



Requirements Engineering: A Survey

Vivek Shukla

Department of Information
Technology, Babasaheb Bhimrao
Ambedkar University, Lucknow

Dhirendra Pandey

Department of Information
Technology, Babasaheb Bhimrao
Ambedkar University, Lucknow

Raj Shree

Department of Information
Technology, Babasaheb Bhimrao
Ambedkar University, Lucknow

ABSTRACT

Software development process is the collection of a variety of phases. Requirement engineering is one of phases that involve in software development processes. From all the phases, Requirement Engineering phase play an important role for every software product. During requirement engineering phase, development team interacts with representative of stakeholders and uses these requirements to build up software product that satisfy the stakeholder's needs. After collecting all the requirements, these requirements can be divided into two categories. First category will take requirements those are important and necessary for software development at present time while the other category take requirements those are not important for the software product at present but can be important in future. The motive of requirement engineering is to find out exact requirement of the stakeholders with the help of step by step predefined procedure. Collected requirements can be first verified by stakeholders. At last, stakeholder and developer team sign contract. It cannot be concluded that requirement engineering phase completed at the beginning of the product development but incorporates parts of next phase of software engineering. In this paper, researcher tries to give basic information about software requirement engineering phase and present why requirement engineering is important for software development?

General Terms

Software Engineering

Keywords

Requirement engineering, types of requirements, requirement engineering framework.

1. INTRODUCTION

Software development life-cycle is made up of various phases. Requirement engineering (RE) is the beginning phase of software development life-cycle. This phase plays an important role for the success of software product as all other phases rely on requirement engineering. Phases that depend on requirement engineering are designing, coding, testing and maintenance. RE is an interactive process which continues interactively until the exact requirements of customer is defined. Requirement engineering process conducts at abstract level with a motive of to develop definite, steady, complete requirements specification of a required product. Therefore, requirement engineering is the procedure of defining, documenting and maintaining requirements after the consultation of customer. Further, it is also observed that requirements are the foundation for every software product that represents the requirements of stakeholders – users, suppliers, developers, customers from a new system and also what a software product must perform in order to satisfy the requirements. To collect exact and true requirements of

stakeholders play an important role in the sense that incomplete and changing requirements can cause failure of intended software product. This situation occurs when developer team keeps poor interactions with stakeholders presuming that they have understanding of the requirements of stakeholders. Therefore, it can be mentioned that a vast number of software products fall short due to bad requirements engineering procedure.

2. REQUIREMENTS ENGINEERING

According to Zave [1], "Requirements engineering is the branch of software engineering concerned with the real world goals for, functions of, and constraints on software systems [2]. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families [3]." The definition of requirement engineering by Zave shows real world goals that inspire the growth of the software product. This definition also point out about precise specifications that become the platform for analyzing requirements, validating about the requirements of stakeholders, defining what developers have to construct and verifying correct product delivery. Finally, the definition provides specification about evolutionary software families that emphasis on reality of changing world and reuse of partially specifications. In other words, requirement engineering offer suitable machinery for understanding what the customer wants, analyzing needs, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements [4, 5]. Because of bad client developer interaction, developer team could face incomplete requirements issues during software development cycles. It may be seen that client could change their requirements during software life cycle. Because of this, the developer team faces a lot of problem. That affects development process of software product. Therefore, a great number of software development life cycles fail because of bad requirements engineering.

According to work done by Lindquist, poor requirements management can be credited to 71 percent of software projects failure. Misunderstanding of requirements can cause missed deadlines, and change management issues. That can increase development cost of product. Another study found that failed or abandoned systems cost \$100 Billion in the USA [6]. According to the [7], The Standish Group's CHAOS survey of over 350 organizations and 8000 projects produced the following results [4, 8, 9]:

1. 16% delivered within budget and schedule
2. 31% cancelled before completion
3. 53% overrun in budget or schedule

4. 89% average budget overrun
5. 122% average schedule overrun
6. 61% average of originally specified content

Therefore, it can be concluded that for software development process, a sequence of steps must be performed in order to complete a set of tasks. Requirement engineering plays an important role for the success of software product and building a high-quality software product. It can come into the picture only by performing step-by-step procedure for requirements engineering. Next section is dedicated to describe types of requirements that can be involved in requirement engineering framework.

3. TYPES OF REQUIREMENTS

According to IEEE standard, Software requirements are (1) a condition or clause required by a customer to solve a problem or attain an objective; (2) a condition or capability that must be met [10] or possessed by a system or system components to satisfy a contract, standard, specification, or other formally imposed document [11]; (3) a documented representation of a condition or capability as in 1 or 2 [12].

According to client's need and developer's wish, Software requirements can be categorized into various clusters for example business requirements, user requirements, functional requirements, non functional requirements, performance requirements and security requirements [13]. The descriptions of requirements are as follows:-

3.1 Business Requirements

Business requirements are normally stated in Business Requirements Document (BRD). Business requirements state about why the organization is undertaking the project. BRD tells about the benefits of undertaking the project with respect to organization and clients. BRD also contains problem statement, project vision, project constraints (budget, schedule and resource) [14], business objectives, project scope statements, business process analysis, stakeholder analysis, IT service impact analysis [14].

3.2 User Requirements

User requirements are stated in User Requirements Document (URD). User requirements describe about what users do with the system. User requirements are considered by development team, only when stakeholders and development team sign off the contract. Not properly defining of user requirements results in failure of software products [15].

3.3 System Requirements

Systems requirements are also called as solution requirements. Solution (System) Requirements describes about what the developers use to construct the system and what the system "shall do." System requirements are categorized as either functional or non-functional requirements [11].

3.4 Functional Requirements

A functional requirement specifies something that the developer needs to build to deliver the solution and also describe how product works and also include input that the software gets and output it generates. For example, a system may be required to enter and print cost estimates [16]; this is a functional requirement. Any uncertainty in functional requirements can affect on product's functionality [16].

3.5 Non-Functional Requirements

Non-functional requirements are described those requirements that are not covered by functional requirements. Non-functional requirements are also called as supplemental requirements, quality of service requirements [17], or service level requirements. Non-functional requirements include availability, business continuity, portability, reliability, testability, efficiency and modifiability, compliance, interoperability, maintainability, performance, usability.

All the above mentioned requirements that can be identified through requirements engineering process are shown in Figure 1.

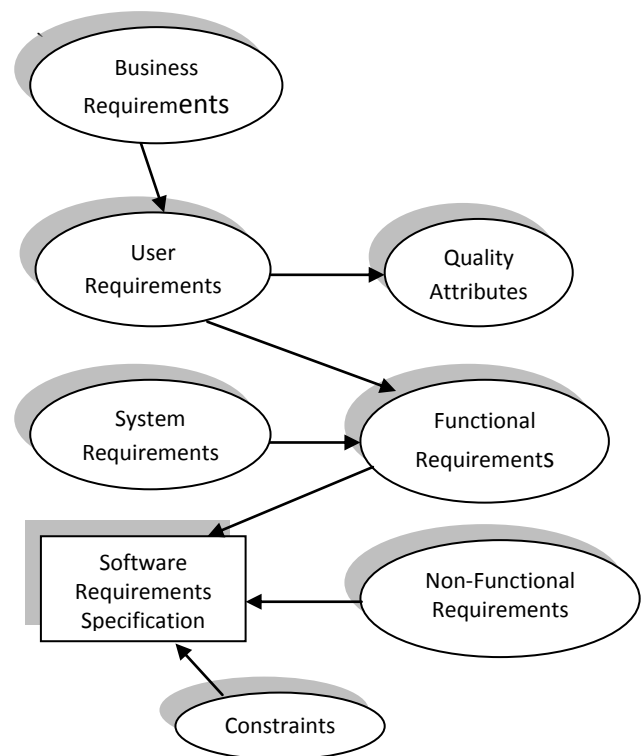


Figure 1. Types of Requirements and Their Relationship

4. REQUIREMENT ENGINEERING FRAMEWORK

It is already described that what is the need of requirement engineering to describe requirements of stakeholders? There are a number of frameworks available to describe requirement engineering phases in literature. It is seen that requirements terminology is not clearly delineated and deferred from one author to another. In this paper, researcher is trying to avoid confusion and describing a framework in which requirement engineering is divided into a set of sub processes. The requirement engineering framework is shown in Figure 2:-

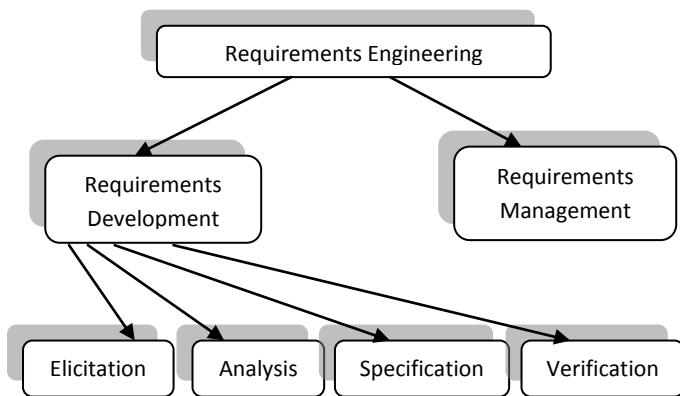


Figure 2. A Hierarchical Decomposition of Requirements Engineering

To understand problem effectively, all the phases of Requirements Engineering are described thoroughly and independently. In reality, all the phases are iterative, interleaved, and can take the whole duration of the entire software systems development life cycle:-

4.1 Requirements Inception or Requirements Elicitation

Requirements elicitation is the first step of Requirement Engineering. In this step, developer team tries to identify new requirements and collect all the requirements of a software product from users, clients and other stakeholders.

4.2 Requirements Analysis and Negotiation

In this step, developer tries to check requirements and resolve stakeholders' conflicts. Requirements analysis plays an important role in the success of a systems or software project. Requirements analysis process can involve many delicate psychological skills and can take long and tiring procedure for thorough analysis of requirement analysis. With the help of requirements analysis, developer team refines the user's needs and constraints.

4.3 Requirements Specification

Software Requirement Specification (SRS) shows an agreement between client and suppliers. All the customer's requirements are documented in Software Requirements Specification. Software requirements specification which includes complete documentation of what the software does externally (without regard to how it works internally) [18]. SRS has the information about software system to be developed and also contain functional and non-functional requirements, a set of use cases that describe user interactions with the software.

4.4 System Modeling

With the help of this phase, abstract descriptions that are amenable to interpretation can be constructed. There are various models available that can be used as elicitation tools. These tools can also be used for further information gathering. Example is Unified Modeling Language (UML).

4.5 Requirements Validation

In that phase, the documented requirements are checked to see that models are consistent and meet stakeholder needs. It is an ongoing process which ensures that right problem is being tackled at any time. This phase examines the specification to

ensure that all requirements are stated unambiguously. If any time errors are presented, these errors can be corrected and omitted according to situation [19].

4.6 Requirements Management

In that phase, changed requirements are managed [20]. During the phase, developer team performs a set of steps to identify, control and track requirements. Requirements management starts with the analysis and elicitation of the objectives and constraints of the organization. Requirements management includes supporting planning for requirements, integrating requirements as well as relationships with other information delivering against requirements, and changes for these [20].

5. CONCLUSION

As per above discussion, it is clear that requirements engineering play an important role for the success of software product. To achieve the goal of requirements engineering, there is a need of applying the full proof procedure to identify the exact need of the client. This paper enables the users to understand the purpose of requirements engineering. To know the goals of project, procedure should clearly elaborate the types of requirements that can be taken for developing software product. This paper helps users to understand about the types of requirements according to condition of software product. The raw requirements can be going through the predefined set of steps for refinement and consistent of requirements. The paper shows the path for raw requirements. After following the path, raw requirements become unambiguous and consistent. As the success and failure of most software product is depend on requirements of the customers, therefore the requirements engineering process should be performed intelligently.

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of requirements engineering.

7. REFERENCES

- [1] Zave, P. (1997). Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 29(4): 315-321.
- [2] Phillip A. Laplante . 2014. *Requirements Engineering for Software and Systems*, Second Edition. CRC press, Taylor & Francis Group. An Auerbach Book. ISBN978-1-4665-6082-6
- [3] Philip A. Laplante. 2007. *What Every Engineer Should Know about Software Engineering*. Press, Taylor & Francis Group. An Auerbach Book. ISBN978-0-8493-7228-5.
- [4] Billy Hendrawan. 1601215765. 34PFT. Chapter 5. <http://billyhendrawan.blog.binusian.org/files/2014/03/Chapter-5.docx>.
- [5] Roger S. Pressman. 2005. *Software Engineering: A Practitioner's Approach*. Sixth edition, tata Mc Graw Hill Higher Education, 2005, ISBN 0-07-285318-2.
- [6] Davey, B. & Cope, C., 2008. Requirements Elicitation – What's missing? *Issues in Informing Science and Information Technology*, 5(1), p.53-57.
- [7] B. W. Boehm, R. Valerdi 2005. *Achievements and Challenges in Software Resource Estimation*. USC CSSE Technical Report. No.USC-CSE-2005-513.



- <http://sunset.usc.edu/csse/TECHRPTS/2005/usccse2005-513/usccse2005-513.pdf>.
- [8] Del Gobbo, D., Napolitano, M., Callahan, J. & Cukic, B. 1998. Experience in Developing System Requirements Specification for a Sensor Failure Detection and Identification Scheme. 3rd High- Assurance Systems Engineering Symposium, Washington, DC, USA, 13-14 November 1998.
- [9] Standish, G.. 1995. CHAOS.
- [10] A Framework for Software Product Line Practice, Version 5.0. http://www.sei.cmu.edu/productlines/frame_report/req_en.htm.
- [11] Kathy Schwalbe. 2014. Information Technology Project Management. ISBN978-1-285-84709-2.
- [12] Brusa, G., Caliusco, M.L., Chiotti, O.. 2008. towards ontological engineering: a process for building a domain ontology from scratch in public administration. Expert Systems: The Journal of Knowledge Engineering. 25(5). 484-503.
- [13] Michael Middleton, Len Asprey. 2011. Integrative Document and Content Management: Strategies for Exploiting Enterprise Knowledge: Systems for Exploiting Enterprise Knowledge. Idea Group Publishing, USA. ISBN1-59140-068-6.
- [14] Business Requirements. Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Business_requirements.
- [15] John Parker. 2012. Business, User and System Requirements http://requirements101.com/Powering_Requirements_Success/bid/123830/Business-User-and-System-Requirements.
- [16] John Parker. 2012. Business Requirements vs. Functional Requirements. <http://enfocussolutions.com/business-requirements-vs-functional-requirements/>.
- [17] Kris Slaboszewski. 2013. Requirement Type Defined. <http://community.rightpoint.com/blogs/viewpoint/archive/2013/08/08/requirement-types-defined.aspx>.
- [18] P. Loucopoulos and V. Karakostas. Chapter 2. System Requirements Engineering. McGraw-Hill. http://www.utdallas.edu/~chung/SYSM6309/RE_chapters/Chapter%202.pdf.
- [19] Software Verification and Validation. Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Software_verification_and_validation.
- [20] Requirements Management. Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Requirements_management