**Characteristics of a Good SRS**

A good SRS document has certain characteristics that must be present. Thecharacteristics are:1.

**Correctness:**

An SRS is correct if every requirement included in the SRSrepresents something required in the final system.2.

**Completeness.**

An SRS is complete when it is documented after:(

*i*

) The involvement of all types of concerned personnel.(

*ii*

) Focusing on all problems, goals, and objectives, and not only on functionsand features.(

*iii*

) Correct definition of scope and boundaries of the software and system.3.

**Unambiguous.**

An SRS is unambiguous if and only if every requirementstated has one and only one interpretation. Requirements are often written ina natural language. The SRS writer has to be especially careful to ensure thatthere are no ambiguities. One way to avoid ambiguities is to use some formalrequirements specification language. The major disadvantage of using formallanguages is the large effort required to write an SRS, the high cost of doingso, and the increased difficulty of reading and understanding formally statedrequirements (particularly by the users and clients).4.

**Verifiable.**

An SRS is verifiable if and only if there exists some cost-effective process that can check whether the final product meets the requirements.5.

**Modifiable.**

An SRS is modifiable if its structure and style are such that anynecessary change can be made easily while preserving completeness andconsistency. The presence of redundancy is a major hindrance to modifiability,as it can easily lead to errors. For example, assume that a requirement isstated in two places and that the requirement later needs to be changed. If only one occurrence of the requirement is modified, the resulting SRS will beinconsistent.6.

**Traceable.**

The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each requirement in future developmentor enhancement documentation. Two types of traceability are recommended:(

*i*

)

*Backward traceability*

. This depends upon each requirement explicitlyreferencing its source in earlier documents.(

*ii*

)

*Forward traceability*

. This depends upon each requirement in the SRShaving a unique name or reference number.7.

**Consistency.**

Consistency in the SRS is essential to achieve correct resultsacross the system. This is achieved by:(

*i*

) The use of standard terms and definitions.(

*ii*

) The consistent application of business rules in all functionality.(

*iii*

) The use of a data dictionary.INTRODUCTION TO SOFTWARE REQUIREMENTS SPECIFICATION

**79**

(

*iv*

) The lack of consistency results in an incorrect SRS and failure in deliverablesto customer.

8.

**Testability.**

An SRS should be written in such a way that it is possible to createa test plan to confirm whether specifications can be met and requirements can be delivered. This is achieved by:(

*i*

) Considering functional and non-functional requirements.(

*ii*

) Determining features and facilities required for each requirement.(

*iii*

) Ensuring that ‘users’ and ‘stakeholders’ freeze the requirement.9.

**Clarity.**

An SRS is clear when it has a single interpretation for the author (analysis), the user, the end user, the stakeholder, the developer, the tester,and the customer. This is possible if the language of the SRS is unambiguous.Clarity can be ascertained after reviewing the SRS by a third party. It can beenhanced if the SRS includes diagrams, models, and charts.10.

**Feasibility.**

RDD-SRS needs to be confirmed on technical and operationalfeasibility. The SRS often assumes the use of technology and tools based onthe information given by their vendors. It needs to be confirmed whether the technology is capable enough to deliver what is expected in the SRS. Theoperational feasibility must be checked through environment checking. It isassumed that sources of data, user capability, system culture, work culture,and other such aspects satisfy the expectation of the developer. These must beconfirmed before development launch