

Change in Requirement Engineering – A Survey

¹Syed Sarmad Ali, ²Hafiz Fahad Hassan, ³Syed Abdus Sami

¹Dept. of Computer Science & Engineering, Beihang University, Beijing, China.

And Dept. Of School of Computing, Coventry University, Coventry, UK

²Department of Research, Director, TxLabz, Lahore, Pakistan

³Department of Management, Karachi University, Karachi, Pakistan

ABSTRACT

Background: In software engineering, the change in requirement engineering poses a greater impact in improving software development life cycle approach. Requirement changes during SDLC cause a risk to budget, plan, time and standards. The major reasons for system fail are requirement modifications.

Objectives: The objective of our research study is a comprehensive analysis based on a survey of different studies conducted in the past. Our paper will highlight the methods being used in the change process in the requirement phase during the progression of the system. **Methods:** In this paper, our intention is to discover the strategies of modification in the requirement and consequence on other stages. For this purpose, we have followed the procedures defined in evidence-based software engineering. **Results:** We have identified the related literature/models and empirical studies. **Conclusion:** We have identified different methods and their impact on entire frameworks of software development stages. Our research would have a significant advantage for the proposition of the framework to handle the change requirement.

Keywords

Software Engineering, Requirement Engineering, Change Management, SDLC

1. INTRODUCTION

In software engineering, the requirement engineering phase (RE) in the software development process (SDLC) is the most crucial phase.[1][12][14] During the life of software project, requirement engineering has the major influence till the completion of the software project. When a modification arises during the development of current requirements, its influence is not restricted to a specific segment. Nevertheless, it has a substantial influence on other succeeding segments of SDLC [8]

The changes in the software product might occur during the development of the software project. The changes have a major impact and importance in the accomplishment of a software product. The organizations are bound to make modifications according to the changes advocated by the clients

[13]. In general, if the software systems change earlier and are according to customer requirement, will lead to minor cost. Throughout the software life cycle, various different artifacts are created, used, changed and deleted. Artifacts can be scenarios, use cases, practical requirements, architectures, designs, code or other consequences generated throughout the creation of the final product. When a modification is brought in the system, a number of these artifacts and artifact types will be influenced [1]. Information originates during a change in requirement can be used to guess the number and type of artifacts to indicate to do a change or not. At a similar period, it can be used to develop a perfect image of all affected artifacts to confirm that no one of the systems other functions is affected by the change. The modification also affects the system functionalities and the user desires, precisely in response to development in the perception and understanding with the continued evolution. So, the method of requirement gathering desires to determine the kind of problems that are produced by different types of requirement modification [2].

We observe the circumstances that arose due to the requirement amendment and compared to alternative methods to deal with them. The target organization and the requirements of users who have been the system design actually, because these provide the ultimate reason for change [3]. Successful and unsuccessful software system basically depends on the way of handling change requirements. Evidence about the motives of changing requirements can increase our proficiency to make effective judgments and handle requirements changing resourcefully.

For in-depth analysis, we have followed the guidelines defined in the evidence-based software engineering of Barbara Kitchenham [9]. We do not claim that we have a systematic literature review, as it needs to describe on a large scale. However, we have tried to find out the frameworks that are being used when there is a change in the requirement of SDLC. We are planning to present SLR on our next research article in a journal publication.

The rest of this paper is organized as follows. Section 2 will include the motivation of the research work requirement evolution; Section 3 will introduce the methodology of the

research study, Section 4 outlines the in-depth analysis of the changes in the requirement and their impact along with the methods and framework that have been used in order to encounter these changes. Moreover, we will introduce the discussion in Section 5. Section 6 will include the findings that have been identified during the research work. Conclusion and Future work have included in Section 7.

1.1 Motivation

A software alteration demand can have many explanations; e.g. the originators may overlook to indicate a requirement and it is a misdirected feature in a subsequent structure [12] [13], Similarly there might be a situation where fault/error/bugs have been recognized, the modification of which is a requirement [14], or misinterpretation of requirements due to the deficiency of active participant contribution [13] [14].

Changes and evolution are such an important part of software development and maintenance. Change analysis helps to guess the parts of the software where changes need or a part that may be affected if a change is made. In this paper, investigation regarding effective changing and evaluating software development life cycle. We have tried to learn the problems that are being faced by the development team after the changes in the user requirements. Throughout the software life cycle, many different artifacts are created with respect to the developers' final goal and that's a modification to affect the system functionalities and user needs [4]. Requirements change during software development causes a risk. The major cause of system failure is requirement modification. The successful and unsuccessful system mostly relies on the way of responding to modification and requirement evaluation. The understanding of the reasons for requirements modification can increase our capacity to make healthier choices and handle requirements modification efficiently [5]. So, the procedure of requirements gathering plays a significant role and will help the researchers to discover the variety of problems that are raised up by many types of requirement changing.

2. Related Work

Change in the requirement engineering is a very important and core field when it comes to Software Development Life Cycle. The abundance of research has been made in the past with respect to change in RE. There has been significant research work have been done in the field. Requirement engineering elicitation techniques have been discussed in papers [16] [17]. A detailed systematic literature review has been given by Muneera et al. [18] in which they found 3251 papers and after elimination of irrelevant paper they found 8 most relevant papers. Similarly, the systematic literature review to identify and classify software requirement errors [19]. Taxonomy for foundations of Requirements Variation was suggested by McGee and Greer [26].

3. Methodology

The research methodologies used in this study is to extract methods that are being used in the domain of requirement change management from newly published books and research papers. This work created a list of initial sources of research data. For the identification of the related research, we have followed the methodology defined in the evidence-based software engineering which was introduced by Barbara Kitchenham [9] in 2004. Different data repository used to investigate how much work has been done in the field of requirement engineering with respect to change request. For our SLR we have a limit of search and we only focused on the recent studies i.e. from 2005 to 2018.

PHASES FOR SURVEY

We had divided our research work into three steps as shown in *figure 1*. In the first phase of our SLR, we have identified research studies according to their title. By reading the title we have selected all the most relevant research papers into our pool. Afterward, we read the abstract of all the selected papers from the pool and select all the most relevant research papers. In the next step, we have read introduction along with skimming of the whole paper in the final round. Thus after the three rounds, we found most relevant research papers, as shown in Table 2. In the last and the final stage, we selected all the paper by reading all the content of the research studies.

a) Research Question

We have created the research question under the PICOC guidelines as described in [10]

- *Population* – Change in RE in Software Development related studies.
- *Intervention* – RCM approaches/methods
- *Outcomes* - The accurateness of the requirement change management procedures/practices.
- *Context* - Any case study/article/practices which are in the context of Change in RE

| |
|--|
| <i>Question 1 - What methods/techniques have been used</i> |
|--|

| |
|---|
| <i>Question 2 – Is there any framework used for RCM</i> |
|---|

b) Search strategy

After defining the research questions, we have set up a strategy to define search string. For the searching purpose we have created research strings that are as follows:

| Search String | Keywords |
|---------------|---|
| S1 | “change in requirement engineering.” |
| S2 | “modification of requirement engineering.” |
| S3 | “Altering in requirement engineering.” |
| S4 | “modification in requirement during development.” |
| S5 | “Software requirement change management.” |

Table 1: Search strings used during Research Study

c) **Databases to be searched**

Following are some of the most reliable databases in the Software Engineering domain. We have search our research articles from the following data repositories

- *IEEE Digital Library*
- *ACM Digital Library*
- *Elsevier*
- *Springer Link*

d) **Inclusion Criteria**

For the inclusion criteria only that journal papers and conference proceeding were included that are related to our project. Following are the inclusion criteria defined for the cost estimation SLR.

- Provide the literature related to change in requirement engineering or modification in requirement engineering AND
- All the frameworks, models, must be in the change in requirement engineering.
- The literature and related practices found in the conference proceedings OR journals OR workshops OR technical workshops AND
- All literature must be found in English language AND

e) **Exclusion Criteria**

For the exclusion criteria all the papers that are irrelevant with the focus of the study. All the studies and research evidences were neglected whom goal is not related or have the following complications:

- Literature or research evidence that do not present change in requirement engineering or modification in requirement engineering OR
- Literature that have the language other English language.
- Literature Studies without experimental evidence.

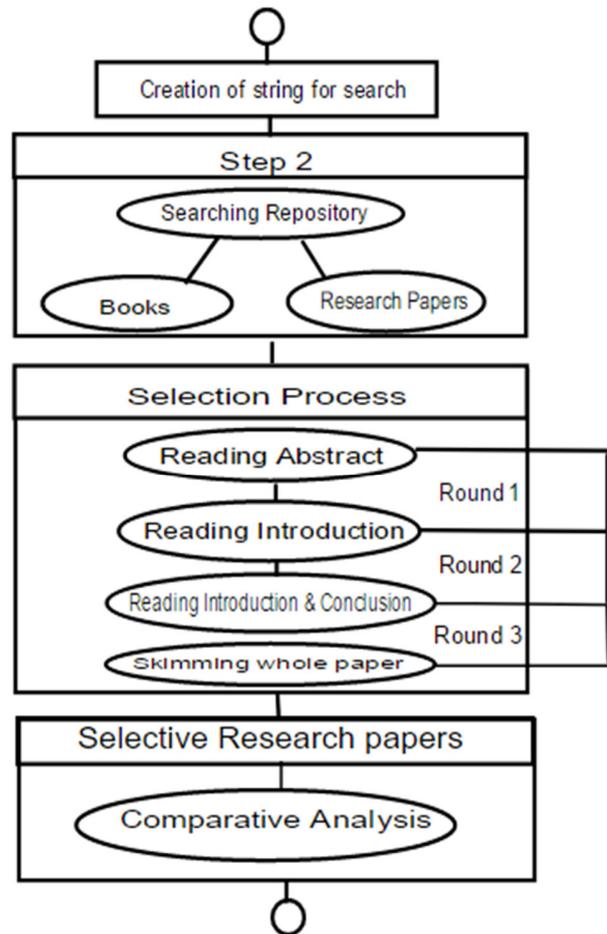


Figure 1 Selection Process Flow

3.1 Results

After the inclusion and exclusion based on the research question formulated, we have found sufficient number of research papers that are related to requirement change management.

Table 1 shows different keywords that were used during our research study. By using S1, we have found 60 research papers in the first round. We read the titles of found papers and in the second round, we identified 8 relevant papers, after eliminating process. In the final round, we identified 3 most relevant papers.

Similarly, using S2 research string, we have found 46 research papers in the first round and after eliminating by relevance in the final round we identified only one most research papers.

Furthermore, using S3 from Table 1, 30 research papers were identified. In the final round (S3), we succeeded with 3 relevant research paper.

We have found 35 research papers of research string S4, in the first round. We identified 3 most relevant paper. In the end, we fail to find any relevant paper using S4 research string.

Initially, in the first round, we have found 80 papers, by reading the titles of research papers, using S5. Furthermore, we have found 12 relevant papers in the second round after eliminating irrelevant papers. Finally, we have identified only 2 most relevant papers

Table 2 shows the three rounds which we have utilized in order to find the relevant papers. After that procedure we read the most relevant papers while reading the research papers, we highlighted the certain points and write that specific points on papers then we summarize this written document into our words.

Knethen, A.V. [6] proposed that if the requirement is not correct or not clearly defined then changes will affect the time and cost. Errors efficient approach focus on functional system requirements. It consists on concept how to change, how to evaluate, how to analyze the changes, how to modify systems after delivery of the system to correct errors, to add new functionalities, to increase the performance of the system or to handle any other changes according to the customer. Similarly, Bano .et .al [4] proposed how to identify and handle the changing and the steps how to detect errors in requirements during development to improve the software quality. The major reason for system flop is requirement modification and unclear requirement. They executed SLR and complete all phases of SDLC. These causes can arise during any stage of development and cause to modification in the project overall requirement where they need. A review of the overall system was developed and some quality standards were developed and applied on the system to ensure value, perfection, and accuracy. Hillman Willis .et .al [7] proposed the process of executing vendor evolution & aspects of changing. Its strategy is to reduce time, reduce cost, and develop a good system with least resources. Willis suggested JIT strategy, JIT said that the accurate need is offered in the exact place in the best quality at the correct time. There are many aspects that are used in buying and developing a process and describe their benefits & drawbacks to develop a new system that's the reason we use the existing system to develop a new system.

William J. Kettinger .et .al [1] proposed the Reengineering process, change methodologies, tools and techniques for modifying requirement based on unique project features is offered to support for those project actions & methods to be modifiable. This flexible technique can brief survey of S-A frameworks, methodologies tools & techniques for evolution to improve changeability and to fulfill customer requirements. Mcgee .et .al [3] proposed about evolutions and change during SDLC can cause a risk to cost, time and quality but if we change existing system to make a new system then it's providing an opportunity to save cost, time and gives a good quality to the customer. Change in existing concepts for new system needs a brief set of requirements from customer and easily needs to collect common requirements change basis data across multiple projects.

Jalote, P. [2] proposed that the procedure of requirements evolution & used to guess the number and type of artifacts to choose to do a change or not. It can also be used to develop a

perfect image of all affected artifacts to confirm that no one of the systems other functions is affected by the change.

Bhatti et al [8] examined the impact of requirement change and analyzed that the major changes are proposed during the maintenance phase rather than any other phase of the SDLC. They further analyzed that the major there is a directly proportional relationship between the change request in the requirement phase and the change occur in the design phase. Changes suggested in the design segment and changes required in testing segment similarly have a substantial association and this noteworthy relation characterizes that if additional modifications are demanded in the design segment then the modifications in the testing segment would be reduced.

Harker.et.al [5] proposed how to handle a change of requirement & process of development. In this paper, they describe the difficulty of handling change in SDLC. If user & organizational requirements are clear then the developer can develop according to customer need easily and can do easily change in the existing system for making a new system and many other different methods to deal with changes.

Hillman Willis .et .al [7] proposed aspects of changing according to the vendor. Its strategy is to reduce time, reduce cost, and develop a good system with least resources. Willis suggested JIT strategy, JIT said that the accurate need is offered in the exact place in the best quality at the correct time. There are many aspects that are used in buying and developing a process and describe their benefits & drawbacks to develop a new system that's the reason we use the existing system to develop a new system.

IDENTIFIED PAPERS

| STRIN G | ALL PAPERS (Round 1) | RELEVANT PAPERS (Round 2) | MOST RELEVAN T PAPERS (Round 3) | PAPERS TITLES |
|------------|----------------------------|---------------------------------|---|---|
| S1 | 60 | 8 | 3 | 1). The change and evolution of requirements as a challenge to the software engineering. [5] 2). Change-oriented requirement traceability [6]. 3). A study of Methodologies tools and techniques [1]. |
| S2 | 46 | 5 | 1 | 1). Causes of Requirement change a systematic literature review.[4] |
| S3 | 30 | 9 | 1 | 1). Software Requirements and evolution in a just-in-time environment.[7] |
| S4 | 35 | 3 | 0 | |
| S5 | 80 | 12 | 2 | 1). Software Requirements Analysis and Specifications. 2).Causes of requirement changes a systematic literature review. |

Table 2: Found Relevant Research Papers

Frameworks Identified for Change Management

| Research Papers | Framework | Description |
|---|-----------|--|
| A Study of Methodologies, Techniques, and Tools [1]. | Yes | (1). Imagine [start organization idea, determine to reengineer, find IT levels, and select Procedure] (2). Begin [update Participants, Reengineer groups, organize Plan, ask client requirements, set aims] (3). Identify [study the existing process, analyze existing Process] (4). Restructure [define and analyze Ideas, model and DD, HR structure, analyze and design] (5). Rebuild [Reorganize, implement, user training, eliminate] (6). Estimate [system efficiency]. |
| Software Requirements Analysis and Specifications [2]. | No | |
| Software Requirement Change Source Taxonomy [3]. | Yes | (1).Initial trainings (2).Structural Environment (3). Workspaces (4). Categorization (5).Classification [idea improvement]. |
| Causes of requirement change a systematic literature review [4]. | Yes | (1) .Evaluate [assess value, cost, requirement, client knowledge] (2). Identify (3). Select (4). Design (5).Conduct (6). Develop |
| Literature The Change and evolution of requirements as a challenge to the practice of software engineering [5]. | Yes | (1).Unchanging requirements [stable requirements] (2).Modifying requirements [changeable, evolving requirements, resultant requirements, flexible requirements]. |
| Change-oriented requirements traceability. Support for the evolution of embedded system [6]. | Yes | (1). Understandability (2).Analyze (3). Changing (4). Implement. |
| Requirements and evaluation in a just-in-time environment [7]. | Yes | (1).Presentation attribute [Financial(Financial stability, Price, size), Service, Technical] |

Table 3: Framework identified for change management.

4. Conclusion

There are clearly recognized many issues related to the requirement which undergoes modification when SDLC takes place. Some techniques which are existing & which are used already to handle changes have been discussed. Requirements changes during SDLC cause a risk to budget, plan, time and standards. The major reason for software system failure is requirement modification. The successful and unsuccessful system basically depends on the way of handling change requirements. We describe in this paper our aim in requirement changing focus on identifying and using techniques efficiently and to decrease the schedule, money, and work.

We have categorized reasons of requirements modification (e.g. “incomplete requirements specification”, “unclear requirement”) etc. Many methods are necessary to identify & used to handle many kinds of changes and its effects. Any of them changing technique is unusable without identification of the accurate requirements.

5. Future Work

We do not claim that we have done the systematic literature review which is one most exhaustive literature survey. Our next publication will consist of the SLR of all the papers that have been published so far with respect to the change in requirement engineering. However, we have tried to find out the frameworks that are being used when there is a change in the requirement of SDLC.

6. REFERENCES

- [1] Kettinger, W. J., Teng, J. T., & Guha, S. (1997). Business process change: a study of methodologies, techniques, and tools. *MIS Quarterly*, 55-80.
- [2] Jalote, P. (2005). Software requirements analysis and specification. An integrated approach to software engineering, 79-158.
- [3] S. McGee, D. Greer, “A Software Requirements Change Source Taxonomy”, *proc. 4th Intl. Conf. Software Engineering Advances*, 2009, pp. 51-58.
- [4] Bano, M., Imtiaz, S., Ikram, N., Niazi, M., & Usman, M. (2012, May). Causes of requirement change-a systematic literature review. In *Evaluation & Assessment in Software Engineering (EASE 2012)*, 16th International Conference on (pp. 22-31). IET.
- [5] Harker, S. D., Eason, K. D., & Dobson, J. E. (1993, January). The change and evolution of requirements as a challenge to the practice of software engineering. In *Requirements Engineering, 1993.*, Proceedings of IEEE International Symposium on (pp. 266-272). IEEE.
- [6] Von Kethen, A. (2002). Change-oriented requirements traceability. Support for evolution of embedded systems. In *Software Maintenance, 2002. Proceedings. International Conference on* (pp. 482-485). IEEE.
- [7] Hillman Willis, T., & Huston, C. R. (1990). Vendor requirements and evaluation in a just-in-time environment. *International Journal of Operations & Production Management*, 10(4), 41-50.
- [8] M. W. Bhatti, N. Ehsan, "An Investigation of changing requirements with respect to development phases of a software project", 2010 International Conference on Computer Information Systems and Industrial Management Applications, pp 323-327.
- [9] B.A. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering, in: Proceedings of the 26th International Conference on Software Engineering, (ICSE'04), IEEE Computer Society, Washington DC, USA, 2004, pp. 273–281.
- [10] B. Kitcheham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” Keele University, Tech. Rep., 2007.
- [11] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Wiley-Blackwell, 2005.
- [12] N. Nurmuliani, D. Zowghi, and S. P. Williams, “Using card sorting technique to classify requirements change,” 2004.
- [13] G. P. Kulk and C. Verhoef, “Quantifying requirements volatility effects,” *Science of Computer Programming*, vol. 72, no. 3, pp. 136-175, 2008.
- [14] L. Lin and J. H. Poore, “Pushing requirements changes through to changes in specifications,” *Frontiers of Computer Science in China*, vol. 2, no. 4, pp. 331-343, 2008.
- [15] S. D. P. Harker, K. D. Eason, and J. E. Dobson, “The change and evolution of requirements as a challenge to the practice of software engineering,” in *Requirements Engineering, 1993.*, Proceedings of IEEE International Symposium on, 1993, pp. 266- 272.
- [16] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. oreno, “Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review,” 2006.
- [17] O. Dieste, M. Lopez, and F. Ramos, “Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques,” in *Proceedings of the 11th Workshop on Requirements Engineering*, 2008.
- [18] Bano, M., Imtiaz, S., Ikram, N., Niazi, M., & Usman, M. (2012). Causes of requirement change - A systematic literature review. *EASE*.
- [19] G. S. Walia and J. C. Carver, “A systematic literature review to identify and classify software requirement errors,” *Information and Software Technology*, vol. 51, no. 7, pp. 1087-1109, 2009.