Software Requirement

Vikkiraman Balarajan, Punithan Balarajan
Faculty of Computer Science and Information Technology
Universiti Selangor
Malaysia
2016

Abstract—Software Requirements is a field within software engineering that deals with establishing the needs of stakeholders that are to be solved by software. When developing a software, one of the most important aspects for success of any software project is to get the requirements right. The process to gather the software requirements from client, analyze and document them is known as requirement engineering [3]. If software requirements are not right, companies will not end up with the software they need. Two types of requirement; User requirements and System requirements

Keywords: Software, Requirement, User, System

I. Introduction

Software requirement is the most essential action in software development as substitute stages in the life cycle of software development relies on upon this important action. As the term as Software requirement, requirements planning is a careful to cover every one of the exercises included in discovery. Maintain and documenting of requirements for a standalone system [1]. An amounts of results may emerge because of wrong requirements. As for an example, the software might be delay in completion, cost increase than the first estimation, end-client won't be fulfilled, the software might be problematic and there might be normal software defects. As indicated by the overview directed by ESPI in 1995 that around 40-60% of all imperfections found in a product venture can be followed back to mistakes made within the requirements stage. As per another study directed by Standish Group Study, 1994 that 13.1% projects fail because of the incomplete of requirement gathering. In our opinion, Software requirement, user requirement and functional and non-functional plays main role in Software requirement. [5]

II. What is a Software Requirement?

Requirements must be determined and agreed to by the customers, users, and suppliers of a software product before the software can be built [3]. Also requirements can help us to trace the quality of the software product

III. User requirements

Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.

The User Requirements Specification should include:

- Introduction – including the scope of the system, key objectives for the project, and the applicable regulatory concerns
- Program Requirements – the functions and workflow that the system must be able to perform
- Data Requirements – the type of information that a system must be able to process
- Life Cycle Requirements – including how the system will be maintain and users trained

User Requirements Specifications should be signed by the system owner, key end-users, and Quality. Once approved, the URS is retained according to your organization’s practices for document retention [3].

IV. System requirements

An organized document setting out detailed explanations of the system’s functions, services and operational constraints. Describes what should be applied so may be part of a contract between client and contractor.

For instance a software requirement,

- The user requirement is to compute the correct value.

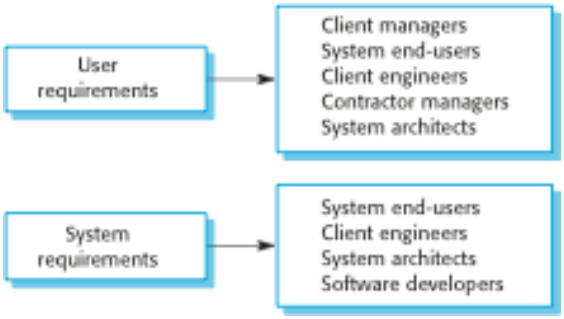
- But the system requirement is only to compute the correct quantity of the limited revenues entered by the user.

If the user enters incorrect partial revenues the software is not required to magically correct them: The output will be the correct as of the inputs, but not the correct overall revenue [2]. The distinction is not excessively intriguing for most basic data frameworks. Questions to ask before the start of any software development project are:

- Why are we building the system?
- What do we need it to do?
- What benefits are we expecting to get from it?

V. Example Readers of different types of requirements specification and specifications

User requirements flow



VI. Why Gather Customer Requirements?

Here are some of the reasons for gathering requirements [5]

- To arrange opinions and thinking in a logical way
- To organize someone else's thoughts and ideas in a reasonable way
- To understand what a software package requirement do beforehand making a choice
- As a point of reference throughout the project

Requirements gathering is an important part of project management and software development [3]. The main purpose is to create a clear, concise and agree d set of requirements that allow you to deliver what the customer needs. From these requirements, developer design and deliver a powerful and credible software solution.

VII. Functional requirement

Functional requirement is describe the main functions of the software. Functional requirement's element is data and functional process requirement [5]

VIII. Functional Process Requirement

Functional Process requirement should clarify what the software main task. Besides that [5], Process requirements relate the entities and attributes from the data requirements to the users' needs. Functional process requirement permit the user understanding the system flow pattern by the increasing numbering in very specifics.

Functional Requirements should include:

- Descriptions of data to be entered into the system
- Descriptions of operations performed by each screen
- Descriptions of work-flows performed by the system
- Descriptions of system reports or other outputs
- Who can enter the data into the system
- How the system meets applicable regulatory requirements

The Functional Requirements Specification is designed to be read by a general audience. Readers should understand the system, but no particular technical knowledge should be required to understand the document [6].

IX. Types of functional requirements

In some cases a requirements analyst generates use cases after gathering and validating a set of functional requirements. The hierarchy of functional requirements is:

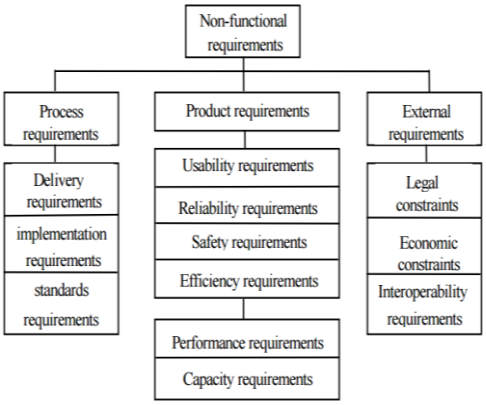
User / stakeholder request → feature → use case → business rule.

Each use case illustrates behavioral scenarios through one or more functional requirements

X. Non Functional requirement

There are other requirement in the software which is never do any contribution to software functions. It is called as Non-functional requirement or also knows as Qualities attributes. [4] As for an Example, there are functions as security, performance, compatibility is non-functional requirement but it's not a feature of the software but is an important element for it. Some Nonfunctional requirement is listed as

- Performance requirement
- Operating limitation
- Platform limitation
- Customization ability
- Portability
- Reliability
- Security
- Usability
- Legal



Classification chart of nonfunctional requirements

XI. How to improve modern software requirement

The software product quality model provided in ISO/IEC 9126-1 (Iso9126-01) defines six quality characteristics: functionality, reliability, usability, maintainability, portability and efficiency, as well as quality in use, which is defined as effectiveness, productivity. Safety and satisfaction. During the data gathering, it is a must to follow all the characteristics to build a success software deliver by know the need [6].

REFERENCES

[1] Autumn. (2004). Software Requirements. 9. Retrieved from http://www.inf.ed.ac.uk/teaching/courses/cs2/LectureNotes/CS2Ah/SoftEng/se02.pdf

[2] Cleland-Huang, J. (2014). Software Requirements. Retrieved from https://www.researchgate.net/publication/228381037_Software_Requirements

[3] Engineering, N. S. (1995). Retrieved from http://worrydream.com/refs/Brooks-NoSilverBullet.pdf

[4] Glinz, M. (2000). Improving the Quality of Requirements with Scenarios.

[5] Reuters, T. (2015). JOURNAL OF SYSTEMS AND SOFTWARE. Retrieved from https://www.elsevier.com/journals/journal-of-systems-and-software/0164

[6] SHAMS-UL-ARIF, M. (2010). REQUIREMENTS ENGINEERING. Retrieved from http://www.ijric.org/volumes/Vol2/6Vol2.pdf

[7] Wiegers, K. E. (1996). Software Process Improvement: Ten Traps to Avoid. Retrieved from http://www.compaid.com/caiinternet/ezine/wiegers-spitraps.pdf

[8] An Overview Of Software Quality Concepts And Management Issues. (2011). Retrieved from http://www.etsmtl.ca/Professeurs/claporte/documents/publications/Duggan_Chapter_SQA.pdf

ICF

ICF

ICF

ICF

ICF