SOFTWARE RQUIREMENTS SPECIFICATION(SRS)

The output of the requirements phase of the software development process is **Software Requirements Specification (SRS).** This document lays a foundation for software engineering activities and is created when entire requirements are elicited and analyzed. SRS is a formal document, which acts as a representation of software that enables the users to review whether it (SRS) is according to their requirements. In addition, it includes user requirements for a system as well as detailed specifications of the system requirements.

IEEE defines software requirements specification as, 'a document that clearly and precisely describes each of the essential requirements (functions, performance, design constraints and quality attributes) of the software and the external interfaces. Each requirement is defined in such a way that its achievement can be objectively verified by a prescribed method, for example, inspection, demonstration, analysis or test.

Each SRS fits a certain pattern; thus, it is essential to standardize the structure of the requirements document to make it easier to understand. For this IEEE standard is used for SRS to organize requirements for different projects, which provides different ways of structuring SRS. Note that in all requirements documents, the first two sections are the same.Organisations like **IBM,TCS** refer the standard format provided by the **IEEE.**

This document comprises the following sections.

1. **Introduction:** This provides an overview of the entire information described in SRS. This involves purpose and the scope of SRS, which states the functions to be performed by the system. In addition, it describes definitions, abbreviations, and the acronyms used. The references used in SRS provide a list of documents that is referenced in the document.
2. **Overall description:** It determines the factors which affect the requirements of the system. It provides a brief description of the requirements to be defined in the next section called 'specific requirement'. It comprises the following sub-sections.
3. **Product perspective:** It determines whether the product is an independent product or an integral part of the larger product. It determines the interface with hardware, software, system, and communication. It also defines memory constraints and operations utilized by the user.
4. **Product functions:** It provides a summary of the functions to be performed by the software. The functions are organized in a list so that they are easily understandable by the user:
5. **User characteristics:** It determines general characteristics of the users.
6. **Constraints:** It provides the genera1 description of the constraints such as regulatory policies, audit functions, reliability requirements, and so on.
7. **Assumption and dependency:** It provides a list of assumptions and factors that affect the requirements as stated in this document.

8. **Apportioning of requirements:** It determines the requirements that can be delayed until release of future versions of the system.
9. **Specific requirements:** These determine all requirements in detail so that the designers can design the system in accordance with them. The requirements include description of every input and output of the system and functions performed in response to the input provided. It comprises the following subsections.
10. **External interface:** It determines the interface of the software with other systems, which can include interface with operating system and so on. External interface also specifies the interaction of the software with users, hardware, or other software. The characteristics of each user interface of the software product are specified in SRS. For the hardware interface, SRS specifies the logical characteristics of each interface among the software and hardware components. If the software is to be executed on the existing hardware, then characteristics such as memory restrictions are also specified.
11. **Functions:** It determines the functional capabilities of the system. For each functional requirement, the accepting and processing of inputs in order to generate outputs are specified. This includes validity checks on inputs, exact sequence of operations, relationship of inputs to output, and so on.
12. **Performance requirements:** It determines the performance constraints of the software system. Performance requirement is of two types: static requirements and dynamic requirements. **Static requirements** (also known as **capacity requirements)** do not impose constraints on the execution characteristics of the system. These include requirements like number of terminals and users to be supported. **Dynamic requirements** determine the constraints on the execution of the behavior of the system, which includes response time (the time between the start and ending of an operation under specified conditions) and throughput (total amount of work done in a given time).
13. **Logical database of requirements:** It determines logical requirements to be stored in the database. This includes type of information used, frequency of usage, data entities and relationships among them, and so on.
14. **Design constraint:** It determines all design constraints that are imposed by standards, hardware limitations, and so on. Standard compliance determines requirements for the system, which are in compliance with the specified standards. These standards can include accounting procedures and report format. Hardware limitations implies when the software can operate on existing hardware or some pre-determined hardware. This can impose restrictions while developing the software design. Hardware limitations include hardware configuration of the machine and operating system to be used.
15. **Software system attributes:** It provide attributes such as reliability, availability, maintainability and portability. It is essential to describe all these attributes to verify that they are achieved in the final system.

16. **Organizing Specific Requirements:** It determines the requirements so that they can be properly organized for optimal understanding. The requirements can be organized on the basis of mode of operation, user classes, objects, feature, response, and functional hierarchy.
17. **Change management process:** It determines the change management process in order to identify, evaluate, and update SRS to reflect changes in the project scope and requirements.
18. **Document approvals:** These provide information about the approvers of the SRS document with the details such as approver's name, signature, date, and so on.
19. **Supporting information:** It provides information such as table of contents, index, and so on. This is necessary especially when SRS is prepared for large and complex projects.

IBM references the standard format provided by **IEEE.** It has the following SRS structure format.

1.INTRODUCTION

2.OVERALL DESCRIPTION

3.SPECIFIC REQUIREMENTS

The format covers a lot of diagrams to understand and achieve the system and user requirements and to make the document more specific, correct and unambiguous. OVERALL DESCRIPTION of the document highly focusses on the non-functional requirements (which conveys how the system is supposed to be). The different OOAD based conceptual representations used to describe non-functional requirements are:

1. USE-CASE DIAGRAMS
2. WEB ARCHITECTURE DIAGRAMS
3. ER DIAGRAM
4. ARCHITECTURE DIAGRAM
5. DATA DICTIONARY etc.

The Functional requirements(what the system is supposed to do) of the system is described under SPECIFIC REQUIREMENTS. CLASS DIAGRAMS, USE CASE REPORTS, SUPPLEMENTARY REQUIREMENTS are used as a measure to describe the functional requirements are: