

Changing Requirements – Correlated to Risk or Quality?

Dr. S. Arumuga Perumal and Ms. G.Kavitha

Abstract—The requirements are the high-level description of the functions provided by the system. Most of the IT projects experiences the changing requirements. Although the analyst paid a full attention to collect the complete set of requirements, the requirements will still change after the analysis phase. There are many reasons for the requirements to change in the development process. The changing requirements should be considered as a chance to improve the quality of the product and mitigate the risk of project failure. This paper discusses the reasons for change and reasons that the changing requirements are good for the software project and the Change Control Management.

Index Terms—Requirements, Changing Requirements, Quality, Risk.

I. INTRODUCTION

The requirements for a system are the descriptions of the services provided by the system and its operational constraints [1]. A successful software project should satisfy all the requirements. According to the CHAOS Report of Standish Group, the factor that caused most of the projects to be cancelled was incomplete requirements. The factors that turned out most of the project to be challenging were incomplete requirements and specifications, and changing requirements and specifications [2]. The requirements will not be stable throughout the software development life cycle. The users often tend to change the requirements because of many reasons like technology change, environment change, etc.

This paper is analyzing the reason for software requirements change. Section 2 describes the processes involved in software requirements engineering. Section 3 discusses the reasons for change, how requirements change reduces the risk and improves the quality.

II. REQUIREMENTS ENGINEERING

Software Requirements Engineering is a process of discovering, refining, modeling and specifying the requirements [3]. The above definition clarifies that the processes of requirements engineering is to collect the requirements from the end user, document the requirements, refines the requirements to validate that the requirements are clear, complete, traceable and so on.

The common practices in requirements engineering are:

- 1) Requirements Elicitation.
- 2) Requirements Specification.
- 3) Requirements Validation.
- 4) Requirements Management.

1) Requirements Elicitation

The initial step in the requirements engineering process is to collect the requirements from the stakeholders. Kathy Schwalbe [4] in her book IT Project Management states that

stakeholders are the people involved in or affected by the software project activities. The software project team has to know what the user wants in the system. Ian Sommerville [1] in his book Software Engineering states that the requirements are the high-level abstract statement of the service that the computer system should provide and also it is a detailed description of the system's functions. Ian Sommerville [1] categorizes the requirements as user requirements to represent the detailed description of the services that the system will provide and system requirements to represent the high level abstract statement of the system's functions.

The requirements are also categorized into functional, non-functional requirements and domain requirements. Functional requirements stands for the services that the system is expected to provide in future. Non-Functional requirements represent the criterion on the services of the system. Domain requirements are from the application domain of the system and they reveal the uniqueness and criterion of the domain [1].

The system analyst can apply any of the Requirements Gathering Techniques are used to collect the requirements from the users.

✓ Interview

It is the formal meeting between the system analyst and the user. This includes question and answer session to reveal the high-level requirements. Interviews give an opportunity for the analyst to clarify some topic in greater detail. Interview is also called as qualitative method.

✓ Questionnaire

Questionnaire is paper-based or electronic-based method to collect requirements from the user. The users are required to fill-in a list of questions of either open-ended or closed questions. Questionnaire is also called a quantitative method as it can cover a large number of users.

✓ Joint Application Development (JAD)

The JAD technique is an extended workshop used to gather large amount of high-level requirements in a short period. The analyst can have a multiple points of view of a topic. JAD is very useful to solve any inconsistencies at once. JAD requires a facilitator to plan, organize and manage the JAD session. It also needs a recorder to documenting the points discussed during the session.

2) Requirements Specification

Software requirements specification is a written document containing the written descriptions and graphical models. The requirements specification is the final work product of the requirements engineering process and it describes the functions of the software and the constraints that will manage the software development [3].

3) Requirements Validation

The requirements need to be validated to ensure that the

gathered requirements are complete, clear, consistent, testable, and traceable and confirms to the business standards [3].

4) Requirements Management

Requirements management is a set of activities that helps the project team to identify, control and track the changes to the requirements at any time during the software development process [3].

III. CHANGING REQUIREMENTS

A. Characteristics of good quality requirements

The good quality requirements also represent the non-functional requirements. Ivan Marsic [5] in his book "Software Engineering" states that the non-functional requirements represent the system properties not the system functions. The good quality requirements should exhibit the following attributes:

- 1) Complete – No missing information.
- 2) Unambiguous – One clear meaning.
- 3) Consistent – Non-Conflicting with other requirements.
- 4) Correct – Accurate to the stakeholders need.
- 5) Feasible – Realistic to implement.
- 6) Modifiable – Easy to change.
- 7) Necessary – Useful and needed for the stakeholders.
- 8) Traceable – Easy to identify.
- 9) Verifiable – Easy to test.

B. Reason for change

The reasons that lead the systems analyst to collect poor quality requirements are as follows:

Sometimes, the users do not know what they want in the new system. In other cases, the user will not express their needs to the systems analyst because they did not ask the question to the user. In The other reasons are: the user may have a poor understanding of the new computer-based system or problem in communicating the needs to the systems analyst or forgot to specify the important needs to the analyst or providing ambiguous or conflicting requirements to the analyst.

In other cases, people who are knowledgeable and know the business process better may not be involved in the requirements capture process. In some other situations, not all the business has the written business operating procedures and it may reside in people's mind. Failure to express the imperative information will direct to requirements change.

Allan Kelly pointed out in his paper titled "Why Do Requirements Change?" that there are lots of possibilities for mistakes in the requirements elicitation process. The mistake can occur when the analyst comprehends the requirement, or when recording the requirement or when communicating the requirement to the developer.

Although the systems analyst paid a full attention during the requirements process to gather the quality requirements, the stakeholders may tend to change the requirements during the development process. There are many reasons for changing the requirements after it has been collected, validated and documented.

The reasons behind software requirements change according to Edberg and Olfman [6] are:

1) External Change

The changes occur in the environment external to the organization. This may require changes in the system. For example, new requirements from the integrated systems (for example: a Pharmacy Information System that connects to a Medicine Vendor. When the Medicine Vendor changes the data format, the Pharmacy Information System should also change the data format also).

2) Internal Change

The changes occur in the environment internal to the organization. For example, changes in the management, new policies, new products, and new services.

3) Technical Change

The changes occur in the technical environment requires changes in the system. For example: system's development in latest software or development of online software.

4) Learning

The changes occur because of the learning process of individual or a group. When a person gets knowledge about his/her work activities, he/she needs change in the software to support the activities. For example, the officer established a new special way of doing calculation; this new calculation needs to be incorporated into the system.

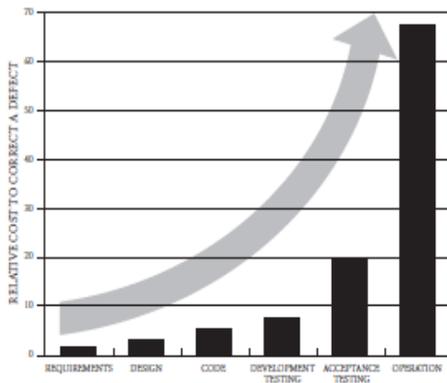
C. Mitigating Risk

Varying requirements is advantageous for one reason that the risk of software failure will be reduced. People often think that the changing requirements have bad impact to the software development process. The software development team needs to look at the optimistic part of changing requirements as it facilitates to reduce the risk of project failure and increase the quality of the resulting product.

Getting the wrong requirements and developing a wrong system is the biggest risk to the project failure. All the requirements cannot be identified completely and correctly in the start of the software project as the users does not know about the software and the analyst does not know about the business. As the development proceeds, the user and the developer will get a clear picture. The developers have to accept the fact that the complete set of requirements cannot be documented during the start of the project and the requirements are going to change as the development progresses [10].

D. Improving Quality

The CHOAS Report [2] state that the major reason for the projects cancelled or impaired is incomplete requirements. Studies reveal that an error identified during post production stage takes nearly eight hours to correct whereas an error recognized during requirements phase takes roughly 15 minutes only [7], as cited in Borland White Paper, [8].



(Source: Barry W. Boehm, Software Engineering Economics)

Although the changing requirements are likely to increase the estimated cost and time, they definitely will improve the quality of the software. Most of the studies / researches are finding the ways to improve the requirements engineering process to acquire quality requirements. As stated by Allen Kelly [9], the requirements specification documents the users' needs that have been collected before the software project starts. But, as the project proceeds, both the software developer and the user will have more clear understanding of the business activities and the software respectively. This may even give greater value to the system.

As mentioned by Allen Kelly [9], instead of giving bad impact to the changing requirements, it should be considered as an opportunity. The varying requirements give the developer a chance to improve the quality of the software. The good quality software compliance with the users needs with no defects and meets the intended purpose. The changing requirements are vital for the business not to be avoided as it might had been forgotten or ambiguously mentioned during the requirement process. The requirements might also been changed due to the technology or environment change. So the software development team should always welcome the requirements change as it will certainly increase the success of the project by satisfying the users.

The solution to accommodate the changing requirements is to have regular meeting between the developers and the users during the development process. The CHOAS Report [2] indicates that the user involvement is the primary factor for the project success. Increased user involvement will increase the software quality also.

IV. CHANGE CONTROL MANAGEMENT

Change is inevitable [12]. As a consequence of many factors as described in section 3, the change in requirements is unavoidable. The change in requirements is needs to be considered for review either to be accepted or rejected based on its impact on the project. When the requirements changes, change control process has to be followed to find out whether the change affects other requirements, or affects the scope or time or budget of the project. Depends on its impact, the change can be accepted or rejected [11].

The change control management should include the following activities:

- 1) Elicit the requirements change
- 2) Analyze the impact of the change

- a. Depends on its impact, either accept or reject
 - b. Collect the justification if the change is accepted
- 3) Prepare and submit a Change Proposal
 - 4) Review the change proposal
 - 5) Upon acceptance, all the parties affected are informed about the change [13] and the project activities are adjusted according to the change proposal.

By having and following a change control management, not all the changes will be accepted. Only the changes having impact on the project will be accepted. This will lead to a successful project development. If the changes are accepted without a change control process, this will lead to project failure as it may direct the project in another direction.

V. CONCLUSION

The changing requirements are always considered as a factor of software risk as it requires rework which in turn escalates the project's scheduled time and estimated cost. At all the times, the varying requirements need to be considered as an opportunity to improve the product quality as it compliance with the stated needs, and reduce the risk of software failure as it aids the project team to develop the right system. Accommodating the changes in the requirements after the change control process will always leads the development team in a right path. According to Edberg and Olfman (2001), 40% of the changes in the requirements are because of the learning. So its time for the development team to understand that the changes in the requirements are unthreatened to the project's success.

REFERENCES

- [1] Software Engineering, Ian Sommerville, Seventh Edition, Pearson Education Limited, 2005.
- [2] The CHAOS Report, The Standish Group, 1995.
- [3] Software Engineering – A Practitioner's Approach, Roger S Pressman, Sixth Edition, McGraw Hill, 2005.
- [4] IT Project Management, Kathy Schwalbe, Thomson Course Technology, 2006.
- [5] Software Engineering, Ivan Marsic, Rutgers University, 2008.
- [6] Organizational Learning through the Process of Enhancing Information Systems, Edberg & Olfman, Proceedings of the 34th Hawaii International Conference of System Sciences, 2001
- [7] Software Engineering Economics, Barry W. Boehm, Prentice Hall, 1981.
- [8] Mitigating Risk with Effective Requirements Engineering, Borland White Paper, 2005.
- [9] Why Do Requirements Change?, Allan Kelly, ACCU Publications, 2004.
- [10] Mitigating Requirements Risk with Agile Processes, Dave Chaplin, Byte Vision Ltd., 2002.
- [11] Daily Challenges of Requirements Engineering, Frank Salvatore, NDIA 11th Annual Systems Engineering Conference, 2008.
- [12] A Brief Overview of the New Project Management Body of Knowledge, Samuel Brown, Fourth Edition, 2008.
- [13] 800 Change Control, Jefferson Science Associates.

Dr. S. Arumuga Perumal received Ph.D in Computer Science from Manonmaniam Sundaranar University (India) in 2004. At present, he is the Head of Department of Computer Science in S.T.Hindu College, Kanyakumari, India. He has 23 years of teaching experience. He published nearly 29 papers in National and International Journals. He is a member of

IEEE, Senior Member of Computer Society of India, Member of IACSIT, and Life Member of System Society of India. He received Best Teacher Award for the year 2008-09 by Tamil Nadu State Council for Higher Education, Chennai (India) for his contribution in teaching and research.

Ms. G.Kavitha received M.Phil. in Computer Science from Mother Teresa Women's University, Kodaikanal (India) in 2001. She is the Chairperson of Department of Computer Science, Jubail University College, Female Branch in Saudi Arabia. She has nearly 13 years of teaching experience. She has presented her research papers in 4 International Conferences. She is a member of Computer Society of India, Member of IACSIT, and Member of ACM.