Int. Res. J. of Science & Engineering, 2017; Vol. 5 (6) :153-158.
ISSN: 2322-0015
http://www.irjse.in
UGC Approved Journal No. 63628

RESEARCH ARTICLE

The Role of Requirement Engineering in Software Development Process

Sachhidanand

Asst. Professor, Govt. First Grade College, Aurad (B). Karnataka, India
Email: sachhi.r@gmail.com

Manuscript Details

Received: 20.11.2017
Accepted: 17.12.2017
Published: 30.12.2017

Available online on
https://www.irjse.inISSN: 2322-0015

Cite this article as:
Sachhidanand. The Role of Requirement Engineering in Software Development Process, *Int. Res. Journal of Science & Engineering,* 2017, Volume 5 (6) 2017: 153-158.

ABSTRACT

The software sector is becoming more globally integrated, which needs high-quality products to be delivered to customers. Business issues are translated into information system needs through the process of requirements engineering. Before we finish building something, we must comprehend what it is that we aim to make. Engineering requirements is crucial to creating high-quality software. The first stage of software development is requirements engineering, during which customer needs are obtained and recorded. Every company should strive to create high-quality software solutions that meet the expectations of their users. This paper's goals are to provide an overview of requirement engineering's importance for software development, demonstrate how it affects the process, and examine the requirement engineering procedure for creating.

Keywords: Engineering, Elicitation, Stakeholders, Risks, Requirement Specification.

INTRODUCTION

The process of developing software is dynamic. A project's and the market's success depends on requirements engineering. A prerequisite for software is a capability or state that a user needs in order to accomplish a task or address an issue. In software engineering, requirement engineering has always held a fundamental role. It frequently results in needs for software changing while it is still being developed.

Frequent modifications to software requirements have the potential to significantly increase project uncertainty. One of the primary causes of a project's challenges, according to reports, is requirements changes. The application of methodical, repeatable procedures to guarantee the accuracy, consistency, and applicability of the system requirements is emphasised in requirements engineering. The process of determining the services is called requirements engineering.

Functional Requirements (FR):

FR outlines how a product must work as seen by its end user. The features and functionalities that the end-user will directly engage with are described here.

Non-functional requirements:

Requirements that place limitations on the design or implementation are known as non-functional requirements. Diagrams used for requirement modelling do not typically include non-functional needs or specific functional requirements.

Business requirements:

These specify the motivation behind the product's development as well as the advantages it will offer to the company and its clients.

Use cases are used to record user needs, which outline the tasks or business processes that a user will be able to carry out with the product.

I. The Process of Elicitation of Requirements

1) Gathering requirements
2) Putting Things in Order
3) Talking and negotiating
4) Records

II. Techniques for Elicitation of Requirements

The several methods for finding requirements are as follows.

1) Conducting interviews is a conventional method that is often used. Analysts are able to comprehend the issue and learn about the goals of the application that has to be created.

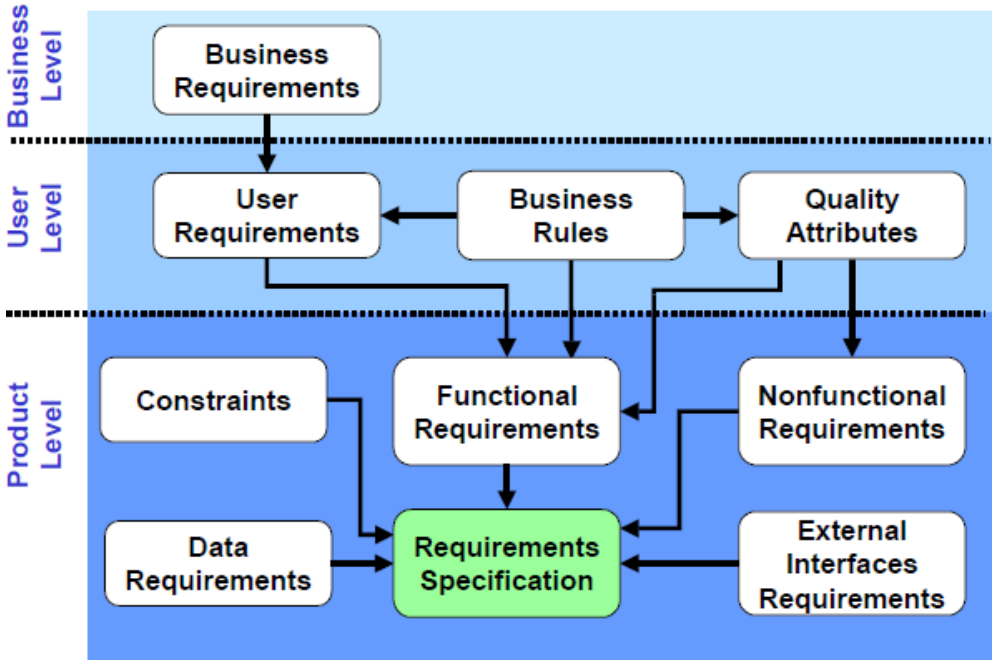

Fig.1 Levels of requirement. [2]

2) One potential substitute for interviews is JAD (Joint Application Development). This is a group technique in which all project stakeholders must take part. That includes users, system administrators, analysts, designers, and clients.

3) Another method for group meetings that is comparable to JAD is brainstorming. This entails gathering information and suggestions from all project stakeholders that have not yet been reviewed. There shouldn't be more than ten project stakeholders present at brainstorming sessions. Moderator duties must fall to one of them.

4) A common method employed by graphical designers in the creation of Web apps is sketching and storyboarding. The many user interfaces are schematically represented in this. By grouping and connecting the sketches with links, a "storyboard" that illustrates the navigational organisation can be created.

5) One technique that was developed more for defining requirements than for capturing them is use case modelling. Actors, use cases, and the relationships between them make up a use case model. Actors represent the environment, while use cases reflect the system's scope.

6) The Questionnaire and Checklist technique is creating a document with questions to which only succinct, specific responses are allowed.

Terminology by itself, comparison is not a solution to the requirements elicitation challenge. This complementing strategy helps users and developers who don't speak the same language communicate with each other more easily.

III. Putting the Needs in Place Engineering Procedure

The steps needed to start requirements engineering are as follows:

1) Stakeholder Identification: A stakeholder is any person who stands to gain, either directly or indirectly, from the system that is being designed.

2) Acknowledging Diverse Points of View: The requirements engineer's job is to classify all information from stakeholders, including contradictory requirements and inconsistencies, so that decision-makers can select a set of internally inconsistent system requirements.

3) Seeking Collaboration: Stakeholders cooperate by offering their perspectives on the needs.

IV. Methods of Requirements Engineering

Engineering requirements is crucial to creating high-quality software. The methodical process of gathering, evaluating, and modelling software requirements is known as requirement engineering, or RE.

1) Elicitation Procedures: IT is the process of determining and compiling the needs of multiple stakeholders. For three main reasons, stakeholders should be included in the definition of operational ideas and requirements.

2) Analysis Practices: The process of determining if a collection of requirements is acceptable, complete, valuable, and of high quality is known as requirements analysis.

3) Specification Practices: A needs specification serves as a foundation for cost and schedule estimation, a means of establishing a baseline for validation and verification, and it also helps suppliers and customers understand what is expected of a product.

4) Validation Practices: The goal of requirements validation is to show that artefacts meet predetermined standards and criteria. Validation of requirements engineering makes an effort to demonstrate that requirement statements follow the previously mentioned specification practices.

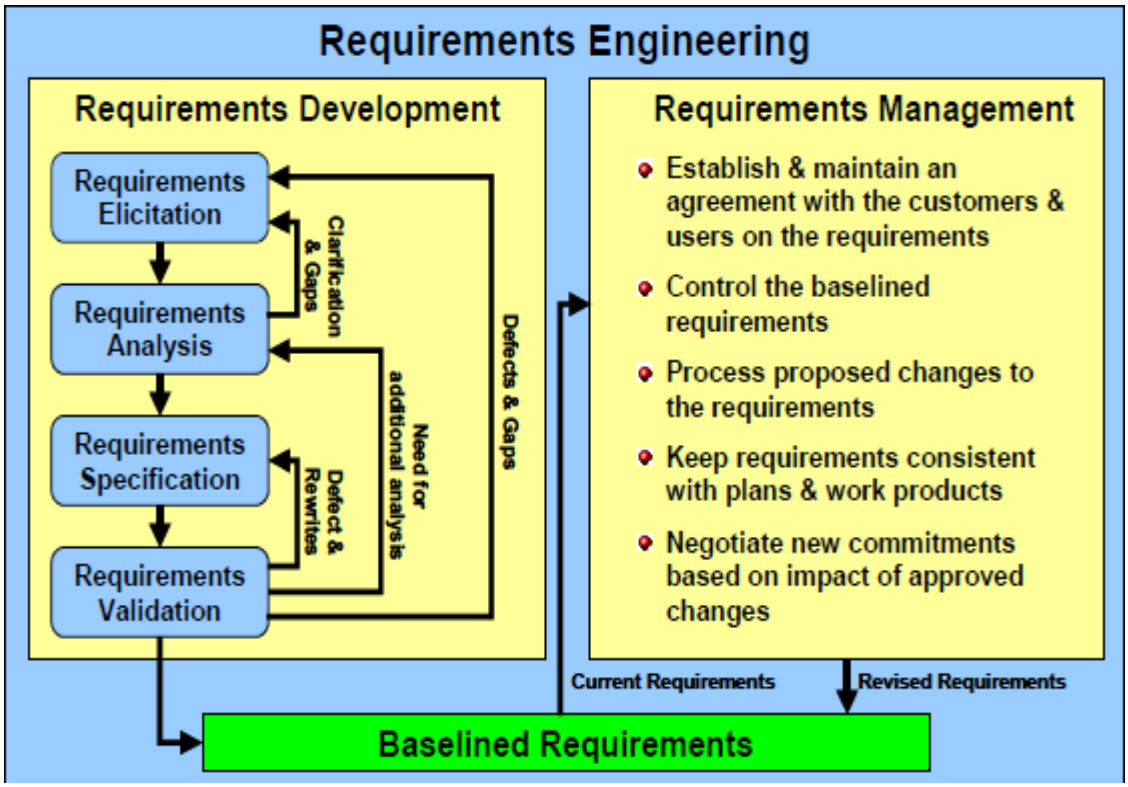

Fig.2 Requirements Engineering Process (based on Wiegers 2003) [2]

V. Requirements Engineering Risk

1) Insufficient customer representation: Reaching a consensus on needs is a key component of RE. We have to ascertain what your clients' actual needs are. If you never deal with them directly, at least a good deal of negotiation will occur.

2) Ignoring a vital requirement: Ignoring a functional or attribute requirement is the biggest danger in RE. We run the risk of losing projects when we ignore a crucial user class. It usually gets worse when a crucial quality or performance trait is overlooked.

3) Failure to inspect requirements: the expense of fixing requirements flaws rises exponentially over time. Once the software is deployed, fixing a requirement flaw will cost at least a hundred times as much.

4) Seeking to ensure all needs are met before starting construction: Before you can estimate the difficulty of the project and the cost of each component, some design and construction work must be completed. After identifying those areas of uncertainty, we must proceed to make sure that someone is in charge of filling in the gaps.

VII. Techniques for Prioritising Requirements

It is possible to prioritise using a variety of scales and methods.

The software community has utilised the Analytical Hierarchy Process (AHP), a systematic statistical technique based on relative assessment, to prioritise software requirements. When it's necessary to take into account both the qualitative and quantitative components of a decision, the Analytic Hierarchy Process (AHP) is a strong and adaptable tool for helping

individuals define priorities and choose the optimal course of action (Bagchi and Rao 1992).

Cumulative Voting (100 $ Test): The 100-dollar test, also known as cumulative voting, is a simple method where participants are given 100 fictitious units. The ratio scale is the foundation of this method. Grouping and Numerical Assignment: Its foundation is the ordinal scale. For reliable classification, requirements are organised into several groups that the stakeholders can understand.

Ranking: Various algorithms, such as bubble sort, binary search tree (Karlsson et al. 1998), Quick sort, and other sorting approaches (Hubbard 2000), may be employed to determine the rankings of the requirements [3]. When a single stakeholder's priorities need to be taken into consideration, as is typically the case in bespoke development, this method makes more sense. [4]

DISCUSSION

Because of its importance to the software development life cycle, requirement engineering has been the focus of a great deal of research. For practitioners, a variety of requirements engineering approaches and techniques are available certain methodologies [5] have been shown to work for some systems but not for others. Choosing the right requirements engineering methodologies and procedures can be challenging and, in the worst case, occasionally result in failure.

Requirement elicitation, requirements analysis & negotiation, requirements documentation, requirements verification & validation, and requirements management are the five separate processes that make up the requirement engineering process. Building products is the main goal of requirements engineering [6], and it excludes many other business analysis-related tasks including creating a business case, streamlining business processes, and providing advantages to the company.

CONCLUSION

The significance, ideas, and comprehension of requirement engineering for software development are presented in this work [7]. I have discussed the requirements engineering causal analysis in this work. The primary task in software development is requirements engineering. The foundation of software is requirement analysis. However, one of the most crucial duties that guarantees software quality was not using this. Customer requirements might be stated clearly or indirectly 22 [8]. It is necessary to change these implicit needs into explicit requirements [9]. We must develop customer-centred strategies and communication approaches that encourage customer participation in order to successfully comprehend the user and meet his wants with the products and contribute to the creation of a high-quality product.

REFERENCES

1. CS2 Software Engineering note 2", CS2Ah Autumn 2004
2. Linda Westfall," Software Requirements Engineering: What, Why, Who, When, and How",2005-2006.
3. Kashif Ahmed Khan," A Systematic Review of Software Requirements Prioritization"
4. Roger S. Pressman (2005)," Software engineering a practioner's approach 6/e"
5. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House
6. Ian Sommerville "Software Engineering" Pearson Education.
7. The Role of Requirement Engineering in Software Development Life Cycle, Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO. 5, May 2012, ISSN 2079-8407.
8. Requirement Engineering Issues and Their Solutions, Muhammad Waqas Boota, International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, Volume-2, Issue-11, November 2014.
9. Identify and Manage the Software Requirements Volatility, Khloud Abd Elwahab, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016.