Requirements Volatility in Software Maintenance

D. Kavitha[1,*] and Ananthi Sheshasaayee[2,**]

[1] Lecturer in Computer Science department,
Nazareth College of Arts and Science,
Chennai, India
d.kavithamoorthy@gmail.com
[2] Computer Science department
Quaid-E-Millath Govt. College for women (Autonomous)
Chennai, India
ananthiseshu@gmail.com

Abstract. Requirements volatility is a common phenomenon present in most software development projects. Change Management dealing with requirement changes is an important function of project managers. If the changes are not handled effectively, then there will be huge difference in efforts, cost, and quality of the Product which results in project delay or project may be failed. Taxonomy of requirements change consists of three components: Change Type, Reason, and Sources. Changes in requirements are additions, deletion or modifications of requirements. These changes to the requirements after the basic set has been agreed to by both clients and maintainers are known as requirement's volatility. Requirements volatility cannot be avoided, but we have to understand the requirements volatility problems to deal with the impact. In this paper we have reviewed the requirement volatility, identified the reasons and sources of changes, and introduced few guidelines to managing changes effectively.

Keywords: Software maintenance, requirement volatility, types of requirement change, reason for requirement change and sources of requirement change.

1 Introduction

Change Management is a set of activities is developed to identify the change, control the change and ensure the change is properly implemented. Changes are indentified and classified into different types under software maintenance.

The requirements change management is concerned with the processes that are used to manage changes to the software requirements. Most of the customers' requirements become clarified during the preliminary study and requirements definition, but it is normal that the requirements change during the project. Therefore, the change management is a very important part of the requirements management.

We have reviewed the requirement volatility, identified the reasons and sources (origin) of changes. Requirement changes are recorded by the change management using the software change request form.

* Research Scholar Bharathiyar University.
** GUIDE, HOD & Associate Professor.

1.1 What is Requirements Volatility?

Changes in the requirements after the basic set has been agreed to by both clients and software maintainers are known as requirement volatility.

Requirements volatility is a common fact that is present in most software development projects. Many research published the problems that has been identified in the requirement volatility and also new approaches has been proposed to manage its impact on the software development. If the project managers were not able to manage changes, results in project delay or project may be failed.

When project managers use any tool for documenting requirements and whatever the category of execution model, requirements are bound to change. The only difference is to what extent the change results or impact on project success or failure. If there is a good and effective management mechanism, then there will be success in the project execution.

Requirements are the foundations of the software release process. Requirements provide the basis for estimating costs and schedules as well as developing design and testing specifications. During the execution of the software maintenance, requirement change is agreed by both the client and the software maintainers, which will impacts the cost, schedule, and quality of the resulting product.

Change Requests can be raised by variety of people including users, system operators, maintainers, and external interface teams. Throughout the software release process, the requirements often change in the changing world. During the software release planning, requirements analysis, design, and test reviews, new priorities are established and changes to the release content are requested in the form of change requests being added and/or deleted from the release.

In the maintenance environment, requirements are grouped through change requests that identify feature additions and/or defect corrections. Requirement change is then categorized by requirement type, by the requirement management. A change is one of the important factors that need to be considered in managing requirements change.

Identify the reasons and sources (origin) of requirement changes for review.

2 Review of Literature

The IEEE Standard defines “Software maintenance as the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment” [8].

The IEEE Standard for Software configuration management Plans is concerned with the activity of planning for Software configuration management (SCM) [9].

Change Management is also called software configuration management (SCM), is a set of activities are developed to a) identify change b) control change c) ensure that change is being properly implemented , & d) report changes to others who have an interest [16].

Software engineering literature is based on “the assumption of fixed requirements”; this misunderstanding leads managers to believe that they should freeze the requirements before project starts.

However, contrary to this traditional belief most software tends to evolve and requirements change both in the beginning as well as later stages of software development lifecycle.

During the maintenance process of the software projects, Stark has examined the impact of requirements volatility [19]. To deal with the impact of the requirement volatility, we have to identify the requirements volatility problems.

As per Harker and Eason [7] to manage requirements change, we need to classify requirements changes in the proper way. When the software developer classifies the changes, they can easily analyze each type of change and also source for the change and its impact on software development process.

In our maintenance environment a *requirement* is defined as *an approved change request* [19]. According to Stark [19], if we improve the prioritization of requirements during release planning we should be able to reduce the number of releases with added requirements. And also deletions should drop if more accurate staff estimation is there.

Evolution of requirements refers to changes that take place in a set of requirements after initial requirements engineering phase [1]. Thus changes in requirements that may occur in initial elicitation, analysis, specification and validation phases are not evolutionary.

According to Jones [11] more than 70% of large applications (i.e., over 1000 function points) experience requirements volatility.

As per Sommerville, Requirement change is the major root cause of the software project failure [18. Software requirement changes are inevitable and the factors that include software and system requirement, market demand, business goals, work environment and government regulation [2].

Requirements volatility (RV) is generally considered as an unwanted property. It has the possible to produce poor impacts on the software development process [20]. Requirements volatility causes major difficulties during development of the development life cycle. For example, a field study conducted by Curtis et al [5] point out that requirements volatility is one of the major problems faced by most organizations in the software industry. Boehm and Papaccio [4] have stated that requirements volatility is an important reason for software project cost goes beyond the limit.

Pfleeger recommends that “We must find a way to understand and anticipate some of the inevitable change we see during software development” [15].

Literature survey was conducted by McGee and Greer [17] on software requirements change source taxonomy. The authors have provided the information about the requirements change by describing the Change domain, triggers and uncertainties. The change domain can be viewed as the platform of possible classification or broad categorization of changes that occur in requirements.

Triggers are the events that cause the requirements to change whereas the uncertainties are the cause of some event to happen that act as triggers for requirements change. The authors have cumulated a comprehensive list of causes and uncertainties that gives rise to requirements change.

3 Taxanomony of Requirement Change

Taxonomy of requirements change consists of three components: Change Type, Reason, and Origin [13].

a) Change Type: Changes in requirements are additions, deletion or modifications of requirements. According to Stark, Adding new requirements, deleted or modified to the existing requirements after the basic set of requirements have been agreed upon, is a common problem in software maintenance. These changes to the requirements after the basic set has been agreed to by both clients and maintainers are known as requirement's volatility [19].

Requirement addition – Adding a requirement to make up for the omission or meet the customer's requirement.

Requirement deletion – deleting or removing existing requirements from the business strategy or the requirements redundancy.

Requirement modification – modifying requirement owing to technical restrict or design improvement.

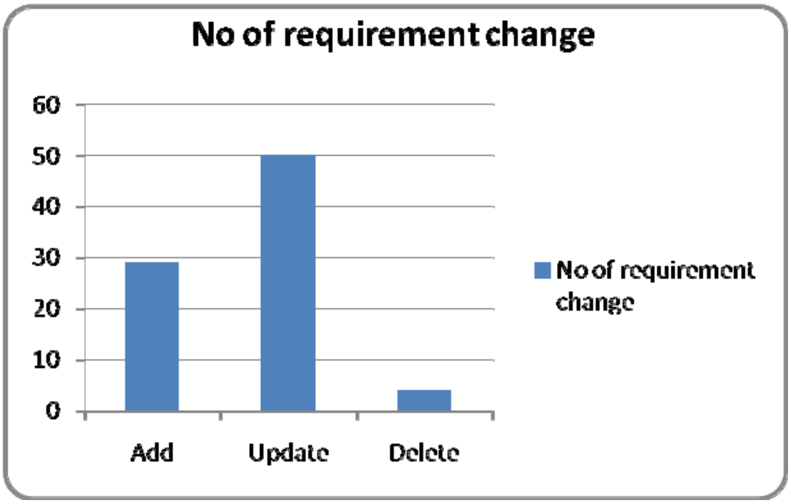

Fig. 1. Distribution of requirement changes by type. [19]

From the above empirical analysis, addition type of requirement changes is taking part the maximum in the software requirement change.

Software Requirement Types are categorized in the below table.

Table 1. Software Requirement Types and Detailed Needs

Requirement Type	Detail Category
Computational	Incorrect Operand in Equation
	Incorrect Use of Parentheses
	Incorrect/Inaccurate Equation
	Rounding or Truncation Error
Logic	Incorrect Operand in Logical Expression
	Logic Out of Sequence
	Wrong Variable Being Checked
	Missing Logic or Condition Test
	Loop Iterated Incorrect Number of Times
Input	Incorrect Format
	Input Read from Incorrect Location
	End-of-File Missing or Encountered Prematurely
Data Handling	Data File Not Available
	Data Referenced Out-of-Bounds
	Data Initialization
	Variable Used as Flag or Index Not Set Properly
	Data Not Properly Defined/Dimensioned
	Subscripting Error
Output	Data Written to Different Location
	Incorrect Format
	Incomplete or Missing Output
	Output Garbled or Misleading
Interface	Software/Hardware Interface
	Software/User Interface
	Software/Database Interface
	Software/Software Interface
Operations	COTS/GOTS SW Change
	Configuration Control
Performance	Time Limit Exceeded
	Storage Limit Exceeded
	Code or Design Inefficient
	Network Efficiency
Specification	System/System Interface Specification Incorrect/Inadequate
	Functional Specification Incorrect/Inadequate
	User Manual/Training Inadequate
Improvement	Improve Existing Function
	Improve Interface

b) Reason for Requirement Change

The identified reasons for change according to Nurmuliani are;

- Defect Fixing,
- Missing Requirements,
- Functionality Enhancement,
- Product Strategy,
- Design Improvement,
- Scope Reduction,
- Redundant Functionality,
- Obsolete Functionality,
- Erroneous Requirements,
- Resolving Conflicts and Clarifying Requirements. [13] [14] [17].

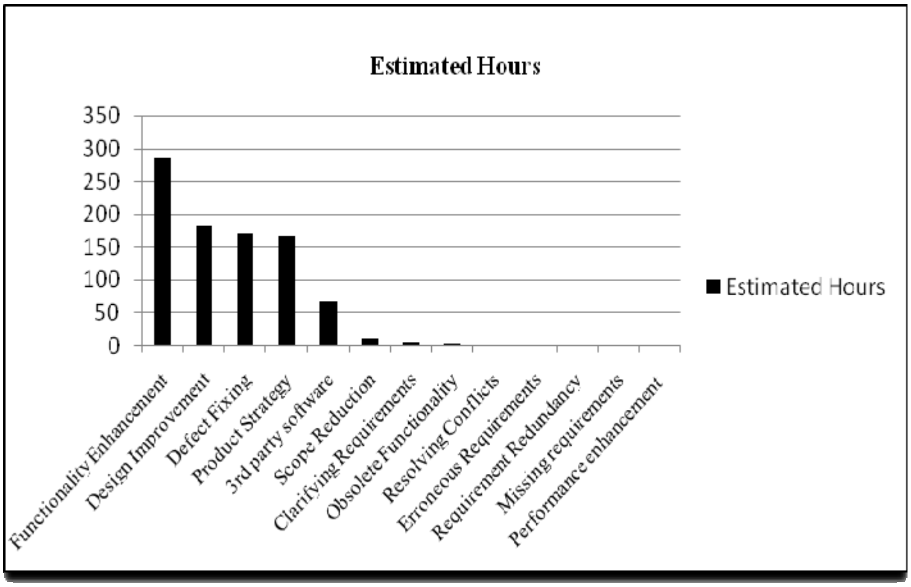

Fig. 2. Reason for Requirement change. [19]

Reason for change has been identified, and the estimated hours have taken from the previous analysis. [19]

It is important to note that these identified reasons for change were mentioned on the basis of previous study, and by no means are inclusive of all the possible reasons for change. Furthermore other reasons have been reported in the literature by other authors such as McGee & Greer et al [17], in their survey on requirements change.

c) Sources for Requirement Change

Sources of requirement change may be any of the following:

- Defect Reports,
- Engineering's Call,
- Project Management Consideration,
- Marketing Group, Developer's Detailed Analysis,
- Design Review Feedback,
- Technical Team Discussion,
- Functional Specification Review,
- Feature Proposal Review, and
- Customer-Support discussions.

These main causes of requirements change can be viewed as the root causes which influence a change in the requirements.

The connection between these sources of change and the reason for change is the fact that these sources require some change in area of 'what' needs to be changed which is covered by reasons for change which in turn are related to the root causes which were considered or which influenced this change. It is important to note that these origins or sources of change were identified based on a particular case study and it is possible that other cases may reveal other sources of change.

In the technical report, Joost says that size is a factor which determines the flexibility showed either by the software development team or by the organization, while dealing with small requirements changes [10].

Joost [10], has listed out the **source of factors** of requirement volatility,

- Internal and external (with client and users) communication and relationships. Inadequate and poor communication could be one of the reasons causing requirements to change.
- Means of communication. Close face-to-face communication makes that changes in requirements are communicated more clearly.
- Presence and influence of client and users. Having the user and/or client present during the project makes communication and the detection of changes easy.
- Project/product size. The size of the project (in terms of budget and manpower for example) and product (Complexity and KLOC) is an important factor.
- Organization size. The size of the organization (number of employee and annual turnover will gives the size of the organization).
- Development methodology. Some development methodologies inhibit flexibility while others advocate it.
- Outsourcing. Whether (part of) the project is outsourced can influence the way change is managed and communicated.
- Project Management (formalization of documents, tools present, etc.).
- Project team Flexible, self-organizing, small teams with a flat hierarchy and experienced team members tend to cope well with requirements change.

Requirement change is due to changes in government policies, business goals, market demands and work environment. Requirement engineers should get knowledge to classify, whether the requirement change has minor impact or major impact on project before managing it.

There are certain issues that should be kept in mind while changing in requirements like traceability, dependency among requirements, prioritization of requirements and decision making to implement that change [6].

In the thesis Mundlamuri says that, requirement volatility is not a problem, there should be a proper approach while managing the impact of requirement volatility [12].

Chua says that, estimating the cost of effort rework on each change is costly, hence making requirements changes will also be expensive [3].

4 Guidelines to Manage Changes Effectively are Introduced

1. Recognize that change is inevitable, and plan for it: change procedures for the project should be planned.
2. Baseline the requirements: baseline provides a clear concept for identifying the set of requirements, which are used in a design phase. It provides mechanisms for distinguishing which set of requirements is old and which requirements have changed or evolved after the base lining.
3. Establish a single channel to control change: for example, in large systems a Change Control Board (CCB) who share the responsibility about the approval of the change requests. In small systems, the responsibility can be given, for example, to someone who is an owner of the artifact or a person who has an overall understanding of system requirements.
4. Use a change control system to capture changes: a change control system should be used to capture all requested system related changes. Change requests should be transmitted to CCB for the decision making.
5. Manage change hierarchically: changes to the requirements should be managed in a top-down hierarchical fashion. For example, changes to the specification can cause changes to the features or to design, implementation and test.

5 Conclusion

Requirements change is a fact of life. The change management is only found in the software industry background, and getting the company data is very difficult. Change request records and change request forms are the main sources of information in case study. Taxonomies of Software requirement volatility have been reviewed and few guide lines to manage changes effectively are introduced. Analyze the requirements volatility data with respect to the type, reason and source.

An accurate measurement program is useful for preventive and controlling requirement's volatility. Requirements change due to defect fix, product Strategy, missing requirements are considered to be of the expensive change type.

Our future work is to reduce the rework effort related to the requirement change in the software maintenance.

Reference

1. Anton, A.I., Potts, C.: Functional Paleontology: System Evolution as the User Sees It. In: 23rd (IEEE) International Conference on Software Engineering, Toronto, Canada, pp. 421–430 (2001)
2. Barry, E.J., Mukhopadhyay, T., Slaughter, S.A.: Software Project Duration and Effort: An Empirical Study. Information Technology and Management 3(1-2), 113–136 (2002)
3. Chua, B.B., Verner, J.: Examining Requirements Change Rework Effort: A Study. International Journal of Software Engineering & Applications (IJSEA) 1(3) (July 2010)
4. Boehm, B.W., Papaccio, P.N.: Understanding and Controlling Software Cost. IEEE Transactions on Software Engineering 14(10), 1462–1477 (1998)
5. Curtis, B., Krasner, H., Iscoe, N.: A Field Study of the Software Design Process for Large Systems. Communications of the ACM 31(11), 1268–1287 (1988)
6. Gorschek, T.: Requirements Engineering Supporting Technical Product Management. Ph.D. Thesis, Blekinge. Institute Of Technology, Ronneby (2006)
7. Harker, S.D.P., Eason, K.D.: The Change and Evolution of Requirements as a Challenge to the Practice of Software Engineering. In: Presented at Proceeding of IEEE International Symposium on Requirements Engineering (1993)
8. IEEE Standard for Software Maintenance. IEEE std 1219-1998 Software Engineering Standards Committee of the IEEE Computer society (1998)
9. IEEE Standard for Software Configuration management. IEEE STD 1219-1998 Software Engineering Standards Committee of the IEEE Computer society (1998)
10. de Wit, J., Ponisio, M.L.: Technical Report, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente (February 8, 2008)
11. Jones, C.: Assessment and Control of Software Risks, pp. 93–98. Prentice-Hall International, Englewood Cliffs (1994)
12. Sudhakar, M.: Managing the Impact of Requirement Volatility. Thesis Department of Computing Science, Umea University, SE-90187 Umea, Sweden (2005)
13. Nurmuliani, N., Zowghi, D., Fowell, S.: Analysis of Requirements Volatility during Software Development Life Cycle. In: IEEE Proceedings of the 2004 Australian Software Engineering Conference (2004)
14. Nurmuliani, N., Zowghi, D., Williams, S.P.: Requirements Volatility and Its Impact on Change Effort: Evidence-based Research in Software Development Projects (2006)
15. Pfleeger, S.L.: Software Metrics: Progress after 25 Years? IEEE Software 25(6) (2008)
16. Pressman, R.S.: Software Engineering A Practitioner's Approach, 6th edn. (2005)
17. McGee, S., Greer, D.: A Software Requirements Change Source Taxonomy (2008)
18. Sommerville, I.: Software Engineering, 6th edn. Addison-Wesley (2001)
19. Stark, G., Oman, P., Skillicorn, A., Ameele, R.: An Examination of the Effects of Requirements Changes on Software Maintenance Releases. Journal of Software Maintenance Research and Practice 11, 293–309 (1999)
20. Zowghi, Offen, R., Nurmuliani, N.: The Impact of Requirements Volatility on Software Development Lifecycle. In: Presented at the International Conference on Software, Theory and Practice (ICS 2000), Beijing, China (2000)