors and researchers (Martin, 2003; Cohen *et al.*, 2004; Pfleeger and Joanne, 2006), eXtreme Programming (XP) is the most known, useful and practical agile method used in the real industry. The XP is already implemented in a number of small and medium software projects and it has been successfully adapted many times (Rumpe and Schroder, 2002). Conversely, there is a debate about using XP in large and complex project.

Most XP projects also incorporate practices and principles from other agile methodologies, but Extreme Programming makes a very useful starting point for studying agile methodologies (Highsmith, 2002). The XP essentially focuses on four of the agility elements that include: communication, simplicity, courage and feedback. The XP includes 12 main practices that are suitable for a small team's workforce of 5-15 team members (Maurer and Martel, 2002).

XP represent most of the agile principle as a role of the development process. The iterative process and fast development are the main and the radical tasks in XP. Furthermore, the XP requires cooperation between the customer, management and development team (Qumer and Henderson-Sellers, 2007). The XP practices and life cycle are showed in Fig. 1.

Beck (2000) gives a list of six phases that show the entire process of XP. These include: Exploration, Planning, Iterations to first release, Productionizing, Maintenance and death. The XP



Fig. 1: XP practices and life cycle (Adapted from Jeffries, 2001)

aims to make the system as simple as possible, which will help provide fast and complete systems that are flexible enough to accept changes "welcome change". The XP, similar to other agile methods, reduces the fuzzy requirement in the system by performing continuing development of the system (Lippert *et al.*, 2003).

The customer, in XP, plays a central role in the development process. Customers and developers are required to work together very closely so that they will be aware of each other (Martin, 2003). The XP practices focus on maximizing communication, which leads to enhancing team-work (Cao *et al.*, 2004). They (customer and developer) should communicate with each other. This will improve the entire development process and will make it more efficient. Customer (or user) can give a story about the intended system. This story, for sure, includes the system requirement and this is the starting point to define the requirement and further changes on the system. When the customer delivers continued stories, the developer will have a higher possibility to cover most of the system requirement faster.

The high communication between customer and developers can be a very critical part of agile methodology. This will reduce the time required to develop the system because the customer is very close to the development team. Any further information that is required from the customer can be collected easily and faster. This will reduce the overall time of the development process. Conversely, this will increase the level of customer's acceptance of the work.

At the time that the team knows the requirement and identify the system specification, the team will be able to plan for the development process. The XP plan should be an iterative plan that shows how the system runs for two weeks as suggested by Martin (2003). The plan works in incremental mode. The iterative approach has a high level of error covering which is suitable for this fast development. This is the best way to build a system that easily accepts the changes. After each iteration, the customer will again be involved to evaluate the developed system. The Acceptance test is the only way to evaluate the system and guarantee that is has met customer satisfaction. However, this test normally runs in latest stages of the traditional development methods (for example, waterfall). This leads to late error's covering.

The most important concept that is introduced by XP is the Pair programming. Jeffries (2001) explains the concept of pair programming as, two programmers share the same computer. One writes the program and the other watches the code (to test the system). Both of the programmers switch places frequently. That will increase the communication between team's members and will increase the possibility of early error covering. However, this might adds a new testing technique in the early stages before unite or functional testing runs.

Based on their experience, Lippert *et al.* (2003) mentioned that XP can be very effective and suitable for use in complex projects, domains or when, limited resources are available. The authors found that XP offers a high degree of security and reliability, further it kept the advantages of agile software development.

One of the examples of implementing XP and the helpful results of the method was noted by Maurer and Martel (2002). The authors studied a small company that uses XP to develop a web-based system over 16 months with 9 developers. The company moved from traditional object-oriented methodology to XP method. The authors study the improvement that was given by XP for the new release compare with previous version. They used three software size metrics:

- NLOC measures new lines of code (Java source code plus HTML code)
- Number of Methods (NOM) measures the number of new methods
- Number Classes (NOC) measures the number of new classes

Table 2: Productivity gains wit	h XP compared with previous projects that	used traditional methodology (M	aurer and Martel, 2002)
Parameters	NOLC/effort	NOM/effort	NOC/et
Average pre-XP	10.2	0.36	0.0

17.0

66.3

Australasian J. Comp. Sci., 2 (1): 1-17, 2015

1.45

302.10

NOC/effort

0.05

0.21

282.60

NOLC: New lines of code, NOM: Number of methods and NOC: Number of classes

Table 2 summarizes their findings and shows productivity gains between 66.3 and 302.1% based on static metrics. In their study they measured effort as the number of hours billed to the customer (Maurer and Martel, 2002).

AGILE SUCCESS FACTORS

Average with XP

Percent change

The agile methodology provided wonderful results that change the overall view of the software process. These results are the outcomes of good implementation and of these agile methods. That software project was rejected, failed, overrun or delayed (Chow and Cao, 2008). This is actually what agile is intend to solve.

There are a number of factors that can lead agile software development houses to be successful. Chow and Cao (2008) said that as a consequence there are 12 success factors that affect all agile methods. Their study builds on a web-based survey of 109 companies that use agile methods in 25 countries around the world. They reached the result that most success factors can be grouping into five main categories: People, Organization, Project, Process and Technical factors (Table 3). Most of the factors that were noted and further the failure factors, are a result of experience and lessons learned from previous project. Software projects are not just a technical concern. Many other matters should come into the picture such as the organizational, management, people and so on.

Several attributes can affect the use of agile methods. As similar to any software development mechanism, the attributes of the quality, cost, time and scope are the most effective attributes that can lead us to the success or failure of implementing agile methodology (Cohn and Ford, 2003).

As we see in Table 3, the five categories that are mentioned before include some factors that are similar to each other. Each category represents one point of view. Organizational factors, for instance, focus on the management aspect of agile methods and look for the cultural and environmental space of the organization. Moreover, the communication between people (customer, management and team) was also considered. However, these organizational factors will improve the overall development process and it will affect the final product.

People factoring is related to the environmental factors so it also interrelates with organizational factors. Where, it increases the motivation and team's work corporation compared with traditional methodologies. The management style is also required to be adaptive and "agility thinking".

Technical factors are a little different than others because it focuses on the technical and procedural issues of the development process (for instance: software design, coding, testing etc). By implementing agility, the developments team will not lose time and effort to produce a complex design. They put what is necessary from the requirements into their software in a simple form. Furthermore, more standardized structures will apply on the development process such as code standards and quality standards. However, the technical issue will focus more on the delivery strategies of the final product (Chow and Cao, 2008).

Project's factors take us back to the environmental aspects and the management side. The nature of the work is very important and it will affect the whole development process. It also

Category	Factors
Organizational	Strong executive support
-	Committed sponsor or manager
	Cooperative organizational culture instead of
	hierarchal
	Oral culture placing high value on face-to-face
	communication
	Organizations where agile methodology is
	universally accepted
	Reward system appropriate for agile
	Facility with proper agile-style work environment
	Collocation of the whole team
People	Team members with high competence and expertise
	Team members with great motivation
	Managers knowledgeable in agile process
	Managers, who have light-touch or adaptive management style
	Coherent, self-organizing teamwork
	Good customer relationship
Process	Following agile-oriented requirement management process
	Following agile-oriented project management process
	Following agile-oriented configuration management process
	Strong communication focus with daily face-to-face meetings
	Honoring regular working schedule-no overtime
	Strong customer commitment and presence
	Customer having full authority
Technical	Well-defined coding standards up front
	Pursuing simple design
	Rigorous refactoring activities
	Right amount of documentation
	Regular delivery of software
	Delivering most important features first
	Correct integration testing
	Appropriate technical training to team
Project	Project nature being non-life-critical
	Project type being of variable scope with emergent
	Requirement
	Projects with dynamic, accelerated schedule
	Projects with small team
	Projects with no multiple independent teams
	Projects with up-front cost evaluation done
	Projects with up-front risk analysis done

Table 3: Agile critical success factors (Chow and Cao, 2008)

includes the team response for the requirement changes factor and the culture of working in small teams. Project factors are strongly interrelated with other factors such as the organizational factors and people factors.

As a conclusion to this section, agile success factors are noted by many researchers and authors. They note various numbers of success factors of agile software development processes for different project types. Most of these factors are interrelated to each other. The organizational factor, for example, affects the people and the project factors directly. Moreover, the improvement of the project needs improvement in the people and organization. Conversely, these factors are depending on the type of project. Each project may have certain requirements which make these factors different from one situation to another.

AGILE ADVANTAGES

Several tangible advantages arise through agile methodologies. Agile as we noted before, shows a new approach to build up a system in speedy form. That is one of the requirements of the

fast-changing modern business environment. It satisfies today's actual business needs (Cockburn, 2001). The traditional method had some problem with rapid changes and that has been avoided by agile method. Agile methods, further, increase the response for changes by making the customer working closer with the development team. The customer then becomes free.

Agile comes out with new concept that are also being adopted by other techniques and methodologies. The next version of the PMBOK (Project Management Body of Knowledge) might include some of the agile practice such as the daily meeting technique (the meeting between team members every morning) (Griffiths, 2004).

Customers are also able to see fast delivery of their requirement by the short term release facility. Agile methods aim to produce release of the product after very short time (every 2 weeks in XP, for example). That will help customer see the progress of implementing their requirements faster. The fast deployment will reduce the time we need to catch and repair the error in the system. This also will reduce the long time and the effort that we spend on testing the software. Less resource is also one of the advantages when using agile methods. Agile teams are formed by a few team member compared by the traditional method. These teams are more flexible, manageable and have high communication than other traditional method's teams.

Agile development means rapid process. It is about the process they are going to do. The firms that is implementing agile methods achieve one of the agile concepts called "agile-oriented". That includes project management, requirements and change of configuration management. Another benefit that we can gain from implementing agile is that the development company increases management control of their project's schedules. This will affect on the entire process by reducing the overrun times and costs.

AGILE WEAKNESSES

The role of success and failure of the software development methodology is applicable for all software methods. Agile is not a special case that can provide success everlastingly. However, the failure of agile method is normally caused by lack of understanding of the method or because of the erroneous implementation of the method. There are a number of the problems noted by Chow and Cao (2008) as a result of their study of agile method based on other's studies (Table 4).

Table 4: Agile failure factors (Chow and Cao, 2008)

Category	Factors
Organizational	Lack of executive sponsorship
	Lack of management commitment
	Organizational culture too traditional
	Organizational culture too political
	Organizational size too large
People	Lack of agile logistical arrangements
	Lack of necessary skill-set
	Lack of project management competence
	Lack of team work
	Resistance from groups or individuals
	Bad customer relationship
Process	Ill-defined project scope
	Ill-defined project requirements
	Ill-defined project planning
	Lack of agile progress tracking mechanism
	Lack of customer presence
	Ill-defined customer role
Technical	Lack of complete set of correct agile practices
	Inappropriateness of technology and tools

Most of the failure factors that were noted in the previous table can be categorized as "lack of management". That is the most difficult part in agile methodology implementation. Moreover, the lack in understanding the agile culture is one of the famous problems in agile development. Many IT professionals and managers view agile as a slightly chaotic methodology (Lippert *et al.*, 2003).

There are some that say about agile in large projects, the principles of the agile methods, some times, are not applicable for the large and critical life based project. Boehm (2002), one of the well-known researchers on the field of software engineering, argues that agile methods are very difficult to implement in complex and large projects. Many software project managers support Boehm's idea. Boehm (2002) explained further that agile compared with other methods is lacking in planning, over focused and has an insufficient test coverage. As well, Cao *et al.* (2004) recommends that we do not use the agile methodology with mission-critical software development. Agile, for example, focuses on simplicity. Hence, Critical life projects can not be done with a simple software design (design's simplicity is one of the agile roles). Anything missing in the design can cause a catastrophic affect on the system.

However, Cao *et al.* (2004) recommended using agile in the large business projects due to the dynamic business environment. Some of the larger projects face the same issues that can be solved by using agile methodologies such as requirement changes, ambiguous user requirements and time pressure. XP, for example, makes the software more reliable and it offers a higher degree of security (Lippert *et al.*, 2003). This shows how agile improve the overall quality of the software (reliability, realability and security is a quality feature).

Several projects around the world have noted a failure of implementing agile methods. One example of the agile failure has given by Berger (2007). The author analyzes a case study for the UK Regional Government Department. The clients department in the organization is responsible for managing the administration and expenditure of the EC's Common Agricultural Policy (CAP) schemes through a number of grants and subsidies. His paper studied the difference between implementing a "bureaucratic" approach compared with an agile development approach. The author studied the relation between stakeholders when they implemented agile methods compared with the waterfall method. As result of his study, Berger (2007) noted that:" Both the Developers and the Client Department suggest that had a traditional 'Waterfall' development approach been adopted the project would have been canceled early during second year. Although the project is still on-going, it is seen in a positive light". That was the good news about the project. However, the cultural issues were one of the failures that are noted by the project. Further, if the project is measured against more traditional method, such as the waterfall model, by time and cost factors (delivered on time and within the budget) the project will be considered a "failure project" (Berger, 2007).

Now a days, many companies around the world prefer to use agile over other methods (Boehm, 2002). However, agile advantages are more numerous than the disadvantages. The number of advantages offered by agile makes a vast difference in the software development industry.

APPLICABILITY

As we noted before, there are a wide number of projects around the world that have adopted the agile method and they found wonderful results. In this section, we are going to look at several types of projects that implemented agile methodology and the achievements of those implementations.

British car auctions: British Car Auctions (BCA) is the largest European vehicle re-marketing company. The companies' role is to take vehicles on consignment from volume sellers and re-market them to dealers and trade buyers. The company uses both physical and electronic methods to do their business. They have 38 centers in eleven countries across Europe. The focus of the project was on the automation of a proactive service level management. The technology that it uses should help the company ensure the maintenance of the service levels at all stages of customer interaction. "Technology offered an answer: a system that automatically generated early warning of required actions and gave more assistance to BCA staff with the privatization of tasks, could really help boost service levels and speed up the vehicle re marketing process" (Conchango, 2008).

For the project, Conchango (the development company) has adopted the Scrum method. The method helps the project and management team to be more directly involved in the project. They could see they were directly influencing the outcomes from early stages of the project. As a final result, the system was delivered on time, within the budget and with satisfied users, who are now rolling it out to customers over a six-month period (Conchango, 2008).

Offshore project: StarSoft Labs implemented a project called offshore for a large multinational telecom in Europe. The project was one of the most challenging, unconventional and rewarding XP exercises for the company. For the remote customers, the company built an internet resource and project management application. The project customer was the remote customer. The project development and testing stages was all done 100% in St. Petersburg, Russia. Till February, 2007, 30 months later the system had 1500 users across four countries. The offshore XP team has grown from three to 17 and has delivered 18 versions of the product. The goal of using XP here was to deliver working software against all odds by using tools, metrics and guerrilla XP tactics.

The most effective achievement of the using agile was that the XP method brings the defect rate down while keeping good velocity and implementing changes rapidly, further, the offshore XP team was productive, even with out on ongoing customer presence.

Link stock solution: The Effective Soft has developed the Link Stock system (Effective Soft, 2008). The project of Link Stock is for a large medical company. The company was to use link exchange method while promoting their website. A special team of SEO form the company specialists was searching the internet or relative website to exchange links with. The link base grew and needed close attention as each link had to be checked for validity on regular basis. The check was done manually by a team member. The marketing manager realized that they needed to optimize the process of posting and checking the links. For that, the marketing manager of the company suggested creating a special solution that would stock the link and check them regularly.

The company signed a contract with Effective Soft to develop the project. The project manager chose to work with the agile method. After a period of time, the customer was presented with the project spring-release. The Effective Soft uses some of the agile method principles. As a direct result of the implementing agile, the firsts release version of the Link Stock solution was ready for use in very short time. Additionally, more features and functionality were added with each (next iteration). The final solution allowed (Effective Soft, 2008):

- Merging all the old bases into one new base
- Automatic check of the link validity
- Providing different access levels
- Optimizing work process

Australasian J. Comp. Sci., 2 (1): 1-17, 2015



Fig. 2: Comparison of the degree of agility for six agile methods as measured for the phase's level and the practices level Qumer and Henderson-Sellers (2007)



Fig. 3: Comparison of the degree of agility for the six agile methods with spiral and waterfall Models Qumer and Henderson-Sellers (2007)

COMPARISON

Agile methods, compared to other methods are very useful and more suitable for a number of applications. According to the advantage that we noted before and to the study by.

This degree is different between practices and phases of the method. Figure 2 shows the degree of agility in different phases and also on practical level. Figure 3 The level of agility for all six agile methods compared with traditional methods (waterfall and spiral models).

CONCLUSION

Agile development methodology addresses the need of the today's business environments. The methods offer a fast development strategy for the software development houses with a higher level of the product's quality, performance and control. Furthermore, the agile method looks primarily to satisfy the customer. This is a direct goal of the development process. Agile method achieves customer satisfaction more than other traditional methods. It makes the customer part of the team, so that the customer will be able to see the work's progress and he/she can be satisfied from early stages. There are some principles we should follow when we develop using the agile method. Implementation of these principles depends on the nature of the project.

Some of the agile methods that are already in existence represent agile principles in different ways for different type of views. The six agile methods include: Feature Driven Development (FDD), eXtreme Programming (XP), Adaptive Software Development (ASD), Crystal, SCRUM and Dynamic Software Development Method (DSDM). These methods share some of the practices and features but still represent a unique way of developing the system.

Agile methods, similar to other methods, have several advantages and disadvantages. The fast development, customer satisfaction, error recovery in early stages, short term release, high communication between stakeholders and the simplicity are the most effective advantages of agile method. These criteria make a forward movement in the development methodologies and cover some problems that arise with traditional methods. The most common limitation of the agile method is that the agile methodology becomes less effective when we use it to develop large scale projects. Moreover, this methodology is not suitable for the critical life systems that can not accept the simplicity as role.

Many software companies and organizations are using agile in their projects. We noted some of the examples that show the success and failure factors of agile in different types of projects. However, agile is still mainly a secret in the software development world. Many companies were changing their methodology from traditional to agile structure. Real word business needs rapid and quality development rather than complexity.

REFERENCES

- Abrahamsson, P., J. Wasta, M. Siponen and J. Ronkainen, 2003. New direction on Agile methods: A comparative analysis. Proceedings of the 25th International Conference on Software Engineering, May 3-10, 2003, May 3-10, 2003, IEEE Computer Society, Portland Oregon.
- Abrahamsson, P., O. Salo, J. Ronkainen and J. Warsta, 2002. Agile Software Development Methods, Review and Analysis Report. VTT Publications, Finland, Pages: 478.
- Advance Development Methods, 1996. The origins of SCRUM, control chaos. Advanced Development Methods, Inc.
- Agile Manifesto, 2001. Agile manifesto and principles: Manifesto for agile software development. http://agilemanifesto.org.
- Beck, K., 2000. Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading, MA.
- Berger, H., 2007. Agile development in a bureaucratic arena-A case study experience. Int. J. Inform. Manage., 27: 386-396.
- Boehm, B., 2002. Get ready for agile methods, with Care. IEEE Comput. Soc., 35: 64-69.
- Cao, L., K. Mohan, P. Xu and B. Ramesh, 2004. How extreme does extreme programming have to be? Adapting XP practices to large-scale projects Proceedings of the 37th Hawaii International Conference on System Sciences, January 5-8, 2004, IEEE Computer Society,.
- Chow, T. and D.B. Cao, 2008. A survey study of critical success factors in agile software projects. J. Syst. Software, 81: 961-971.
- Cockburn, A., 1999. A methodology per project. http://alistair.cockburn.us/Methodology+per+project.
- Cockburn, A., 2001. Agile Software Development. Addison-Wesley, UK.
- Cohen, D., M. Lindvall and P. Costa, 2004. An Introduction to Agile Methods. Elsevier Inc., UK.
- Cohn, M. and D. Ford, 2003. Introducing an Agile process to an organization. IEEE Comput. Soc., 36: 74-78.
- Conchango, 2008. British car auction: How agile methodologies helped BCA delivery its smoothest ever project. British Car Auction, Conchango, UK.
- Dyba, T. and T. Dingsoyr, 2008. Empirical studies of agile software development: A systematic review. Inform. Software Technol., 50: 833-859.
- Effective Soft, 2008. Agile case study. Software Development Company, USA. http://www.effectivesoft.com.

- Fowler, M., 2005. The new methodology. December 13, 2005. http://www.martinfowler.com/articles/newMethodology.html.
- Griffiths, M., 2004. Using Agile alongside the PMBOK. Proceedings of th PIM Global Congress, (PMI'04), Anaheim, California.
- Hansson, C., Y. Dittrich, B. Gustafsson and S. Zarnak, 2006. How Agile are industrial software development practices? J. Syst. Software, 79: 1295-1311.
- Highsmith, J., 1998. Adaptive Software Development: An Evolution Approach to Managing Complex Systems. Dorset House Publishing, USA.
- Highsmith, J., 2002. Agile Software Development Ecosystems. Pearson Education, USA.
- Highsmith, J., 2004. Extreme programming. E-Business Application Delivery Newsletter, Cutter Consortium.
- Jacobson, I., 2002. A resounding "yes" to agile processes-but also more. Cutter IT J., 15: 18-24.
- Jeffries, R., 2001. What is extreme programming. November 8, 2001. http://ronjeffries.com/ xprog/book/whatisxp/.
- Lippert, M., P. Becker-Pechau, H. Breitling, J. Koch and A. Conrstadt *et al.*, 2003. Developing complex projects using XP with extensions. IEEE Comput. Soc., 2003: 67-73.
- Martin, R., 2003. Agile Software Development: Principle, Patterns and Practices. Person Education Inc., New Jersey.
- Maurer, F. and S. Martel, 2002. Extreme programming: Rapid development for web-based applications. IEEE Internet Comput., 6: 86-90.
- Pfleeger, S. and A. Joanne, 2006. Software Engineering: Theory and Practice. 3rd Edn., Pearson Prentice Hill, New Jersy.
- Pressman, R.S., 2005. Software Engineering: A Practitioners Approach. 6th Edn., McGraw Hill, New York.
- Qumer, A. and B. Henderson-Sellers, 2007. An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering, Elsevier B.V., USA.
- Riehle, D., 2000. A comparison of the value systems of adaptive software development and extreme programming: How methodologies may learn from each other. Proceedings of the 1st International Conference on Extreme Programming and Flexible Processes in Software Engineering, Addison-Wesley, USA.
- Rumpe, B. and A. Schroder, 2002. Quantitative survey on extreme programming project. Proceedings of the XP2002, (XP'02), July 2, 2002, Sardinia.
- Schwaber, K. and M. Beedle, 2001. Agile Software Development with Scrum. 1st Edn., Prentice Hall, New Jersey.
- Shine Technology, 2002. Agile methodology: Survey results. Shine Technologies Pty. Ltd.