Sci., Tech. and Dev., 32 (1): 44-47, 2013

EMPIRICAL ANALYSIS OF TRADITIONAL AND AGILE REQUIREMENT ENGINEERING PROCESS

SIDRA ARSHAD[1*], SHAH MUHAMMAD[1] AND SARA SHAHZAD[2]

[1]*Virtual University of Pakistan, Lahore, Pakistan.*
[2]*University of Peshawar, Peshawar, Pakistan.*

Abstract

Requirements Engineering (RE) plays a very important role in software development. It is the process in which the constraints, requirements, needs and deficiencies of previous system are exposed. Although an error-free software is just an ideal state but through RE process, developers can get rid of many big problems that may be faced during software development. The purpose of this paper is to expose the difficulties of the users which are faced during Requirements Elicitation. The initial most phase of RE and a framework is presented as well. A survey was conducted in a Public Sector University (i.e., University of Peshawar), Pakistan. During this survey, different questions were asked regarding the problems during Requirements Elicitation. Questionnaires were also distributed amongst students in their final semester. These students were working on software development projects for their degrees.

Keywords: Requirement engineering (RE), Survey, Requirements elicitation.

1. Introduction

Requirements Engineering (RE) has five phases viz. Requirements Elicitation; Requirements Analysis and Negotiation; Documentation; Requirements Verification and Validation and Requirements Management (Husnain et al., 2009). However, the focus of this paper is on the "Requirements Elicitation" phase.

Exploring software requirements is essential for a software development project. If the requirements are not clearly identified, it may cause serious errors and may lead to either software failure or heavy budget loss. Therefore, it is important to make the users involved throughout the development process and to ask about their requirements from time to time.

The most common challenge is communication gap. The other big challenge is ambiguous data that is not well organised or properly specified (Asghar and Umar, 2010).

It is clear from research that problem of requirements is major issue at different levels of software development. When asked about the causes of such failure, executive managers identified poor requirements as foremost of the problems. Others are lack of user involvement (13%), requirements incompleteness (12%), changing requirements (11%), unrealistic expectations (6%) and unclear objectives (5%). On the European side, a recent survey of over 3,800 organisations in 17 countries similarly concluded that most of the perceived software problems are in the area of requirements specification (>50%) and requirements management (50%) (European Software Institute, 1996).

2. Literature Review

During the study of RE, specifically in Requirement Elicitation phase, different methodologies were studied that could be adopted for gathering requirements from users. In the light of which problems of the users are tried to be explored.

- The two main types of methodologies, namely, Traditional Methodologies and Agile Methodologies, are discussed below:

2.1 Traditional Methodologies

There are different types of traditional methodologies for Requirement Elicitation but user involvement is not continuous throughout the whole process, unlike agile methodologies. Less user involvement can cause severe damage to the developing software and may lead to great financial loss.

***Author for correspondence E-mail: sidra_arshad143@yahoo.com**

2.1.1 Interviews

Interview is a technique that requires sufficient knowledge and communication skills on behalf of both interviewee and interviewer.

2.1.2 Questionnaire

Questionnaire is a technique in which list of questions is prepared. Mostly, these questions are MCQ-type. The questions included in the questionnaire are either open ended or close ended. These questions are simple and clear to the users so that the users can fill the questionnaire according to their requirements for the developing software.

2.1.3 Ethnography

Ethnographer discerns people during their work. This technique is superior for those users who feel perturbed in taking interviews and face-to-face meetings as they cannot specify their requirements verbally (Davis, 1993). The observations of ethnographer may be direct or in direct through any electronic device, like, camera (Kotonya and Sommerville, 1998).

2.1.4 Focus Groups

The main focus of this technique is to discuss the prototype of the software by different people. The needs, requirements and pros and cons of the prototype are noticed by these concerned people, having different domains and experiences in the related field (Paetsch et al., 2003).

2.2 Agile Methodologies

Agile Methodologies are iterative and involve users throughout the development process by arranging different meetings. Normally, a user representative is selected who has sufficient computer knowledge and better communication skills with the users and development team.

There are different types of agile methodologies, viz., Extreme Programming (XP), scrum, DSDM, FDD, ASD, and crystal methodology but, in this paper, only XP is discussed as the defined framework is related with XP.

2.2.1 Extreme Programming (XP)

XP is simple and initial most agile methodology. Its formation is the outcome of flaws in the limited development cycles of traditional methodologies (Beck, 1999). XP is centered on different rules or practices including planning game, small releases, metaphors, simple designs, unit testing, refactoring, continuous integrations, pair programming and 40 hours a week.

It is a technique that takes the users and developers closer for better understanding the developing system. Planning game uses the technique of story cards for requirement elicitation (Paetsch et al., 2003).

XP can accept changes in requirements, e.g., coding and design through its iterations. XP motivates communication amongst the development team and the users for getting obligatory results exactly without ambiguities. The key features of XP are communication, feedback, courage and simplicity (Kavitha and Sunitha, 2011).

Scrum is similar to a part of XP. Scrum is not as inclusive as XP and it covers the area of planning game of XP only. However, scrum and XP can be used jointly in any development project (Asghar and Umar, 2010).

Communication is highly prioritized which is done through on-site customers, regular meeting, users' stories and pair programming. However, feedback also is an important principal. XP has five phases viz. exploration phase, planning phase, iteration to release phase, productionising phase, maintenance phase and death phase.

2.2.2 Planning Game: Customers are totally involved in the development of the system by close contact with the development team. The development process depends upon short releases so that confusions and unclear requirements can be removed. Planning game is based on the synchronisation of users and programmers. Acceptance tests are used to check that the establishment works appropriately. These tests are usually written by the users to check functionality. More than one test can be performed and the story gets completed but all tests should be passed in all iterations. These tests are automated and the score is reported to the development team.

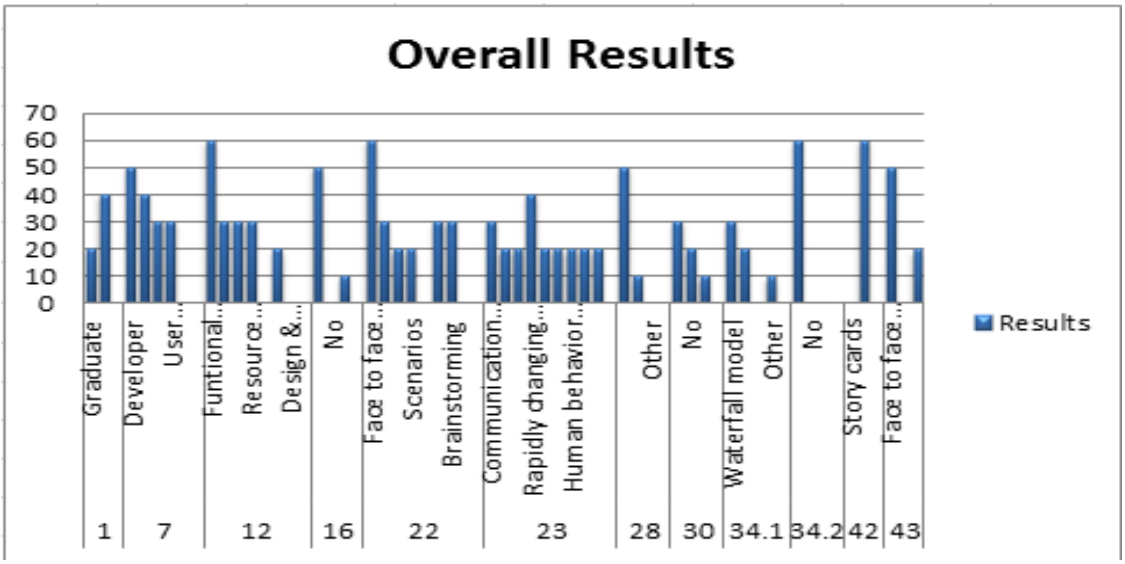
3. Data Collection Methodology

Data were collected from different groups of students of University of Peshawar through survey questionnaires and interviews. The aim of the survey was to discover the problems faced by the users/students during elicitation process.

4. Evaluation

Total 70 questionnaires were distributed, out of which 60 were correctly and completely filled by the students. According to the results, 40 were undergraduate students and 20 were graduate students. Majority of the students play the role of an analyst as well as a developer, during the development process. Similarly, most of students preferred to gather functional requirements while domain requirement and design and implementation was 0%. Face-to-face communication is preferred by heavy majority. The problem that is faced commonly by the users is rapidly changing requirements, i.e., 40/60 and the second big challenge is communication gap, i.e., 30/60. According to 83% students during the development process exactly the gathered requirements are to be followed. 33% students focused on cost/benefit analysis while 50% focused on risk analysis before starting the development process. 100% students preferred agile methodologies to be adopted for Requirement Elicitation.

Following is the graph that shows the overall result of different questions answered by the students. Y-axis shows the number of students while X-axis shows the number of questions in the given questionnaire.



5. Framework

According to the survey, 100% students preferred agile methodologies to be adopted during software development but they have not much awareness about agile. During interviews, some technical expertise was explained to these students regarding agile methodologies and, therefore, a framework was developed regarding XP as this methodology is not complicated and can easily be adopted by the students.

The framework depends on XP methodology and the use of story cards. The use of story card is better for collection of unambiguous requirements. All the problems, which are faced in previous system and for which the new system is going to be developed, should be properly defined and sorted according to their complexity and possible solutions. When the story cards are collected, they should be prioritised according to the priority numbers given by the users.

After collection and prioritisation, the requirements should be compared with the list of sorted problems of existing system to see, if the fulfillment of these requirements will remove the problems of previous system. Otherwise, a meeting should be arranged for further discussion regarding to the solution of the existing problems.

Following are the figures for two sides of the user story card that may be used for requirement elicitation. These story cards can be printed out or can even be manually designed for the collection requirements.

User Story Card	Priority # ----------

Requirement Title: _____

Description: _____

Is this requirement negotiable in case of any conflict?

a. **Yes**

b. **No**

Fig. 3.1. Front Side of Story Card Template

Acceptance Criteria

What are the acceptance criteria of your requirement?

a. _____

b. _____

Fig. 3.2. Back side of Story Card Template

Future Work

The survey was conducted in a Public Sector University of Pakistan and the developed framework is according to the skills and level of students. This survey and framework can be extended to larger organisations and multiple institutions for getting more information about the problems of the users.

References

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.

Beck, K. 1999. Extreme Programming Explained. Embrace change. Reading Mass, Addison-Wesley.

Davis, M.A. (1993). Software Requirements: Objects, Functions, and States, 2nd Edition, ISBN-10: 013805763X, ISBN-13: 978-0138057633

European Software Institute 1996. "European User Survey Analysis", Report USV_EUR 2.1, ESPITI Project, January 1996.

Husnain, M., M. Waseem and S.A.K. Ghayyur. 2009. An Interrogative Review of Requirement Engineering Frameworks. *International Journal of Reviews in Computing Pakistan*, 2: 1-8..

Kavitha, C.R. and M.T. Sunitha. 2011. Requirements Gathering for Small Projects Using Agile Methods (IJCA Special Issue on "Computational Science - New Dimensions & Perspectives"). (India). pp. 122-128.

Kotonya, G. and I. Sommerville. 1998. Requirements Engineering: Processes and Techniques, (Michigan, USA). ISBN: 0471972088, 9780471972082.

Paetsch F., A. Eberlein and F. Mannheim. 2003. Requirements Engineering In Agile Software Development. Proc. of Twelfth IEEE International Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises. IEEE Computer Society. (Germany). pp. 308-313.