

AN EFFECTIVE QUALITY REQUIREMENT FRAMEWORK FOR REQUIREMENTS ACCURACY AND SOFTWARE DEVELOPMENT

Usman A. Khan, ukhan@kau.edu.sa & Ibrahim A. Albidewi, ialbidewi@kau.edu.sa
Faculty of Computing & Information Technology, King Abdulaziz University

Abstract

Requirement engineering is critical for software development process. It is the key factor to satisfy the stakeholders' need. An inadequate, inconsistent or ambiguous requirements may have a critical impact on the resulting software and may lead to failure. Requirements need to be organized reviewed continuously. This paper introduces quality requirements framework for software development process. This framework involving repeated analysis and evaluation of requirements until satisfactory requirements are identified. A concept of requirement fusion is introduced in the framework. Implementing this framework will enhance the production of quality software product.

Keywords: Requirement engineering, requirement Abstraction, requirement analysis, requirement fusion, requirement elicitation, requirement development.

Introduction

Requirement Engineering is a 30-year-old term, designed to describe the actual process of answering the most important question of any software project. Requirements are the conditions and competence that the stakeholder need in his desired system [1]. Requirements are classified as functional and nonfunctional. A functional requirement describes what a software system should do, while non-functional requirements place constraints on how the system will do so.

Requirements engineering process is vital process to determine the resulting product. It implements all collected requirements into software development processes. Requirements engineering process is the start of any system development lifecycle [2]. Requirements engineering is a progressive and iterative process, it takes place alongside with other development activities. Requirements engineering contains a set of activities for discovering, analysing, documenting, validating and maintaining a set of requirements for a system [3-7].

Requirement Engineering Phases

Software requirements express the needs and constraints that are placed upon a software product that contribute to the satisfaction of some real world application, alternately, the properties that must be exhibited in order to solve some real world problem. Requirements engineering is an important aspect of any software project, and is a general term used to encompass all the following phases related to requirements.

A. Requirement Abstraction and Elicitation

Abstraction permits the arrangement of requirements on diverse levels and helps the mechanism to break down the requirements to make them equivalent to one another. The most vital activity in Requirement Engineering process is the elicitation of requirements process. Requirement Elicitation involves sifting through large amounts of information and deciding what exactly is relevant. Information gathered from Requirement gathering phase often has to be translated, studied, designed and tested before conducting further gathering of system requirement. A standout amongst the most critical objectives of elicitation is to figure out what issue requires to be unraveled and thus distinguish system boundaries. Goals denote the objectives the system must have. Eliciting centers the requirement engineer on the issue scope and needs of the stakeholders, instead of feasible solution for those issues. A number of elicitation techniques are available to the requirement engineer. The selection of which technique is more suitable is based on the evaluation of its strengths and weakness in this specific domain.

B. Requirement Analysis

Requirement Analysis is a procedure of investigating requirements to catch and resolve clashes between requirements, uncover the limits of the system and how it should cooperate with its environment, expand system requirements

to software requirements [8,9]. Analysis brings about the details of software's operational qualities; shows software's interface with other framework components and makes restrictions that software must meet. Requirements analysis gives the software designer a representation of data, capacity and conduct that could be interpreted to architectural, interface and component-level designs. In few software projects, certain things reoccur across all projects within a specific application domain. These are called as analysis patterns and represent something within the application domain that can be reused in many applications. These analysis patterns speed up the analysis process by suggesting design patterns and reliable solution to common problems [10].

C. Requirement Specification

A specification can be a written document, a set of graphical models, a formal mathematical model or a collection of these. It is sometimes necessary to remain flexible when a specification is to be developed. A precise specification may be an informal document which is a work product produced by a requirement engineer. It will serve as a foundation for further activities like evaluation with stakeholders etc. An informal requirement specification will informally document the function, scope, requirements etc. of a computer based system and the constraints that will govern its development.

D. Requirement Validation

In this phase the product which is produced based on stakeholder's requirements is inspected to make sure that all the specified requirements are achieved. Also in this phase errors should be detected and corrected. Explicitly outlining the requirements is a basic prerequisite for validating requirements and settling clashes between stakeholders [11]. A key troublesome in requirements validation focuses on the issue of difference between stakeholders.

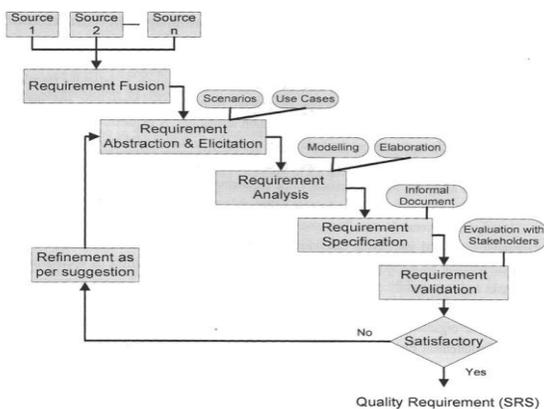
Proposed Framework to achieve quality requirement

The proposed framework includes all the stages of Requirement Engineering arranged in an iterative manner to achieve an effective and efficient product.

Above defined framework, i.e Fig.1, all the phases of Requirement Engineering process are arranged in a sequence inhibiting an iterative behavior. This framework will produce those requirements which are actually necessary and intended by the stakeholders. By applying an iterative behavior to these phases, a repeated execution can improve the quality of requirements by subsequent evaluation by stakeholders. This process first gathers requirements from various sources and collects them at one place for further engineering using the process of fusion. Many different approaches have been proposed for requirement gathering. For example, meetings are conducted attended by both software engineers and stakeholders; an agenda is suggested to cover all the important points, rules for preparation and participation are established etc.

Requirement Fusion:

Requirement fusion is a process of collecting the needs to solve a problem or issues and achieve an objective. It is not quite accurate to say that requirements are in the minds of the clients; it would be more accurate to say that they are in a social system of client organization. They need to be invented and that invention has to be a cooperative venture involving the clients, the users and the developers. All the requirements gathered from various sources are fused together at one place. Important aspect of fusion is that it will collect the related and similar requirements and store them in groups. This phase will enable the next phase i.e elicitation, to select the intended requirements from the predefined groups of related requirements. Necessary and useful requirements are then elicited using various elicitation techniques. A requirement engineer can have recourse to eliciting information about the undertakings, clients presently perform and those they may need to perform which might be regularly spoken to being used cases, used to depict ostensibly obvious requirement of a system. All the more particularly, the requirement engineer might select a specific way through an use case, a situation, with a specific end goal to better comprehend a few parts of utilizing a system. The identified requirements are then analyzed. Elicitation will help in identifying goals, determining scope and identifying functional and non functional requirements. Analysis is the process of understanding the relations between organization body; the condition or constraints of the organization operations. Modeling is often used to capture the purpose of a



system, by describing the behavior of the organization in which that system will operate. Elaboration is an analysis modeling action. Elaboration is driven by the creation and refinement of user scenarios that describe how the end user will interact with the system [12,13]. To sum up, analysis will help to model requirements and data, determine feasibility and to refine requirements. An informal specification is created that specifies all the necessary requirements for the development process and the final product. It will document stakeholder as well as developer's requirements and give a specification of the interaction among all. The created description/specification is evaluated by the stakeholders and users to identify any lack of requirements or any ambiguous requirements being included in the specification. The problems of inconsistency, completeness and unambiguity are addresses so that these can be again analyzed and an appropriate set of requirements can be created. These changes defined by the stakeholders are communicated and the defined requirements are again reviewed and refined. Validation will also include the usability check and will provide some test cases to simulate requirements. This refinement of requirement will include the analysis phase where the requirements will again be analyzed according to the changes made by the stakeholders. This iterative process goes on until the requirements gathered are satisfactory and in accordance to the specification of various stakeholders. If the requirements identified are satisfactory, then the final product of Requirement Engineering i.e. formal specification of requirements (SRS) can be submitted to the system for use in further phases. This specification is the final product achieved after executing various phases of Requirement Engineering iteratively. This framework provides a product in the form of a formal document of requirements which will ensure the requirement veracity and will be an input to the next phase of the process life cycle. Since Requirement Engineering is an important phase in software development life cycle, this framework will be applicable to all process models of Software Engineering.

Conclusion

It is necessary to understand requirements before design and construction of a computer-based system can begin. To accomplish this, a set of requirements engineering tasks are conducted. The repeated evaluation of requirements this framework results in the achievement of those requirements which are actually intended by the stakeholders. The newly introduced phase of requirement fusion in the proposed framework will further support the requirement engineering process to achieve quality requirements for a high quality product i.e. quality requirements, ensures the accuracy of the requirements and leads to the development of high quality software.

References

- [1] P. Jalote, "An Integrated Approach to Software Engineering", 3rd edition, Narosa Publishing house, India, 2005.
- [2] G. Alan. "Software quality: theory and management." Lulu. com, 2011.
- [3] R. H. Thayer & W. W. Royce, "Software Systems Engineering, IEEE System and Software Requirements Engineering". IEEE Software Computer Society Press Tutorial. IEEE Software Society Press. Los Alamos, California, 1990.
- [4] P. Clarke,&R. V. O'Connor."The situational factors that affect the software development process: Towards a comprehensive reference framework." Information and Software Technology, Vol. 54, No.5, pp.433-447, 2012.
- [5] J. Siddiqi, "Requirement Engineering: The Emerging Wisdom", IEEE Software, 1996, pp.15- 19.
- [6] Sommerville & P. Sawyer, "Requirements Engineering: A Good Practise Guide". John Wiley & Sons, 1997.
- [7] Villegas, M. Norha, A. Müller, G. Tamura, Laurence Duchien, and Rubby Casallas. "A framework for evaluating quality-driven self-adaptive software systems." In Proceedings of the 6th international symposium on Software engineering for adaptive and self-managing systems, pp. 80-89. ACM, 2011.
- [8] R. Stevens, P. Brook, K. Jackson & S. Arnold, "Systems Engineering - Coping with Complexity", Prentice Hall, London, 1998.
- [9] Edward Colbert, "Requirements Analysis with the Object—Oriented Software Development Method", Absolute Software Co., Inc., 1992.
- [10] Geyer-Schulz, A. and M. Hahsler, "Software Engineering with Analysis Patterns", Technical report 01/2001, November 2001.
- [11] G. Kotonya & I. Sommerville, "Requirements Engineering: Process and Techniques". John Wiley & Sons, 1998.
- [12] J. D. Sailor, "System Engineering: An Introduction. IEEE System and Software Requirements Engineering", IEEE Software Computer Society Press Tutorial. IEEE Software Society Press,1990.
- [13] W. W. Royce, "Managing the Development of Large Software Systems" Proceedings of IEEE Wescon, Reprinted in Proceedings 9th International Conference Software Engineering IEEE Computer Society Press, Los Alamitos. California. USA, 1987, pp. 328-338.