



REQUIREMENT ENGINEERING: HOW TO MAKE IT COMPLETE AND CORRECT

Dr. Ajay D. Shinde

Associate Professor, Department of Computer Studies,
Chhatrapati Shahu Institute of Business Education and Research, Kolhapur, Maharashtra.
ajshinde@siberindia.edu.in

Manuscript History

Number: IRJCS/RS/Vol.04/Issue12/DCCS10082

DOI: 10.26562/IRJCS.2017.NVCS10082

Received: 11, November 2017

Final Correction: 25, November 2017

Final Accepted: 05, December 2017

Published: December 2017

Citation: Shinde, D. A. D. (2017). REQUIREMENT ENGINEERING: HOW TO MAKE IT COMPLETE AND CORRECT. *International Research Journal of Computer Science*, Volume IV, 12-15. doi:10.26562/IRJCS.2017.DCCS10082

Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India

Copyright: ©2017 This is an open access article distributed under the terms of the Creative Commons Attribution License, Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Abstract: The software development process heavily relies on requirement engineering as it forms the base for entire process. Although software engineering is full of methods for requirement analysis, the problem we face is which method to select and how to apply it. It is expected that we should be able to get clear and complete idea about what is expected by the user from the proposed system. This puts emphasis on requirement analysis process. The method we need to adopt should enable us to get clear and complete set of requirements. The requirement engineering process dependent on abilities of the persons carrying out the process also the nature of system puts certain constraints on the process. . This paper is an attempt to look at certain problems posed by the requirement engineering process and possible corrective measures against it to help improve overall software quality.

Keywords: Software Engineering; Requirement Analysis; Software process; user requirements; requirements engineering;

I. INTRODUCTION

The software development process starts with requirements engineering process, here we must collect user requirements, understand it, and specify in appropriate manner. Requirements engineering is important task, because it is observed over many decades that many software's failed due to inconsistent, incomplete or simply incorrect requirements identification and specifications [2]. The requirement engineering is systematic way for understanding user requirements this can be achieved using iterative and co-operative process, once requirements are gathered they can be specified using variety of format, by checking the accuracy of the understanding gained [1]. At the time of application of requirement engineering process we face numerous problems, but as this process forms base for entire software development we must be aware about the problems and how to resolve them or at least minimize the effect of problem, so that we get clear and complete idea about user requirements. In this paper we will discuss problems with requirement engineering and possible corrective measures.

II. REQUIREMENT ENGINEERING BASICS

In this section we will take a look at some of the basic concepts in requirement engineering. According to IEEE standards requirement engineering is defined as follows [IEEE- 610.12] [IEEE-830] [IEEE-729]

1. A condition or capability needed by a user to solve a problem or achieve an objective.[4]

2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.[4]
3. A documented representation of a condition or capability as in 1 or 2. [4]

There are many stakeholders in requirement engineering process. According to IEEE standards the stakeholders in requirements engineering are [IEEE-610.12].

Customer: A person or group of persons that order the product and sometimes may decide the requirements.

Supplier: The person, or group of persons that are involved in actual development of the product.

User: The person, or group of persons that operate and use the product.

The technique used to collect requirement of customer and/or user is called as Elicitation. Many elicitation techniques are currently in practice in the IT industry. Choosing correct elicitation techniques based on software type and experience is the first and foremost thing. This will help in collecting the correct and complete requirements and it will also make the development process faster [3]. Once the requirements are collected through elicitation technique the next task is specification. This will result in description of the product to be delivered, but will not consider how it will be developed. Requirement engineering is a difficult process, why is it so is answered in [5] by Zielczynski P, Also What the customer wants is seemed to be easy question but It is very difficult to answer [1].

III. TYPES OF REQUIREMENTS

In his paper [6] Michel Srivasthan has given five types of requirements, but ultimately these types may be mapped to only two types of requirements. The most common types of software requirements are [6]:

A. Business Requirements (BR)

These are high-level business goals set by organization or the customer. Every business requirement can be mapped to functional requirement.

B. Market Requirements (MR)

These still are high-level business goals, they also outline market needs. Every market requirement can map to either functional or non-functional requirement.

C. Functional Requirements (FR) – Use Cases

These cover the functionality of the product in detail.

D. Non-Functional Requirements (NFR)

These are related to Reliability, Scalability, Security, Integration, usability etc.

E. UI Requirements (UIR)

User interface specs are not considered as “requirements” in traditional requirements management theory and are mapped to non-functional requirement such as usability.

IV. PROBLEMS WITH REQUIREMENT ENGINEERING

In [7], Donald Firesmith has presented 12 problems their negative consequence and solutions. The problems presented in [7] can be resolved by simple considerations and precautions. Following are the precautions and considerations.

A. Requirements Quality

It is expected that the requirement should be unambiguous, complete, consistent, correct, should not be specified in technical jargon, feasible to implement, relevant to system being built, verifiable, valid, and usable to stakeholders. It is the responsibility of supplier to make sure that requirement specification adheres to factors given above. This is possible through reviews.

B. Simplistic Use Case Modeling

Although the use case modeling is useful for functional requirement it is not fit for non-functional requirements. While using use case modeling one should keep this in mind that many a times non-functional requirements play more important role. It becomes mandatory for requirement engineer to assure that the essential non-functional requirements are also taken care of.

C. Constraints

The requirement engineer should be able to distinguish between requirement and other design considerations such as architecture, design, implementation, and installation /configuration Constraints. Many times these are unnecessarily specified as requirements.

D. Requirements Tracing

Every requirement identified with elicitation should be traced to at least one of the system component. It is observed that many times requirement engineer fails to trace the requirement to system component or at least is unable to connect it to the component.

E. Missing requirements

Many times the requirement engineer emphasizes on functional requirements thus neglecting nonfunctional requirements.

Some of these include availability, interoperability, performance, portability, reliability, robustness, safety, security, stability, and usability. These are essential requirements that are expected from all professionally developed software's.

F. Requirements volatility

The customer and/or user have a myth about software and software development process. They assume that change can be implemented at any time in the software. This creates unnecessary problems for both the software and process. To tackle this problem change control should be in place because requirements constantly change; the stakeholders may add new requirements and may change existing constantly.

G. Verification of Requirements Quality

It is expected that after completion of elicitation formal technical review should be conducted this will help in identifying requirements defects. Immediately after defect identification requirement engineer should try to remove defect otherwise it will become costly affair and take significantly more time to fix.

H. Requirements Validation

As user requirements form base of validation/user acceptance testing a proper requirement validation technique should be established and used. The lack of proper requirements validation technique will result in incomplete requirements. This will lead to the system, not fulfilling user requirements and hence unacceptable to major classes in stakeholders. Fixing these problems later can have major negative impacts on cost and schedule, and some functionality may be missing upon delivery.

I. Requirements Management

If you want to manage requirements successfully instead of putting it on paper use some software tool to manage it. This will help you to create, manipulate, and maintain requirements it will also help to capture, analyze, and report requirements metrics (e.g., requirements stability, maturity, and completion).

J. Requirements Process

A well proven and well documented method should be used for requirement engineering. Otherwise you will get processes that produce poor products. The process shall help you to generate clear and complete set of requirements with appropriate requirements specification documents. If, requirements are clear and complete it will help architects, designers, implementers, and testers. If the methods are lacking in necessary detail the requirements engineer has to waste time arguing over what to do and how to do it.

K. Tool Support

Always use appropriate tool for requirements management to document user requirements. As requirement management is extremely labor-intensive without tool support you may find, inconsistencies significantly increase and the documented requirements easily get out-of-date.

L. Requirements Engineers

Not everyone is good at requirements engineering, it requires lots of efforts, training and skills to be good at it. In addition to this personal abilities are also important. The Requirements engineers should know various methods and its usage. Anybody who wants to be requirement engineer shall undergo sufficient training and gain necessary expertise in requirement engineering. This will make their job fun and interesting.

V. CONCLUSIONS

Requirement engineering is not easy, this may be true for those who are not well versed with the process, but it is possible to make it easy and fun. The problems discussed in this paper and corrective measures against it will help requirement engineers to understand and carry out the process in better manner. This will improve overall effectiveness of requirement engineering process thereby improving the quality of requirements and requirement specification. It will make the life of architect, designer, implementer and tester easy also it will improve overall quality of software.

REFERENCES

1. Dharendra Pandey, Vandana Pandey, requirement engineering: an approach to quality software development, Journal of Global Research in Computer Science, Volume 3, No. 9, September 2012, ISSN 2229-371X
2. Romi Satria Wahono, analyzing requirements engineering problems, Proceedings of the IECI Japan Workshop 2003 ISSN 1344-7491, Chofu Bunka Kaikan Tazukuri, Japan pp. 55 – 58
3. Hafeez MS1*, Farhan Rasheed1 and Khan MR2, An Improved Model for Requirement Management System, Journal of Information Technology & Software Engineering, 7: 196. doi: 10.4172/2165-7866.1000196
4. IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE, NY, 1990
5. Zielczynski P (2008) Requirements Management Using IBM® Rational® RequisitePro®. Pearson, InformIT
6. Michael Shrivathsan. Practical Requirement Management, Blog Post, <http://rmblog.accompa.com/2012/04/types-of-software-requirements/>
7. Donald Firesmith, Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them, JOURNAL OF OBJECT TECHNOLOGY, Vol. 6. No. 1, January-February 2007