J. Basic. Appl. Sci. Res., 3(2s)57-61, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

The Role of Scrum Methodologies in the Design and Production Software

Parastoo Fathi[1*], Arman Morovat[2]

Ms of Computer Engineering

Received: June 10 2013
Accepted: July 10 2013

ABSTRACT

Agile methodologies were introduced in the mid-nineties, largely based on the rules and principles for the design, programming and testing requirements. Many agile methodologies are lightweight and have a process to comply with the Charter of agile methodologies. In this methodology, modeling and documentation to recommend to the extent needed and useful. This view, the mere manufacturing models and documentation follow one methodology or framework as hazardous waste. Due to the lack specified framework for building software by using agile methodologies, assessment and selection specific method according to characteristics of a project is a difficult activity. In this paper, after the analysis of agile methodologies and explain their strengths and weaknesses points, the survey Scrum methodology, how it operates, role of individual and development team, its different meeting and tools that are used in it, is paid. That according to the specific needs of each project, you can use this method to choose.
KEYWORDS: Agile methodologies, Scrum methodologies, Light weigh process, XP, FDD.

1. INTRODUCTION

Agile methodology in response to the problems that traditional approaches to create software product, is presented. In short, agile modeling can be used in situations where no other available models, can be used. Some of these conditions are as follows:

- Small projects (number of people involved are from 2 to 30 people).
- Customers and software developers are in constant contact with each other.
- Requirements may be modified during the project.
- Project fully defined territory.
- The issue is complex [1].

Software development methodologies generally be divided into two categories: heavyweight and lightweight. The main axis of heavyweight methodologies such as RUP include comprehensive planning and documentation from beginning to end perfect and widespread design. In other words, heavyweight methods act as a soothsayer to predict that the beginning of everything. Here the question arises whether everything is predictable from the start?

Certainly in the early analysis phases due to the change requirements can not predict everything so came to Fife lightweight methodologies. Most of these methods focus on simplicity and speed. In this methods in a development work, the group developer only focus on the tasks that need at first, and they will be quick to deliver and at each stage are to gather feedbacks and respond to information received [1]. If a development application has the following features, then an agile methodology would be:

- Incremental: Small software emissions with rapid turnovers
- Collaborative: Users and developers with a close relationship working constantly together.
- Straightforward: the approach to learning and changing is easy and well documented.
- Compatible: has the capability of variability in the final moments, which is meaning it is coincident with conditions.

Lightweight methodologies are used mostly in small projects whereas for large projects should be used from heavyweight methodologies but this issue does not reduce the popularity of these methodologies because the number of small projects is far more than large projects [2].

In the heavyweight methodologies, management is absolute and autocratic, whereas the management of the lightweight methodologies is a free and decentralized, that this decentralized management can make better decisions for these methods. [3]

One of disadvantages of the heavyweight methodologies, is heavy documentation, which is very difficult and time consuming task and should be comprehensive (For example in RUP should be prepared all documents for each phase completely) whereas In the lightweight methodologies, documentation, limited and depending on the project need to be done [4].

At heavyweight methodologies number of cycles is low, but their time is too much so be prolonged cycles caused will be longer waiting time to reach emissions and this is not interesting to Impatient employers who would

[*]**Corresponding Author:** Parastoo Fathi, ms of computer engineering fathi.parastu@yahoo.com

otherwise break down the project and will have also cost tolerance but in the lightweight methodologies measure of success will be determined based on the value of the work, is that it makes this methodology very flexible whereas in the heavyweight methodologies, this flexibility, because of the original plan, does not exist [5].

Heavyweight methodologies need to have a great team who must do own operations on the basis of their expertise in each phase. In the lightweight methodologies, the size of team is small (maximum 30 people) that small team can lead to further innovation and collaboration in the team [6].

As mentioned above, in the lightweight methodologies, the sooner we will reach to software effect that will makes faster return on investment, during the project whereas in the heavyweight methodologies must wait until the end of the project! so we can say that lightweight methodologies are economically affordable. Overall, in Iran those who use software development methodologies (if they do it is in principle) can use lightweight methodologies rather heavy methodologies [7].

An agile process, a process that is always ready to embrace the demands of society and the degree of the adaptation might have. Therefore, the purpose of speed, is not just lower the volume or speed of software production process presented to the market; rather the intent is flexibility and maintain the quality.

2. Definition of Agile Software Development:

A conceptual framework, which is responsible for software engineering projects. Agile, is a combined of philosophy with a set of guidelines for developing software [8]. Agile design guidelines:

i. Pressure for early delivery of products to rather than analysis and design
ii. Continuous communication between developer and customer

3. What is Agility?

- Respond effectively (fast and friendly) to changes
- Effective communication between all stakeholders
- Putting customer, In the software team

Percentage of activities carried out in software development:
Analysis: 16%
Design: 17%
Code testing / integration: 18%
Documentation: 8%
Implementation / Installation: 7%

3.1 Agile Methodologies Project profile:

- Cup of being alder in the surface of process.
- Reproducibility with repeated short courses that address In the performance is fast and accurate.
- Schedule, with repeated periods of one to six weeks.
- Savings in the development process with the withdrawal of all non-essential activities.
- Adaptable to new risks those are likely to occur.
- The incremental process, which is subject-oriented projects have to be made in small steps.
- Convergence and incremental process, reducing the risk.
- Support personnel, processes and technology in subtle processes [7].

3.2 Agile methodology statement:

In 2001, 17 people who were active In the the field of agile methods including Kent Beck, Fowler Martin, Alistair Cockburn and ... , Bob Martin came together and initiated the meeting, a statement that the statement was a feisty reputation. This statement refers to the following four principles:

- ❖ Individualism, and engage the best of processes and tools.
- ❖ Software runs better sense of documents.
- ❖ Cooperation with our customers, superior contract-oriented negotiations.
- ❖ Respond to changes better than the original track plan [9].

3.3 Impact of Agile Requirements:

Traditional software agreement is based on the following points:

- Design and implementation of a system should be clearly specified.
- A specification must be unambiguous, correct and traceable and is independent of the implementation.
- Requirements and models of user needs is a valid statement.
- All requirements have equal weight.
- Changing requirements are higher cost than the progress of the project.

But Agile processes explicitly quantitative analysis, analysis and design. Reports show that 45% in the project failures are caused by poor requirements gathering. Many systems have been delivered, will provide business metrics. Agile approaches allows requirements to provide more business metrics while developing [7].

4. Types of Agile methodologies, processes and lightweight

4.1 Methodology "XP"

The methodology in 1997 by "Beck" was presented. Beck believes that "XP" is rather a system software engineering methodology [9]. This method, for use in small teams, which require fast production software, is designed in a flexible environment. This approach allows manufacturers to respond to customers' changing needs without any hesitation and your main objective (customer satisfaction) to achieve this, the success of "XP" is.

4.2 Methodology "FDD"

This method is based on iterative development, with the best and most effective activities and focused on quality aspects. This method involves detailed follow-up versions of tangible progress in the year 2002 by "Palmer" and "Flesing" was introduced.

4.3 Methodology "DSDM"

Method 1997 «DSDM» as a general framework for rapid construction of applications was introduced. In this framework, a set of controls that developers create software they use for process control [7].

4.4 Methodology crystal

this method involves a different set of practices most appropriate choice for each project is unique. Crystal Method is a systematic way for different conditions on the project is customized. Currently, there are three ways the original Crystal: Crystal Clear, Crystal Orange and Crystal Orange Tvrynh [7].

4.5 Scrum Methodology

Scrum is one of the most popular agile methodologies. This methodology was introduced in 1993 by Ken Shabr and Jeff Sutherland. Scrum has three main it is very interesting. The first property is simply, this methodology can be less than a day to learn the basics. The second feature is scalability Scrum, the Scrum can build a variety of different applications, and they can be used. Another important feature is that Scrum values for all project stakeholders [4].

Scrum values for customers and product users. Scrum an iterative and incremental nature of reality is the priority scheduling requirements. This feature allows the customer to the most precious and the most important feature for a faster delivery, choose. In addition, the end of each client can decide to terminate the project, because the current product is good enough, or that the projects do not face value [4]. Scrum remaining tasks by providing tools such as charts, product Karanbar , Karanbar sprint, and session events, such as the Daily Scrum, sprint planning meetings, sprint review meeting to clarify various aspects of the project, contributing to the Director of the Scrum values. This transparency allows the administrator to have more control over the projects and problems that may occur during a project life of ten, he quickly identified [1]. Scrum for development - their values. They usually are stimulated greatly by learning opportunities and technical challenges. Join the team as a team, multitasking and self-organizing Scrum, an ideal place for them, because they can all development activities, from requirements analysis to code, become involved. The multitasking, different Batkhsshay are a group of people who work together for a common goal [4].

5. Summary of Scrum

Scrum is a framework for iterative and incremental software product development or that it will organize a cycle of works called sprint. Speed for both two and four weeks and are done one after the other. During this time period is fixed at a certain date can not be extended to the end and never, whether it is complete or not [9].

At the beginning of each sprint, where a priority list - sort by a few minutes - a task are selected. The list includes customer needs. This team is committed to complete the sprint to the finish. If the team was not able to finish all the items, rather than an increase in sprint time, should be prioritized list of items remaining to be restored. During the sprint, the selected items are not altered in any way - does [2]. Every form and completed a brief session, team members work together to update report charts with an easy. In this diagram, the remaining tasks in order to perform certain moves. At the end of two hours, the team, during a meeting with other concerned stakeholders to sprint review and sprint has it that during her shows. The next Dvsrthay feedback sessions are used [9]., At the end of each sprint, Scrum cares that the product works really "done" is about software, "done" means that a code fabric, fully tested, and delivered to the customer. Roles, artifacts, and events are summarized in Figure 1, see [9]:

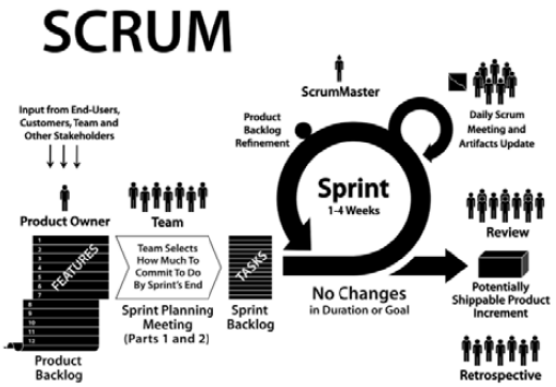

Figure 1: Methodology Scrum [9]

6. Roles in Scrum

There are three roles in Scrum: Product Owner, Team Manager and Scrum.

6.1 product owner

Product owner to identify product features, prioritized list of attributes to determine which features should be top of the list for the next sprint, and the re-prioritization and ongoing refinement of the list for the maximum return on investment, and profit and loss responsibility for the project 's. In some cases, the customer owns the product. The program - internal, such as a software company dedicated to meet the needs of their specific uses are common. In other cases, the client may be millions of people with diverse needs such as word processing software, in this case the owner of a product is a product manager or product marketing manager in many organizations. However, the product owner is somewhat different from the product manager, rather than because of his decision to resign as a director of the project, actively and frequently interacts with the field. He personally identifies priorities and results of each iteration is assessed in two to four weeks [9].

6-2 – scrum team

The product allows the customer to use it. For example, an application or a web - site. In a Scrum team - a job. It contains all the necessary expertise to provide the product is delivered to the customer in every sprint. The self-organization - is a very high degree of autonomy and accountability. The team decides that must be committed to doing what is it and how to cope. Team of five to nine people. For a software product may team of analysts, programmers, user interface designers and testers are formed. During a sprint, the team should have - specifically to work on one and only one product, so it should work on a product or multiple projects simultaneously to avoid. Higher productivity associated with the permanent teams have changed so Hdmmkn team members should be prevented. Programming with a large number of members, must be organized in multiple Scrum teams and each team will work closely with other teams to focus on different characteristics of the product [9].

6-3 - Manager Scrum

Director to achieve business value of Scrum, Scrum will help the project team to learn and apply. He will do what is teams can protect against external obstacles, and the product owner and team teaches and leads the skilled use of Scrum. Scrum Manager checks whether all team members (plus the product owner) the Scrum practices are possible to be a successful team. Manager Scrum, project manager or team manager is not, in turn, serves Manager Scrum teams, the understood and followed or not. Scrum Scrum teams need a manager to have a dedicated, full-time, although in small teams of Aghzay team could play this role [9].

7. Scrum meetings

7.1 sprint planning meeting

A one-day meeting held at the beginning of each two-hour period, it is limited and does not extend in any way. It's time to sprint four weeks - usually eight hours. Session time is divided into two parts. In the first part, the product owner's product Karanbar higher priority determines how much of the product Karanbar can do in the next sprint. The second part specifies the sprint goals accordingly, certain things split from Karanbar product, whatever the period of time estimated to constitute Karanbar sprint [2].

7.2 - sprint review meeting

The long sprint sessions for four weeks, usually four hours and not extendable. At the end of the two hour session will be held with the participation of all stakeholders. In this session, what's done in the last sprint is complete, reports. The session usually shows where the product is complete Karanbar starts. Discussing opportunities, limitations, and findings and what should be done in the future will continue. This issue potentially affects product Karanbar changes [2].

7.3 - Performance Review Meeting

Looking Jlyh Weekly sprint, there is a performance review meeting. These sessions are usually three hours long and are extendable. At this meeting, the Scrum team and manager of what has been good in recent sprint and what could be improved in the next sprint, talk. The product owner is not present [2].

7-4 - Daily Scrum Meeting

A 15 minute session every day before starting work with all stakeholders to be held on an outpatient basis. This meeting is not a solution. In this session, everyone must answer three questions: What is yesterday, today, what will it do, and what are the obstacles on his path. This session will help to avoid other unnecessary meetings - to [8].

8. Scrum Tools

8.1 - product Karanbar

A prioritized list of requirements the estimated time to complete each item is indicated. The list of the beginning of the project team in collaboration with product owners, customers and end users are prepared to be prioritized by the product owner. Owner - the beginning of each sprint priorities can change, add or delete items to the list, or change things. These items should be listed as possible, the owner of the product are simple and understandable [8].

8-2 - Karanbar sprint

The list of things that must be done during the sprint. It's possible to browse during the sprint to sprint goals change but do not change the sprint goals. These targets have already been set in sprint planning meeting. For each item of the list, when it is estimated that the time is not more than 16 hours, otherwise it should be broken into smaller tasks [2,3].

8.3 - Diagram of the remaining work

Graph of the amount of work remaining in the sprint shows every day. It's time to end sprint to zero. This chart is updated every day by a team. In practice, this graph is a straight line or a curve only downside, which may occur due to changes in sprint Knaranbar, where show increased [2].

9. Conclusions

In this paper the basic principles of Scrum methodology is briefly introduced. The important point here is that it is never - say what is the best way to do Scrum. One reason is that a situation may be different than the other team. Scrum can be faced with a different situation with a different way of doing. Another point of view, this methodology is that Scrum does not tell us exactly what we should do; Scrum defines only the general framework.

REFERENCES

[1] H.Keniberg. (2012, dec 5). Scrum and XP from the trenches [Online]. Available : http:// www.infoq.com

[2] Ramsin R., The Engineering of an Object-oriented Software Development Methodology. PHD thesis, Y.ORK. UK. Department of computer science. April 2006.

[3] G. Asproni. (2012, dec 4). An Introduction to Scrum [online]. Available: http://www.giovanniasproni.com

[4] pressman, R. S. 2006. Software engineering: a practitioner's approach. 6th ed. McGraw-Hill.

[5] ambler, S.W.,More process patterns: Delivering Large-Scale Systems Using Object Technology. Cambridge university press,1999.

[6] palmer, s.r., and j.m.felsing. 2002. A practical guide to feature-driven development. Upper saddle river, NJ: prentice Hall PTR.

[7] e.nazemi., Analysis and evaluation model for agile software development methods. Journal of Information Science and Technology. Volume 22 Issue 1 and 2 (Fall-Winter 85) :1-28

[8] P. Deemer, G. Benefield, c. Larman and B. Vodde. (2012, dec 4). The scrum Primer(version 1.2)[online]. Available: http://www.scrumfoundation.com

[9] M. Cohn (2012, dec 4). An Introduction to Scrum [Online]. Available : http:// www.mountaingoatsotware .com

[10] beck, k. 1999. Embracing change with extreme programming. Ieee computer 32(10): 70-77.