International Journal of Education, Modern Management, Applied Science & Social Science (IJEMMASSS)
ISSN : 2581-9925, Impact Factor: 6.882, Volume 04, No. 04(II), October - December, 2022, pp. 90-92

ROLE OF SOFTWARE REQUIREMENTS SPECIFICATIONS IN SOFTWARE ENGINEERING

Amit Khatri[*]

ABSTRACT

This paper describes the role of Software requirements specification (SRS) in software Engineering. SRS provides the brief idea about the software system or product would do and how it would perform. SRS commonly focus on the functional requirements of the software to be developed. In the SRS phase, the detailed requirement is gathered from the user by conducting face to face communication or any other way and based on that information software is developed by the developers' team. This paper explores the quality aspects in SRS and defined attributes that contribute to that quality.

Keywords: *Software Requirements Specifications, Quality Attributes, User and Developers, Security.*

Introduction

Software Requirement Specification (SRS) is the description of a software system to be developed. It could be written by the developer of the system and secondly by the customer of the system. Both the two scenarios have different purposes and have different situations for the document. In the first case SRS is written for different purpose and it is a contract between customer and developer of the system and in the second case SRS is used to define the actual need and expectation of the user. It lays functional and non-functional requirements of the software to be developed. E.g., University Management System. Functional Requirements which are related to functional working aspect of a software fall into this category. Non-functional requirements are expected characteristics of a targeted software like security, storage, configuration, performance, cost, interoperability, and flexibility of the software. It may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Structure of SRS

- **Purpose:** It refers to the description of the software to be developed and why it is to be developed.
- **Scope:** It identify the future scope of the software.
- **Definitions:** It includes all the definitions of all the terms that are used, all the acronyms and abbreviations used in the SRS.
- **References:** It specify the sources and provide complete list of documents through which the information and reference been taken like online media etc.

Nature of SRS

- **Functionality:** It describes what the software is supposed to do
- **External interface:** It describes about the system hardware and software and how the software does interact with the user.
- **Performance:** It talks about the speed, recovery time in case of any failure, availability etc. of the software functions.
- **Attributes:** This part of SRS includes portability, correctness, ambiguity, maintainability, security etc.
- **Design constraints:** It contains the policies for database integrity, resource limits, operating environment.

As we know the Software Requirements Specifications has a specific role to play in the software development process so the writers of SRS should be careful not to go beyond the bounds of that role. This means that SRS should define all the software requirements correctly and it should not contain design and implementation details. These should be defined in the design stage of the project.

---

[*] M.Tech, Computer Science.

Characteristics of SRS

- **Correct:** What is stated is exactly what is desired and expected functionality matches the requirement present in SRS.
- **Consistency:** There should be no conflicts between the requirements. Every requirement must be specified using a standard terminology.
- **Unambiguous:** Every stated requirement has only one unique meaning and the words used in SRS should be specified with meaning. Software requirement specification language can be used to remove ambiguity.
- **Complete:** It includes all functional and non-functional requirements, constraints. It specifies expected output from all kinds (valid/invalid) inputs from the user.
- **Traceable:** Origin of each requirement should be clear. It is important because future references may be required for development or maintenance phase.
- **Verifiable:** SRS is verifiable if and only if each requirement is verifiable. If there isa process to check if the software meets each of the requirements.
- **Modifiable:** Easy to make changes in SRS retaining its structure.
- **Ranked for stability/importance:** SRS requirements should be rank according to their importance like essential, conditional, optional.

Attributes of SRS

There are no of quality attributes of a software system that can serves as a requirements:

- **Reliability:** It specify probability of failure free operation for a given time duration. Failure is an execution event where the software behaves unexpectedly.
- **Availability:** It includes the factors required to guarantee a defined availability level for the software system such as recovery and restart.
- **Security:** It includes the factors that are required to protect the software from malicious action, use, fraudulent, destruction etc.
- **Maintainability:** It deals with how ease with which software system can be modified to correct failure, improve performance or other attributes.
- **Portability:** It specifies the effort needed to transfer from one hardware or software environment to another.
- **Reusability:** It specifies how the one software can be used in another application.
- **Testability:** How easy software is to test.
- **Interoperability:** How easily a software system can be coupled with another.
- **Efficiency:** It refers to number of resources and code required to perform a function.

Uses of SRS

- To fulfilling the need of developing product, development teams need SRS.
- Testing group creates test plans based on the detail external efforts.
- Maintenance team need it to identify what the software product is expected to do.
- Project team manage their estimates, costs, resources on it.
- It is used for documentation purpose.
- It is used as agreement between customer and developer.
- End user can rely on it.

Importance of SRS

- At the initial stage it gives quick idea about the software to the client and the user.
- The objectives and the goals as well as the expected results are defined properly.
- It provides the outline for software design.
- It provides a basis for the contract between the user and developer.
- It reduces the efforts of the developer's team in terms of cost and time.
- It can be used as a reference at later stage.
- It provides the basis for reviews.

For writing the SRS following steps should be follow:

- The developer should listen the user and concentrate on their requirements.
- Identification and understanding the problem before you meet.

- There should be agenda and the someone who facilitate the meeting.
- Developer and user should have face to face communication.
- Presentations and discussion on the requirements.
- Strive for collaborations and decisions.
- Document all the decisions.

Problems Without an SRS Document

- Without SRS document it would be difficult for the development team to develop a software according to the customer needs.
- Without SRS document the developers will not know whether they are building the product according to the user need.
- It would be difficult for the software engineers to understand the basic functionality of product at the maintenance phase.
- Without understanding the SRS documents the user document would face problems writing the user's manual.

Advantages of good SRS

- A good SRS provides the basis for contract between the developer and the customer on what the product will perform.
- A good quality SRS reduces the development cost and time.
- A good SRS provides the references of the final product.
- A good SRS helps in ranking the requirements.

Conclusion

SRS is helpful for both the users and software development team. SRS helps the user to define their requirements with accuracy, while it also helps the software development team in understanding what the user wants in terms of development. SRS should be kept upgraded and evaluated throughout the building of project. Putting efforts and time in writing the SRS document will lead to successful development of the product that the end user wants.

References

1. A. M. Davis, Ó. Dieste, A. M. Hickey, N Juristo, A. Moreno, "Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review". 14th International Conference on Requirements Engineering (RE 2006), IEEE Computer Society, 11-15 September 2006, Minneapolis, USA, pp. 176-185.
2. Guide to Software Requirements Specifications, ANSI/IEEE Std.830-1984, 1984
3. Effective Requirements Definition and Management: Improves Systems and Communication (White paper), Ed. 8310 North Capital of Texas Highway, Corporate Office: Whitepaper of Borland Software Corporation - The open ALM Company, 2009.
4. Elrsili, V., Md D. Weiu, 'Evalrution of a SohareRqukmrh Document by Analyob of Change Data," RphIEEEInt? an &+ hg., 1981, pp. 314-323.
5. A. Herrmann, M. Daneva: Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research, 16th International Requirements Engineering Conference, IEEE Computer Society, 8-12 September 2008, Barcelona, Spain, pp. 125-134
6. Davis, A.M.: Software Requirements, Objects, Functions, & States, F'rentice-Hall, Inc., NJ, 1993.
7. DeMarco, T.: Structured Analysis and System Specification, F'rentice-Hall, Inc., NJ, 1978.
8. Myers, G. J.: Composite/Structured Design, Van Nostrand Reinhold Co., 1978.

❖◆❖