Requirements in Agile Software Development

Mahdad Khelghatdost [1*], Zahra hajilary[2], Hesam Pourkarim[3]

[1,2,3] Mazandaran University Of science And Technology
Mahdad_12@yahoo.com

Abstract: Nowadays, software organizations spend much time and resources to analyze and investigate software requirements. Thus some factors like the need to fast software delivery with a high quality, the superiority in competition with other companies, satisfying customers, considering the market dynamics, the change of customer needs, and addressing market challenges have finally led to develop agile methods as well as the requirements issue becomes one of the most important topics. As it is time consuming to define needs at one stage and impossible to spend much time in order to determine requirements completely using traditional ways in agile methods, screening and some other ways are required to identify needs in the agile method. This paper is aimed to study requirements and identify strategies of useful and useless ones to develop the software via agile methods because of the need to the software and different competitions between developers in rapid software providing perfect and with a high quality.

[Mahdad Khelghatdost, Zahra hajilary, Hesam Pourkarim. **Requirements in Agile Software Development**. Life Sci J 2013; 10(8s): 326-330] (ISSN: 1097-8135). http://www.lifesciencesite.com. 51

Keywords: Software development, agile methods, Software requirements, XP, Scrum

1. Introduction

There are two key challenges for the software engineering which lead to separate it from other engineering fields. It is clear-cut that the software is a conceptual and often intangible product. Although it could be changed, the cost increment is a function of changes in the project lifecycle.[1]

Over the last decade, many agile methods such as XP and Scrum have been put forward. While these methods have their own characteristics, they are aimed to a common goal which is responding rapidly to variations. The ability of rapid responding to changes will reduce risks and costs as it is costly to respond at the end of the project. But this is possible in agile approaches by a comprehensive understanding of requirements, strategies and methods. The present study aims to approach the strategies of determining requirements and their effects in both XP and Scrum. Indeed, the goal is to examine processes to fulfill requirements faster rather than study how it could be done.

2. The Agile Methodology Statement

In 2001, a group of 17 members who works on the agile approaches provide a statement known as the agile statement. Its concepts are in order:

- The superiority of the individualism and interaction to processes and tools.
- The superiority of the executable software to conceptual documents.
- The superiority of the collaboration with clients to contract-based negotiations.
- The superiority of the variations responding to following an initial plan.[8]

And some explanations are followed:

- The superiority of the individualism and interaction to processes and tools: A group of average programmers with good relationships usually do much better than a group of professional ones without any good relationships.
- The superiority of the executable software to conceptual documents: Software has no limited documents because it is meaningless to produce such documents. Also, excessive documents are worse than limited ones as it is time consuming to produce them.
- The superiority of the collaboration with clients to contract-based negotiations: A successful project is a project that frequently receives clients’ responses.
- The superiority of the variations responding to following an initial plan: It is not possible to produce the final product considering a preliminary framework and design, because minor and major requirements of the design often become visible during the development and so the developer team should make required changes in designing with respect to these requirements.[3]

3. The Definition of Agile Software Development

A framework is a concept that deals with the accomplishment of software engineering projects. Thus, the agile is a combination of a philosophy with a series of software developing instructions. [4]

3.1. The Overall Philosophy of the Agile Designing

The philosophy could be summarized in the form of below concepts which in turn result to the agile popularity:

1. Encouraging customers.
2. The incremental and early delivery.
3. A small but full of motivations team.
4. Informal methods.
5. The least activity.
6. Simplicity.

3.2. The Instructions of Agile Designing

Below, there are two important instructions in agile designing:

1. Enforcing in order to deliver products early instead of analysis and designing;
2. A continuous contact between the developer and customer.

3.3. The Agile Methodology Projects' Characteristics

- Being modular at the development process level;
- Being repeatable by short repetition periods,
- Scheduling by 1-6 week repetition periods,
- Scrounging in the development process by abandoning all of the unnecessary activities,
- Being adaptive by new probable risks,
- The incremental process trend which leads to the functionality of the project and making it in small steps,
- The incremental and convergence trend which decreases risks.[5]

4. Agile Methods

Nearly after 2000, various methods in agile categories have been introduced. They often seek to be committed to agile concepts. Two methods have been suggested in this section:

4.1. The XP Approach

XP is one of the most famous agile methods. It uses simultaneously some programming concepts such as designing and implementation, software testing and the client participation in the development team. Team members including the user set a meeting and define project priorities. They are usually some programmers, software testers and analysts who collaborate in the whole project. Programmers work dual and write some tests before writing the program. While one of them writes a part of program, another one controls the accuracy of codes and fixes the probable problems. This is one of the XP advantages which enhance the product software quality. [6][7]

4.2. The Scrum Approach

Nowadays, the Scrum is one of the software production ways especially for small ones. Small software could be made using this method. This way is mainly emphasizes the flexibility, compatibility and usefulness. Each member should understand his/her task in the project and follow a certain goal in all of the operational and executive phases.

As Scrum is a framework, it does not give complete and accurate details about how works should be done in the project and so assigns these affairs to the team. Actually, the Scrum manager is responsible to identify members' tasks, investigate the evolutionary process of software construction and abilities of members, and work in order to reduce the risks of project. It is one of the Scrum advantages that it could be divided a big project to small and understandable parts which could be simultaneously accomplished. [8][9][10]

5. An Introduction to Roles and Responsibilities in the Agile

Several roles are introduced in the agile. Hence, it is necessary that everyone knows his/her role and responsibility in the project. [11]

As can be seen in the below chart, roles and responsibilities in both XP and Scrum approaches are studied.

Concept Agile	XP	Scrum
Number of teams	1team tcejorp rep	1-4 or more
Team size	3-18	5-9
Team Roles	Customer, Programmer, Tester-Tracker-Coach	Scrum team(Experienced Engineer, Junior Engineer,[Tester], [writer])
Non team Roles	Big Boss	Scrum master, Product owner

Table1. The study of roles and responsibilities in both XP and Scrum approaches

5.1. Customers

In the agile, as the customer is more involved in producing the software, he/she has one of the most valuable roles in comparison with other software production ways [12]. For instance in traditional methods like the cascade method, clients just interact with developers to determine requirements at the first stage as well as they are involved at the end to receive alpha and beta versions of the project while in agile methods it is recommended that customers stay with the production and development team during the construction of the software product. In this method, it is necessary to become aware of the availability of customer representatives and their awareness of final requirements before starting. [13][14]

In agile methods, there is usually a beneficiary one as a candidate showing all who are involved in the project. He/she should be an expert and be able to make important decisions such as the acceptance of the product, prioritization, etc. [14]

5.2 Features of customers in agile

Features that customers should have agile methods: [15]

1. Availability: the customer has to be always available to answer questions

Coming from the development team. Any delay in the answer delays

The development of the product.

2. Complete knowledge: the customer is the representative for all the

Stakeholders. Therefore, he is able to answer all questions, since he is

The domain expert and knows how the application should work and the

Input / output data required. Again, this is possible if the size of the project

Is limited.

3. Decision power: the customer is able to make final decisions and commitments.

Changes in requirements, acceptance of the features implemented,

Etc. can be decided directly by the customer, allowing a fast

Decision making process.

Having available to the client, which is able to meet these needs is not easy.[16]

6. Software Requirements

For specifying software requirements is one of the main topics and the requirements of the most complete and comprehensive software product that creates a good product for the customer, or stakeholders . The following aspects the requirements to be examined.

6.1 Impact of Agile the requirements: [17]

Traditional software agreement is based on the following points:

- Design and implementation of a system should be obvious.
- A specification must be unambiguous, accurate, reliable trace is independent of the implementation.
- Requirements and models of user requirements is a valid statement.
- All the requirements have equal weight.
- Changing the requirements is higher cost than the progress of the project.

But agile processes explicitly quantitative analysis and design. Many delivery systems, business metrics do not provide. Agile attitudes to the requirements, allowing the developer to provide more business criteria [5].

Standish study group also reported that five of the eight major failures in relation to the requirements of the projects that you can see in the table below:

Problem	%
Incomplete requirements	13/1
Low customer involvement	12/4
Lack of resources	10/6
Unrealistic expectations	9/9
Lack of management support	9/3
Changes in the requirements	8/7
Lack of planning	8/1
Useless requirements	7/5

Table2. Eight major cause of failure in software projects, according to study group Standish.

6.2 Requirements Management Tools

The most popular tool in requirements engineering in agile methodologies is paper, pens and pins and board. For example, the XP requirements as user stories, is accumulated. User stories, very short, Activities that will be implemented in the programming team to explain. These stories are written on small pieces of paper and pins are mounted on the board. Board is divided into three user stories to be done, User stories are ongoing, and the stories that have been done. But there are some tools that will help development teams too.

The following are the three types of tools:

• Unified Modeling Language Tools

Two examples of the application of these tools:

1 - Write a program to higher levels

2 - Reverse engineer code to create documents

• Tools for Requirements Negotiation: This type of tools for the identification, prioritization and management agile requirements in different environments, including environments that help customers and developers.

• Instant messaging tool: To be used when the client is not the place to discuss and communicate with him through this.

6.3 Requirements Prioritization in Agile

Requirements Prioritization in Agile Requirements is one of the main topics in the field, usually requirements to have an important role in software output. In most projects that use agile methods, Prioritization criteria based on the initial output and output requirements that are needed. One of the methods of prioritizing requirements is the Kano model.

This model attempts to classify the four main requirements and take customer satisfaction, these four principles include [19] [18]:

1 - Requirements that the product should have and without it does not work.

2 - Requirements that the customer's satisfaction, but should be.

3 - That cause requirements to be greater understanding of the product and customer satisfaction is a check based marketing lists.

4 - Requirements that go beyond the normal requirements and to enhance customer satisfaction and customer are advised.

More models prioritized in a table with two variables are the features and characteristics of their customers and the customers they are valued, and the weight of the score that features the ability to see what needs - features - more value is closer to the customer.

6.4 Additional requirements and waste

In agile methods try to reduce this surplus to requirements.

Specifically, identify and reduce the additional requirements of one of the best ways to prevent the creation of additional requirements.

Effects Additional requirements include:

1 - Charge more for coding.

2 - Increased complexity in code.

3 - Delay in delivery of the final version of the software

4 - Complex and costly maintenance

5 - The increasing complexity of customer applications (for example, a more complex interface that customer is getting annoyed)

6 - Increased use of program resources such as memory, network, processing power, etc.

The only cost to the customer without additional requirements and will not do any good for him and certainly not a requirement for these projects will be done faster.

7. Result

Changing requirements in software might often be very expensive, It is a methodology that can respond quickly to changes, Can be considered a very useful tool for the project manager. In agile methods before starting a project, Profile of the project must be reviewed and evaluated, thus, to create a good product, Based on customer demands, many parameters can be involved, and Customer's team is one of the factors useful for identifying customer requirements. Tools can be introduced to increase speed , And to expedite the delivery of the first copy and deliver it to Customer - This is one of the characteristics of agile methods - Identify additional requirements and prioritize requirements is of great importance .One of the best known and most popular methods of prioritization, Kano model is. Certainly in modern times the speed of delivery , And delivery of software based on customer demands, Two important factors in the competition for the market, These two With proper understanding of requirements And activities that can easily be done by a good team.

References

1. Highsmith J., Cockburn A., "Agile Software Development: The Business Of Innovation", Computer,
   September 2001, pp120-122.
2. "Manifesto for Agile Software Development", http://www.agilemanifesto.org.
3. Rasmin R.,"The Engineering of an object oriented Software Development methodology", PhD Thesis, york.uk.department of computer science, April 2006.
4. Amber S.,"More Process patterns: Delivering large scale system using object technology", Cambridge university.
5. Paul M.C., "Agile methodologies and process discipline", Crosstalk (Oct. 2002).
6. Somerville I.," Software Engineering", 9th Edition.
7. Beck K.,"Extreme Programming Explained", Addison Wesley, 2000.
8. Schwaber K.,"Scrum development Process".
9. Pressman R.,"Software Engineering: a practitioner's approach" 6th Edition, McGraw Hill.
10. brahamsson P., Salo O., Ronkainen J., Warsta J., "Agile software development methods",VTT Publications 478 (2002).
11. Thomas S.,"Agile Roles and Responsibilities".
12. Bailey P.,Ashworth N., Wallace N., "Challenges for stakeholders in adopting XP", 3rd International on extreme Programming and agile processes in

software Engineering (XP 2000) , Alghero ,Italy ,26-29 May

13. Coran M., Bohner S., "The Impact of Agile Methods on Software Project Management", Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)
14. Cockburn, A. And J. Highsmith, "Agile Software Development: The People Factor," Computer, November 2001, pp. 131-133.
15. Sillitti A., Succi G.," Requirements Engineering for Agile methods".
16. Ebert C. Requirements BEFORE the requirements: understanding the upstream impacts. 13th IEEE International Conference
On Requirements Engineering (RE'05), IEEE Paris, France. 2005.
17. Paetsch, F., A. Eberlein, and F. Maurer, "Requirements Engineering and Agile Software Development," Proceedings of the 12th IEEE international Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June 2003, pp. 308 – 313.
18. Ullman, David G., the Mechanical Design Process, McGraw-Hill, Inc., U.S.A., 1997 pp. 105-108 ISBN 0-07-065756-4
19. Jacobs, Randy, Evaluating Satisfaction with Media Products and Services: An Attribute Based Approach, European Media Management Review, Winter 1999.

4/2/2013