

# Importance of Requirement Management: Requirement Engineering Concern

T. Rajani Devi

Computer Science and Engineering, Kakatiya University, Hanamkonda, Andhra Pradesh, India

**Abstract:** *Requirement engineering is first phase of software development processes and it is most important phase for every software development model. In requirement engineering phase we can gather the requirements from user and use this requirement to software development and produce software product that satisfy the user needs. In other words, we have to see all requirements during the software's life cycle, whether they are important and necessary for our software at present time or they are not important for the software currently but will become important in future. Requirements engineering , aims is to recognize the stakeholder' requirements and their verification then gaining agreement on system requirements, is not just a phase completed at the beginning of system development not required any more, but includes parts of next phases of soft ware engineering as well. To achieve this purpose, we acquired a comprehensive knowledge about requirements engineering. First, we defined requirements engineering and explained its aim in the software production life cycle. The main activities and purpose of each requirements engineering activity is described. Moreover, the techniques used in each activity are described for a better comprehension of the subject. In this research paper we describe the fundamental description of requirement engineering and present the basics dimensions of requirement engineering. Also, in this research paper we also give the basic idea of software requirement specification and present the concept of why requirement management is important for software development.*

**Keywords:** compatibility, Unambiguous, Dimensions, stakeholder

## 1. Introduction

Requirements engineering is a process based method for defining, recognizing, modeling, linking, documenting and maintaining software requirements in software life cycle that helps to understand the problem better. In other words, requirements engineering (RE) means that requirement for a system are defined, managed and tested systematically. Even though requirements engineering has a fairly narrow goal – to determine a need and define the external behavior of a solution. It seems to be a challenge for organizations. Introducing requirements engineering is a change of behavior and culture and not just a change of process and technology. It is also important that the cost of fixing a requirements defect later in the development stage is much higher than the cost of identifying and fixing it in the early stages of development. In order to do this the system requirements must be properly identified, analyzed and reviewed early in the development process. Requirements engineering is such a process that focuses on discovering, analyzing, documenting and managing system requirements. Several techniques have been employed in requirements engineering process to guarantee the completeness, compatibility, unambiguousness and correctness of requirements. Success of each software project is measured by considering level of fulfilling the project's goals. Requirements engineering is process of discovering these goals that is completed by identifying stakeholder, their needs and documentation of these needs in a way that responses to analyzing, making communications and implementing them in future .But it should be noted that requirements do not specify the way of implementation.

## 1.1 Definitions of the Terms

The term requirement engineering is used to describe a systematic process of developing requirements through an iterative, co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained. Requirement Engineering is a transformation of business concerns into the information system requirements, “WHAT” the system needs in order to achieve the organizational goals. Requirements engineering process, is the other key term used to describe the decomposition of RE into interacting non- linear activities. These proceed from informal, fuzzy individual statements of requirements to a formal specification that is understood and agreed by all stakeholders. The final term requirements engineering effectiveness is used as the measure of the accuracy and completeness of achievement of the RE process goals.

## 2. Dimensions of Requirement Engineering

The effectiveness dimension is captured in such a way that it can be translated into meaningful quantitative statements concerning quality, cost and time schedule. It shows the description of how a system should behave. Requirements are also knowledge of application domain and constraints on operation of a system. Requirements management is the process of managing changes to the requirements. Requirements of a system change to reflect the changing needs of stake holders. They also change due to change in environment, business plans and laws. Requirements engineering is a process of discovering the needs of stake holders and documenting them for analysis, communication

and implementation. Many errors can be detected in the requirements phase. Davis [8] claims that fixing of errors detected in later stages of software development is more expensive than the initial stages. If errors are not detected in the requirements phase it leads to wrong product development. Wrong requirements can also lead to wastage of valuable resources. Collecting requirements is not an easy task. Requirements engineering has critical problems which can be due to lack of stake holder's involvement in the requirement process. Lack of requirements management skills also leads to bad requirements engineering. Unclear responsibilities and communication among stake holders can also lead to bad requirements engineering. Functional requirements or behavioral requirements define functions of the product. Functional requirements include input that the software gets and output it generates. Non-Functional requirements or non-behavioral requirements are the properties of software such as portability, reliability, testability, efficiency and modifiability. Requirements are developed through requirements engineering. Requirements engineering is a process which include a set of activities such as requirements elicitation, requirements analysis and requirements negotiation and validation. This process adopted to derive, validate and maintain a system requirements document. Requirements management is the agreement between software development organization and the customer. Both reach An agreement by stating. Communicating, reviewing and negotiating requirements. Ambiguous requirements, addition of requirements, less specification and insufficient user involvement are reasons for bad requirements generation.

### 3. Requirement Engineering Tasks

#### 3.1 Inception

During inception, the requirements engineer asks a set of questions to establish A basic understanding of the problem The people who want a solution The nature of the solution that is desired The effectiveness of preliminary communication and collaboration between the customer and the developer Through these questions, the requirements engineer needs to:

- 1) Identify the stakeholders
- 2) Recognize multiple viewpoints
- 3) Work toward collaboration
- 4) Break the ice and initiate the communication

#### 3.2 Elicitation

Eliciting requirements is difficult because of Problems of scope in identifying the boundaries of the system or specifying too much technical detail rather than overall system objectives:

Problems of understanding what is wanted, what the problem domain is what the computing environment can handle (Information that is believed to be "obvious" is often omitted) Problems of volatility because the requirements change over time.

Elicitation may be accomplished through two activities:

- 1) Collaborative requirements gathering
- 2) Quality functional deployment

#### 3.3 Elaboration

During elaboration, the software engineer takes the information obtained during inception and elicitation and begins to expand and refine it Elaboration focuses on developing a refined technical model of software functions, features, and constraints. It is an analysis modeling task Use cases are developed, Domain classes are identified along with their attributes and relationships. State machine diagrams are used to capture the life on an object. The end result is an analysis model that defines the functional, informational, and behavioral domains of the problem.

#### 3.4 Negotiation

During negotiation, the software engineer reconciles the conflicts between what the customer wants and what can be achieved given limited business resources Requirements are ranked (i.e., prioritized) by the customers, users, and other stakeholders Risks associated with each requirement are identified and analyzed Rough guesses of development effort are made and used to assess the impact of each requirement on project cost and delivery time Using an iterative approach, requirements are eliminated, combined and/or modified so that each party achieves some measure of satisfaction.

#### 3.5 Specification

A specification is the final work product produced by the requirements engineer. It is normally in the form of a software requirements specification. It serves as the foundation for subsequent software engineering activities, It describes the function and performance of a computer-based system and the constraints that will govern its development, It formalizes the informational, Functional and behavioural requirements of the proposed software in both a graphical and textual format.

#### 3.6 Validation

During validation, the work products produced as a result of requirements engineering are assessed for quality. The specification is examined to ensure that all software requirements have been stated unambiguously inconsistencies, omissions, and errors have been detected

and corrected the work products conform to the standards established for the process, the project, and the product. The formal technical review serves as the primary requirements validation mechanism. Members include software engineers, customers, users, and other stakeholders.

### 3.7 Requirements Management

During requirements management, the project team performs a set of activities to identify, control, and track requirements and changes to the requirements at any time as the project proceeds. Each requirement is assigned a unique identifier. The requirements are then placed into one or more traceability tables. These tables may be stored in a database that relate features, sources, dependencies, subsystems, and interfaces to the requirements. A requirements traceability table is also placed at the end of the software requirements specification.

### 3.8 Related Tasks from Other Disciplines

According to the OPEN Process Framework (OPF) Repository Organization, the preceding tasks clearly fall completely within RE. However, there are three other tasks that technically and logically belong to other disciplines, but which nevertheless are absolutely critical to the success of the requirements engineering effort. In some organizations and according to some development methods, this is why these tasks are included within RE. In either case, they must be properly addressed when developing a RE method. These tasks include:

- Scope Management is the management task that manages requirements changes that could significantly change the scope of the endeavor.
- Requirements Verification is the quality engineering task that controls the quality of the requirements and other requirements work products such as requirements models and requirements specifications.
- Requirements Configuration Control is the configuration management task that manages and evaluates the impact of proposed changes to baselined requirements and other requirements work products.

### 4. Importance of Requirement Engineering Tasks

Requirement Engineering is logically comprised of many important tasks, not just the three or four that are most often cited. Depending on the specific needs of individual projects, the RE teams should ensure that their RE method contains all of the appropriate tasks and each of these tasks should be tailored appropriately. Once selected and tailored, the RE tasks should be performed in a manner that is consistent with the project's chosen development cycle, and this typically means iteratively, incrementally, concurrently, and constrained by appropriate time-boxes. Only by

understanding all of the RE tasks can the development team ensure that the necessary RE tasks are appropriately staffed, scheduled, and performed. Finally, RE is a complex and often messy process in practice. Hopefully, the preceding summary of RE tasks will help you better perform requirements engineering and ensure that no important work slips through the cracks.

### 5. Requirements Engineering Product

The product of requirements engineering is software requirement specifications. In other words, it is a document that describes requirements of functions, limitation of design, functionality and qualitative specifications of software and external interfaces clearly and exactly. Things to be considered when software requirement specifications (SRS) are being written are as follows:

Nature, environment, qualities, preparation, evolution, making prototype, prepared design and finally project requirements improvised in software requirement specifications.

Qualities of good software Specifications includes being revisable, correctable, traceable and usable in operation and maintenance phase.

### 6. Conclusion

As explained in previous sections, necessity of defining the requirements lies in the need to pass different phases. This necessary action for defining requirements engineering, whose goal is to identify requirements of stockholder, their validation and providing agreement on requirements of developing system, is not just a phase to be completed at the beginning of system development process that is not required any more, but includes parts of next software engineering phases as well. Importance of requirements engineering becomes apparent in special aimed software which require high costs for maintenance and support in long time. For developing special aimed software, we should categorize, combine and prioritize functional and non functional needs, co-requisites, prerequisites and different requirements. Requirements engineering process helps to categorize requirements. In other words, we have to see all requirements during the software's life cycle, whether they are important and necessary for our software at present time or they are not important for the software currently but will become in future. Using requirements engineering completely and correctly, software developer organizations confront less fails and they will not lose their capital, profit, marketing and resources.

### References

- [1] Alan M. Davis. 1994. Software Requirement Revision Objects: Functions & States, Prentice Hall PTR.

- [2] Kotonya, G. Sommerville. 1997. Requirements Engineering, John Wiley & Sons.
- [3] IEEE Computer Society. 1994. IEEE Recommended Practice for software requirements Specifications, New York.
- [4] Kotonya,G. sommerville,I.1992. ViewPoints for Requirements Definition, software engineering journal. 7 5thSASTech 2011, Khavaran Higher-education Institute, Mashhad, Iran. May 12-14. Nuseieh,B. Easterbrook,S., Requirement Engineering: A Roadmap, London, Toronto.
- [5] Westfall,L. 2005-2006.Software Requirements Engineering : What, Why, Who, When, and How , Westfall Team.
- [6] Young, R. R., Effective Requirements Practices, Boston, MA: Addison-Wesley, 2001.
- [7] Hooks, I. F., and K. A. Farry, Customer-Centered Products: Creating Successful Products through Smart Requirements Management, New York: AMACOM, 2001.
- [8] Robertson, S., and J. Robertson, Mastering the Requirements Process, Harlow, UK: Addison-Wesley, 1999.
- [9] Markert, C., "Partnering: Unleashing the Power of Teamwork," 2002, briefing available from markert@facilitationcenter.com. See also Frank Carr et al. Partnering in Construction: A Practical Guide to Project Success, Chicago: American Bar Association Publishing, 1999.

### Author Profile

**T. Rajani Devi** has completed my M.Tech (S.E) from Kakatiya University. And working as assistant professor in Chaitanya P.G College, and also working as part time faculty in informatics department at Kakatiya University, Warangal, Andhra Pradesh, India.

# IJSR