

# Requirements Elicitation and Access: A Research

Gopichand Merugu

**Abstract:-** Requirements Elicitation and Analysis is one of the most important activity in software development life cycle. Requirements Elicitation and Analysis is about gathering and analyzing the requirements as per customer needs. If this analysis is not done properly, it may cause for problems in further phases that in turn may lead to failure of the software. Hence the requirements Elicitation and Analysis has to be given at most priority in SDLC. This paper presents a survey on requirements elicitation and analysis, particularly on functional Requirements.

**Keywords :** Requirements, Elicitation , Analysis.

## I. INTRODUCTION

Requirements for a software system are defined as the descriptions of the services that are to be provided by the system and its operational constraints. Generally, requirements reflect the customer needs. Requirements engineering is defined as, the process of identifying, analyzing, specifying, and documenting the services and their constraints [1]. Creating and maintaining system requirements document is the main goal of the requirements engineering process. The requirements engineering process includes four activities. These are like

- Feasibility Study
- Requirements Elicitation and Analysis
- Requirements Specification
- Requirements Validation

### Feasibility Study:

In requirements engineering process, feasibility study is the first activity that happens in developing a new system. It is the study of possibilities to complete a project successfully taking into account, technological, legal, scheduling economic and other factors [2]. The inputs for the feasibility study are the purpose of the system to support the business process, preliminary requirements set and an outline of system description. Based on the result of the feasibility report one can say that whether the project is worth investing in or not.

### Requirements Elicitation and Analysis:

Elicitation is the process of seeking, uncovering, acquiring, and elaborating requirements for software systems [3]. Requirement analysis is about modeling and analyzing the requirements which are captured in requirements elicitation process [4].

In an organization different people take part in requirements elicitation and analysis of a software system.

These people are referred as stakeholders. The main goal of requirements elicitation and analysis is to answer the question “have we got the right requirement”? [5]. The research work carried out about requirements elicitation and analysis can be found in [6,7,8,9,10,11]. This process includes different activities that are mentioned below.

### Requirements discovery:

It is the process used by requirements analysts to identify or extract problems of software systems and solution requirements from the different users [12]. In this process requirements analyst may interact with different stakeholders to collect the requirements because each stakeholder may have different views about the system.

### Requirements classification and organization:

This is the process of organizing the requirements into different groups that are gathered from different stakeholders. Requirements can be classified based on number of dimensions [1, 13, 14].

**Requirements prioritization and negotiation:** As requirements gathering involves multiple stakeholders hence requirement conflicts may occur during requirements elicitation and analysis. In this process priorities are assigned to the requirements and the conflicts that are found. These requirements conflicts can be resolved by negotiation with the stakeholders [15, 16, 17].

**Requirements documentation:** This is an essential and integral part of software requirements process. The requirements gathered are documented and will be given as an input to the further phases of the development cycle. The requirements document obtained is either formal document or informal document [18].

**Requirements Specification:** A software systems' requirements specification specifies all the necessary requirements for software system. This task explains the software system behavior which is supposed to be developed. It contains use cases, which describes the interactions among the users and the software system. A Software Requirements Specification (SRS) is a document that describes all external observable behaviors and characteristics expected of a software system. Along with functional requirements the SRS should include nonfunctional requirements. This SRS refers to the constraints imposed on the software system at all the stages such as design, implementation and testing, like design constraints, quality standards or performance engineering

**Revised Manuscript Received on September 10, 2019.**

Gopichand.Merugu, Professor & Head, Department of Information Technology, Vardhaman College of Engineering, Hyderabad, Telangana, India.

(E-mail Gopi\_merugu@yahoo.com)

requirements. This SRS document provides necessary information about all the requirements which are needed for the development of a software system [19].

### *Requirements Validation:*

Requirements validation is the process of confirming whether the requirements gathered actually defines the software system according to the user needs or not [20]. This process plays an important role because if there are any irrelevant or improper requirements specified in the SRS document, they may lead to the wastage of time and effort in making extensive rework during the development of a system or after the system is implemented, when they are identified [21, 22].

This is the brief information about requirements engineering and requirements engineering process activities, which explains the importance of the requirements engineering process. There are two types of requirements one is functional requirements and other one is nonfunctional requirements.

## II. FUNCTIONAL REQUIREMENTS ELICITATION AND ANALYSIS RESULTS

The requirements can be classified into two types. One is functional requirements and other one is nonfunctional requirements [1]. Functional requirement is defined as a requirement that specifies a function that a system or system component performs [23]. The functional requirements describe what the system supposed to do and what the system should not do explicitly in some cases. The requirements depend on the type of software being developed, users who are expected to use the software and the organizational general approach taken while generating the requirements. The functional requirements are used to explain the functionality of the software system in detail such as its exceptions, inputs and outputs.

In the literature several methods have been proposed with various sophisticated tools, which support the process of requirements analysis. The widely used methods are Goal Based Analysis (GBA) [24], Coupling Goal with Scenario [25], Use Case driven Analysis (UCA) [26], Scenario Based Analysis (SBA) [27], Requirements Elicitation and Analysis using Goal [28] and Multi view approach using goal and scenario [29].

In use case driven analysis, the use case represents the functional requirement which is the basic unit. This approach provides the useful relationships such as dependency, association and generalization, which are important for software developers in designing and implementing the software systems. With this use case driven analysis the software developers can easily understand the structure of the system as well as its functionalities. The limitation of this approach is that these notations emphasize the requirements in the developer orientation rather than the user orientation.

Moreover, the use case analysis cannot support the process of the requirements identification. In the scenario based analysis, the scenarios are used to describe the existing system environment by providing enough context information. Based on this information the requirements can

be identified and validated, and also describes agents behavior. The actual experiences with a system captured from users are the instances of scenarios. Scenarios familiar to users are analyzed as the requirements. The scenario based models are created in order to explain the software system environment, which can identify the dependencies and connections between the system and its environment. Scenario modeling may appear to be as same as the system modeling. In scenario based models it is found that the environment model or the system content depends on where the intended system boundary exists. During requirements investigation, boundaries of the system will change frequently which causes the single model of the intended system environment upon which a boundary will be imposed. Further, the alternative technical specifications may be described within the model. The interactions among the agents in some particular context will generate a scenario. SBA does not provide any mechanism to obtain the relationships between the scenarios, therefore entire software system requirements can't be identified at a higher level [30]. The requirements which are high-level abstractions of objectives of the system are represented and analyzed as goals in the GBA and in the coupling approach. Extraction of goals can be from various methods of gathering information which includes process descriptions such as entity relationship diagrams or flow charts or unified modeling language diagrams. From the process descriptions, goals can be identified by searching statements which seems to guide design decisions at different levels within an organization of a system. The whole system requirements represent the goal which is very abstract. By using "OR", "AND" and "Conflict" relations the relationships between goals are expressed. During the refinement of goals these relations are identified typically.

In [28] an approach is proposed to elicit the requirements of software system. This approach includes requirements elicitation rule and requirements analysis process. It mainly considers stakeholder view, using this approach, all possible requirements can be gathered from all stakeholders. Using this approach, requirements can be elicited and analyzed in short time.

In [29] a multi view approach is proposed. It has four different views, like, abstraction view, structure view, function view and quality view. All these views are useful to improve the elicitation and analysis process. The structure view shows the structure of entities in the system domain. Requirements elicitation is supported by functional view through scenarios particularly based on the interaction between system and agents. Relationships between different scenarios is represented as goals and sub goals in abstract view. Quality view represents the mapping of functional requirements into quality attributes.

This helps the developers to identify the important scenarios in the system.

### III. CONCLUSION

Requirements elicitation and analysis is considered as most important activity in the development of software. There are different approaches proposed in the literature for this purpose. This paper provides the description about some of the approaches like Requirements Elicitation and Analysis using Goal, goal-based analysis, use case-driven analysis, Multi view approach using goal and scenario and scenario-based analysis. All these approaches are very useful in eliciting and analyzing the requirements in software development process.

### IV. REFERENCES

1. Ian Sommerville, "Software Engineering", seventh ed., Addison-Wesley, 2005.
2. Robert Pergl, "Feasibility Study Inputs based on Requirements Engineering", EOMAS10 Proceedings of the 6th International Workshop on Enterprise & Organizational Modeling and Simulation, pp. 121-132.
3. [Didar Zowghi](#) and [Chad Coulin](#), "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools", Engineering and Managing Software Requirements, 2005, pp. 19-46.
4. Umar Sajjad Muhammad and Qaisar Hanif, "Issues and Challenges of Requirement Elicitation in Large Web Projects", Master Thesis Computer Science Thesis no: MCS-2010:05, School of Computing, Blekinge Institute of Technology Ronneby Sweden, January 2010.
5. Maciaszek, L. A, "Requirements analysis and system design", 2005, Harlow, Pearson/Addison Wesley.
6. Hooper, W. James and Pie Hsia, "Scenario-Based Prototyping for Requirements Identification", ACM SIGSOFT Software Engineering Notes, vol. 7, Issue.5, 1982, pp.88-92.
7. H. Holbrook, "A scenario-based methodology for conducting requirements elicitation", ACM SIGSOFT Software Engineering Notes, Vol. 15 Issue 1, Jan 1990, pp. 95-104.
8. Seo-Young Noh, Shashi K. Gadia, "RAST: Requirement Analysis Support Tool based on Linguistic Information", 2003.
9. Yuqin Lee and Wenyun Zhao, "An Ontology-based Approach for Domain Requirements Elicitation and Analysis", Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006, pp. 364-371.
10. Gopichand.Merugu, A.AnandaRao,"Four Layered Approach to Non-Functional Requirements Analysis" International Journal of Computer Science and Issues (IJCSI), ISSN: 1694-0814, Volume 8, Issue 6, pp 371-379, November 2011.
11. Gopichand.Merugu, A.AnandaRao" Five Layered model for identification of software performance requirements" International Journal of Software Engineering and Applications(IJSEA), ISSN: 0975-9018, Volume 3,No.5, pp 47-61,September 2012
12. [Jeffrey L. Whitten](#), [Lonnie D. Bentley](#), [Kevin C. Dittman](#), "System Analysis and Design Methods", 6<sup>th</sup> edition, McGraw-Hill Irwin, 2004.
13. A.M. Davis, "Software Requirements: Objects, Functions and States", Prentice Hall, 1993.
14. G. Kotonya and I. Sommerville, "Requirements Engineering: Processes and Techniques", John Wiley & Sons, 2000.
15. S. Easterbrook, "Handling conflict between domain descriptions with computer-supported negotiation," Knowledge Acquisition, 1991, vol. 3, pp. 255-289.

16. B. Boehm and A. Egyed, "Software Requirements Negotiation: Some Lessons Learned", presented at 20th International Conference on Software Engineering, Kyoto, Japan, 1998.
17. Sabrina Ahmad, "Negotiation in the Requirements Elicitation and Analysis Process", 19<sup>th</sup> IEEE Australian Conference on Software Engineering, 2008, pp.683-689.
18. Norah Power, Tony Moynihan, "A Theory of Requirements Documentation Situated in Practice", [SIGDOC '03](#) Proceedings of the 21st annual international conference on Documentation, pp.86-92.
19. S. Robertson and J. Robertson, "Mastering the Requirements Process", Addison-Wesley, 1999.
20. A.Terry Bahill and Steven J. Henderson, "Requirements Development, Verification and Validation Exhibited In Famous Failures", Systems Engineering, Vol. 8, No. 1, 2005,pp. 1-14.
21. Barry W. Boehm, "Guidelines for Verifying and Validating Software Requirements and Design Specifications", IEEE Software, Vol. 1, Issue. 1, 1984, pp.75-88.
22. Juan F.Sequeda, "A Taxonomy of Verification and Validation of Software Requirements Specifications", Univeristy of Texas, USA 2008.
23. IEEE Std 610.12-1990, IEEE standard Glossary of Software Engineering Terminology.
24. Anton, A.I "Goal-based requirements analysis", Proceedings of the 2nd International Conference on Requirements Engineering, Colorado Springs, CO,pp. 136-44.
25. Rolland, C., Souveyet, C. and Achour, C.B. (1998), "Guiding goal modeling using scenarios", IEEE Transactions on Software Engineering, Vol. 24 No. 12, pp. 1055-71.
26. Bittner, H. and Spence, I. (2002), "Use case Modeling", Addison-Wesley, Reading, MA.
27. Sutcliffe, A.G. "Scenario-based requirement analysis", Requirements Engineering Journal, Vol. 3 No. 1, pp. 48-65.
28. M.Gopichand, A.AnandaRao, K.Narender Reddy and J.Kiran Kumar, "An approach to requirements elicitation and analysis using goal" ICSTE 2010: 2<sup>nd</sup>IEEE international conference on Software Technology and Engineering, October 3-5,2010 San Juan, Puerto Rico,pp.218-221.
29. Jeongwook Kim, Jintae Kim, Sooyong Park and Vijayan Sugumaran " A Multi-view Approach for Requirements Analysis using Goal and Scenario" Industrial Management and Data Systems, Volume 104, Number 9, 2004, pp. 702-711.
30. Weidenhaupt, K., Pohl, K., Jarke, M. and Haumer, P. "Scenario usage in system development: a report on current practice", IEEE Software, Vol. 15 No. 2, pp. 34-45.

### V. AUTHORS PROFILE



GopiChand.Merugu received Ph.D. in Computer Science &Engineering from JNTUA, Anantapur, India and he received his M.Tech. in Software Engineering from the same university. He received B.Tech. degree in Information Science & Technology from Nagarjuna University, India. He is Professor & Head, department of Information Technology Vardhaman College of Engineering, Hyderabad. He is a member of IEEE, ACM, IAENG, CSI and ISTE.

