3.2 SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Many software engineers treat the SRS document as a reference document. However, it is more appropriate to think of the SRS document as a contract between the development team and the customer, which can also be used to resolve any disagreements which may arise in the future. Once the customer agrees to the SRS document, the development team proceeds to develop the product conforming to all the requirements mentioned in the SRS document.

3.2.1 SRS Document

An SRS document should clearly document the following:

- Functional requirements of the system.
- Non-functional requirements of the system.
- Constraints on the system.

We have already emphasized that it is desirable to consider most systems as performing a set of functions $\{f_i\}$. Each function f_i of the system can be considered as performing a transformation of a set of input data to the corresponding set of output data. The functional requirements of the system should clearly describe each of the functions that the system needs to perform along with the corresponding input and output dataset. Non-functional requirements deal with the characteristics of the system that cannot be expressed functionally, e.g. maintainability, portability, usability, etc. The non-functional requirements also include reliability issues, accuracy of results, human-computer interface issues, operating and physical constraints, etc. The constraints on the functions of a system may describe certain things that the system should or should not do. For example, a constraint on a function can describe how fast it can produce results so that it does not overload the functions of another system to which it supplies data.

3.2.2 Characteristics of a Good SRS Document

A good SRS document should possess the following qualities:

- It should be concise and at the same time unambiguous.
- It should be consistent.
- It should be complete.
- It should be well-structured and easily modifiable. In practice, the SRS document undergoes several revisions, because customer requirements often evolve over a period of time. Therefore, it is important to make the document well-structured so that recording of modifications becomes easy.
- It should specify *what the system must do and not state how to do it*. Thus the SRS document should specify the external behaviour of the system only and refrain from discussing any implementation issues. In other words, the SRS document should view the system to be developed as a black box, and should specify the externally visible behaviour of the system. For this reason, the SRS document is also called the *black-box specification*.
- It should specify all goals and constraints concerning implementation. A **goal** is a statement that guides trade-off among design decisions. For example, a customer may express concern more about maintainability of the software, but may not care much about efficiency. A goal may become a requirement, if it can be expressed quantitatively.
- It should record references to maintainability, portability, and adaptability. **Maintainability** is the collection of qualities connected

with how easy it is for the system to be changed. **Portability** refers to the ease of moving the system to a new operating system or a new hardware platform. **Adaptability** refers to the ease of changing the system to meet predictable new requirements, e.g. a payroll program may need to be adapted to yearly changes in tax laws.

- It should characterize acceptable responses to undesired events.
- It should show conceptual integrity so that the users of the system may easily understand it.