Software Requirements Document

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is not a design document. As far as possible, it should set of what the system should do rather than how it shoulddo it.

Software Requirements Specification

The software requirements provide a basis for creating the Software Requirements Specifications (SRS).

The SRS is useful in estimating cost, planning team activities, performing tasks, and tracking the team's progress throughout the development activity.

Typically software designers use IEEE STD 830-1998 as the basis for the entireSoftware Specifications. The standard template for writing SRS is as given below.

1. Introduction

- **Purpose of this document**

  Describes the purpose of the document.

- **Scope of this document**

  Describes the scope of this requirements definition effort. This section also details any constraints that were placed upon the requirements elicitation process, such as schedules, costs.

- **Overview**

  Provides a brief overview of the product defined as a result of the requirements elicitation process.

2. General Description

- Describes the general functionality of the product such as similar system information, user characteristics, user objective, general constraints placed on design team.
- Describes the features of the user community, including their expected expertise with software systems and the application domain.

3. Functional Requirements

This section lists the functional requirements in ranked order. A functional requirement describes the possible effects of a software system, in other words, *what* the system must accomplish. Each functional requirement should be specified in following manner

Short, imperative sentence stating highest ranked functional requirement.

1. **Description**
   A full description of the requirement.
2. **Criticality**
   Describes how essential this requirement is to the overall system.
3. **Technical issues**

Describes any design or implementation issues involved in satisfying this requirement.

4. **Cost and schedule**

   Describes the relative or absolute costs of the system.

5. **Risks**

   Describes the circumstances under which this requirement might not able to be satisfied.

6. Dependencies with other requirements

   Describes interactions with other requirements.

4. Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

- **User Interfaces**
- **Hardware Interfaces**
- **Communications Interfaces**
- **Software Interfaces**

5. Performance Requirements

Specifies speed and memory requirements.

6. Design Constraints

Specifies any constraints for the design team such as software or hardware limitations.

7. Other Non-Functional Attributes

Specifies any other particular non functional attributes required by the system. Suchas :

- Security
- Binary Compatibility
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Application Compatibility
- Resource Utilization
- Serviceability

8. Operational Scenarios

This section should describe a set of scenarios that illustrate, from the user'sperspective, what will be experienced when utilizing the system under various situations.

9. Preliminary Schedule

This section provides an initial version of the project plan, including the major tasks tobe accomplished, their interdependencies, and their tentative start/stop dates.

10. Preliminary Budget

This section provides an initial budget for the project.

11. Appendices

11.1 Definitions, Acronyms, Abbreviations

Provides definitions terms, and acronyms, can be provided.

11.2 References

Provides complete citations to all documents and meetings referenced.

Characteristics of SRS

Various characteristics of SRS are

- Correct - The SRS should be made up to date when appropriate requirements are identified.
- Unambiguous - When the requirements are correctly understood then only it is possible to write an unambiguous SRS.
- Complete - To make the SRS complete, it should be specified what a software designer wants to create a software.
- Consistent - It should be consistent with reference to the functionalities identified.
- Specific - The requirements should be mentioned specifically.
- Traceable - What is the need for mentioned requirement? This should be correctly identified.