RESEARCH ARTICLE   OPEN ACCESS

Survey Based Reviewof Elicitation Problems

Sidra Arshad
Department of Computer Science, University of Lahore, Sargodha, Pakistan

Abstract
Any software development process is the combination of multiple development activities and each activity has a vital role in the software development cycle. Requirement Engineering is the main and basic branch of Software Engineering, it has many phases but the most initial phase is Requirement Elicitation. In this phase requirements are gathered for system development.
This paper provides a literature review of the requirements engineering processes performed in traditional and modern development processes and analyses the problems in the requirements elicitation phase. This problem analysis is based on a survey which was conducted in University. A questionnaire posing questions regarding the problems in requirement elicitation was given to final year computer science graduate students who are working on their final year project as a requirement for their degree. The theoretical analysis of the questionnaire further clarifies the problems. This problems analysis will help to find out the main problems which are faced by the perspective software developers.
Keywords: Elicitation, Agile, FDD, DSDM, ASD, story cards.

I. Introduction

Requirement Engineering (RE) is the very initial step in software development process. The success of software is based on RE process as it is the process that finds out the need for developing software. It has five phases i.e. Requirements Elicitation, Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation, and Requirements Management. However the focus of this paper is on the "Requirements Elicitation" phase. [1]

The dictionary meanings of "elicitation" is" to gather" or "obtain". It is the most initial and critical phase of requirement engineering process in which information and requirements are gathered from stakeholders about the under developed project. The results obtained from this process should be unambiguous and clear from confusions. Any minor ambiguity can cause serious results like financial damages or deficiencies in the system, and so the system can collapse. Now a day's agile methodology are taking place of traditional methodologies. The motto of agile methodologies is to fully involve customer with the development team to reduce ambiguities upto maximum extent. For this research work empirical analysis is done through questionnaires and interviews.

The most collective challenge is communication gap. The next immense challenge is ambiguous data that is not well structured or properly stated [2].

II. Literature Review

The development of software depends upon methodologies which are to be followed for the development process. In early stages the development of software were based on traditional ways. But the main drawback of traditional methodologies is that these ways require a lot of documentations and less user involvement than agile methodologies and solead to serious ambiguities and errors that may cause failure of the newly developed system. On the contrary agile methodologies are process oriented and have the aim to deliver high quality product to its customers.

In this section different traditional and agile elicitation techniques are discussed briefly. In the next section the results of the survey will be discussed.

2.1. Traditional Methodologies
2.1.1. Interview
It is a better to identify the authorized person as an interviewee and all other concerned people before taking interview. The interviewee should be reminded before 2 or 3 days and the objective of the interview should be known to him/her. The relevant and proper questions should be properly listed Interviewing is the technique which is mostly used by the analysts. It is not as simple as it is considered as it requires social skills, ability to speak and ask questions confidently and should be to the point, and should have enough knowledge of the required field. [3]

2.1.2. Brainstorming
It is a group activity i.e. based on group of people with a leader. It is a Facilitated Application Specification Technique (FAST). In this technique all the people are allowed to generate ideas. This

technique is based on two sessions. In the first session people generate ideas as much as possible without discussing pros and cons. In the second phase all the idea are discussed and organized. Leader of the group is responsible to start the session; a person is dedicated to record all the ideas on flip charts. At the end of the session al the ideas along with their priorities are produced.

2.1.3. Scenarios

It is an example of interaction session. Software engineers imagine that how the system will be used in real life situation. They think about the behavior of the users during the usage of an application. This technique is based on the experience of the engineers. There is a main drawback that the engineer cannot imagine about the new options in the developed systems. Use cases are used in this technique.

2.1.4. Questionnaire

Questionnaire is a technique which helps to get large amount of data in a calculated way. The results acquires from questionnaires are accurate due to precision. The questions may be MCQs, fill in the blanks, yes or no, rating scales or open ended questions, depends upon the requirement. [Arif, S. 2010]

2.2. Agile Methodologies:

Agile methodologies are gaining fame during the near past. The main objective if agile is to deliver more reliable and modifiable software with less documentation and more user interaction.

2.2.1. Extreme Programming (XP)

XP is a simple and most initial methodology of agile. It was developed due to the drawbacks in the restricted development cycles of traditional methodologies. It was created by Knet Beck in 1990's. It is based on Planning Game, small releases, unit testing, continuous integration and pair programming.

1.2.2. Scrum

Scrum is comparable to a part of XP. Scrum is not as comprehensive as XP and it covers the area of planning game of XP only. However, scrum and XP can be used mutually in any development project [2]. Scrum is a framework. Most of the tasks completions are dependent upon the development team without a full documentation of the whole process of system development.

1.2.3. Crystal Methodologies

Crystal methodologies are family of different type of methodologies. One of the most suitable methodologies is selected from this family according to the project need. In this approach different methodologies of this family can be customized for unstable situations. All the members of this family are represented with different colors according to the heaviness i.e. darkness of the color is directly proportional to the heaviness of the methodology. Darker colors of methodologies are mostly suitable for larger projects. [5]

1.2.4. Feature Driven Development

FDD is an iterative methodology. The theme of which is to deliver the concrete product in the given time i.e. without any delay. [6]

This sequential methodology covers design and building phase instead of entire software development process [7]

1.2.5. Dynamic System Development Method

DSDM is the best framework for Rapid Application Development (RAD). The key point of this methodology is that time and resources are adjusted and then amount of functionality is fixed accordingly unlike other methodologies in which amount of functionality is preferred to set first and then adjustment of time and resources is done [8]

1.2.6. Adaptive Software Development

ASD is a framework for iterative development of large and complicated projects. It encourages frequent prototyping. There are three phases in this methodology i.e. Speculate, Collaborate and Learn. [9].

III. Survey Based Study of Elicitation Problem

The main focus of the survey in this research was to find out the problems during requirements elicitation phase which are faced by the customers and development team. The survey is conducted through questionnaire in which all the questions were about requirements engineering, elicitation and agile methodologies. The project students were selected for this study as they face a lot of problems during their project. The final year project is the first complete software development endeavor which is done as a requirement for the graduate degree. These students usually choose traditional software development methodologies for their project.

2.3. Data Collection Methodology

To conduct this study a questionnaire was designed and distributed among project groups. Some face to face interviews were also conducted to further analyze the problems and to clarify the answers given by the project groups. This questionnaire is meant to find out the problems in the requirements elicitation process performed under traditional software development methodologies in an attempt to define a well-organized process of requirements elicitation for

the students who do not have any prior experience of software development and fail to identify all the requirements of the system to be designed and for this reason fail to develop usable and useful software.

2.4. Results

The main emphasis of the questionnaire was on the question about the problems, students faced during problem elicitation. Following is the graph that shows different problems faced by different project students during Requirements Elicitation Phase.
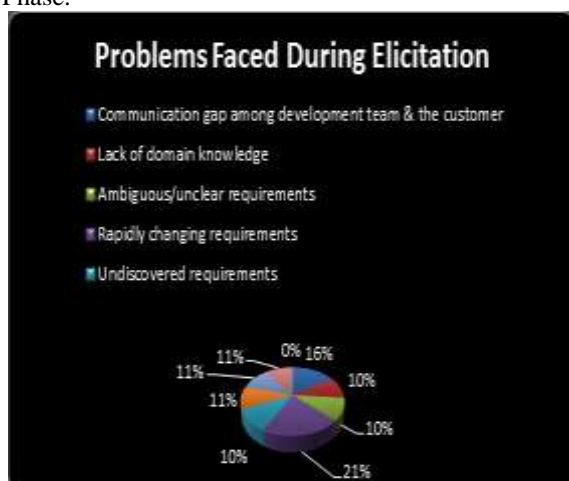

Fig 1 Elicitation Problems

If we see the above figure, it is clear that the most common problem faced by students is the rapidly changing requirements of the customers during development and the second big problem is communication gap of development team and the stake holders or customers. If agile methodologies are followed instead of traditional methodologies then these two major problems can be overcome because agile methodologies are based on customer involvement during the system development process. Further results show us that the students give priority to the agile methodologies but they are following tradition methodologies due to having lack of knowledge for implementing agile methodologies. Figure 2 and 3 are the results about the preferences of students.
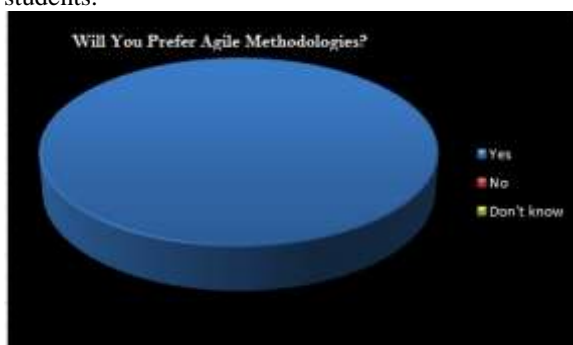

Fig 2 Preference of Agile

All of the students prefer agile methodologies during the system development but they are unaware about the practical implementation due to having lack of knowledge.

Fig 3 Model Used for Development

As the students are unaware of agile therefore they follow traditional methodologies for their software development. It is clear from Fig 3 that majority prefers to follow waterfall model, the second major model is V Model to be followed for their project development.

Fig 4 Prioritization

Although the students prefer agile methodologies but their practical implementation is based on traditional methods. Therefore from Fig 4 it is clear that they will prioritize the requirements of the customers through simple ranking rather than using story cards.

IV. Conclusion

A survey was conducted for taking a review of the problems which are faced during elicitation. According to the survey the major problems which are faced are Rapid changing requirements and communication problems. These problems are caused due to less customer involvement during development process. Further results showed us that the students who are facing these problems during elicitation are following traditional methodologies although they prefer agile methodologies but due to lack of knowledge these students cannot practically implement agile methodologies.

It is required to work more on agile methodologies and make people aware about it and blowout its knowledge about practical

implementation. Different conferences and seminars should be arranged to educate people about agile methodologies and their practical implementation,so that the problems that are faced during elicitation can be overcome and can lead to a better system development.

REFRENCES

[1] Husnain, M., M. Waseem., S.A.K. Ghayyur, (2009). An Interrogative Review of Requirement Engineering Frameworks. International Journal of Reviews in Computing Pakistan

[2] Asghar, S. and M. Umar. 2010. Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS) Components. Int. Journal Software Engineering. (Kuala Lumpur, Malaysia). 1(2): 32-50. ISSN: 2180-1320.

[3] Paetsch F., A. Eberlein and F. Mannheim. 2003. Requirements Engineering In Agile Software Development. Proc of 12th IEEE International Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises.(Mannheim, Germany) pp.308-313. ISBN: 0769519636.

[4] Arif, S., Q. Khan and S. A. K. Ghayyur. (2010). Requirements Engineering Processes, Tools/Technologies, and Methodologies.International Journal of Reviews in Computing. (Islamabad, Pakistan). Vol.2: pp. 41-56. ISSN: 2076-3328.

[5] Cockburn, A. 2001. Agile Software Development. Addison-Wesley Professional. (Boston, USA). pp. 164-173.ISBN: 0201699 699.

[6] Rychly, M and P. Ticha, 2008. A Tool for Supporting Feature Driven Development Balancing Agility and Formalism in Software Engineering.Springer-Verlag.(Berlin, Heidleberg, Germany).pp: 196-207

[7] Palmer, S. R. and J. M.Felsing.2002. A Practical Guide to Feature-Driven Development.Upper Saddle River, Prentice-Hall.1st Ed. (NJ, USA). pp. 55-57. ISBN: 0130676152.

[8] Stapleton, J. 1997. Dynamic System Development Method- The Method in Practice. Addison Wesley. (Masachusetts, USA). pp. 3-12. ISBN-10: 0201178893.

[9] Highsmith, J. A. 1999. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing.(New York, USA). pp. 37-45. ISBN: 978-0-932633-40-8.