**COMPONENTS OF SRS**

The following requirements are used in the specification of the SRS:

1. Functional requirements

2. Performance requirements

3. Design constraints

4. External interface requirements

**FIGURE 3.10** Components of SRS Document

1. **Functional Requirements.** Functional requirements specify which outputs

should be produced from the given inputs. They describe the relationship

between the input and output of the system. For each functional requirement,

**76** SOFTWARE ENGINEERING AND TESTING

a detailed description of all the data inputs and their sources, the units of

measure, and the range of valid inputs must be specified.

All the operations to be performed on the input data to obtain should be

specified. This includes specifying the validity checks on the input and output

data, parameters affected by the operation, and equations or other logical

operations that must be used to transform the inputs into corresponding

outputs. For example, if there is a formula for computing the output, it should

be specified. Care must be taken not to specify any algorithms that are not

part of the system but that may be needed to implement the system. These

decisions should be left for the designer. In addition some abnormal input,

system behavior for invalid inputs, must be specified.

2. **Performance Requirements.** This part of an SRS specifies the performance

constraints on the software system. All the requirements relating to the

performance characteristics of the system must be clearly specified. There are

two types of performance requirements: static and dynamic.

Static requirements are those that do not impose constraints on the execution

characteristics of the system. These include requirements, such as the number of

terminals to be supported, the number of simultaneous users to be supported,

and the number of files that the system has to process and their sizes. These

are also called capacity requirements of the system.

Dynamic requirements specify constraints on the execution behavior of the

system. These typically include response time and throughput constraints

on the system. Response time is the expected time for the completion of an

operation under specified circumstances. Throughput is the expected number

of operations that can be performed in a unit time. For example, the SRS may

specify the number of transactions that must be processed per unit time,

or what the response time for a particular command should be. Acceptable

ranges of the different performance parameters should be specified, as well as

acceptable performance for both normal and peak workload conditions.

3. **Design Constraints.** There are a number of factors in the client’s environment

that may restrict the choices of a designer. Such factors include standards

that must be followed, resource limits, operating environment, reliability

and security requirements, and policies that may have an impact on the

design of the system. An SRS should identify and specify all such constraints,

including:

(*i*) *Standards Compliance*. This specifies the requirements for the standards

the system must follow. The standards may include the report format and

accounting procedures. There may be audit-tracing requirements, which

require certain kinds of changes, or operations that must be recorded in an

audit file.

INTRODUCTION TO SOFTWARE REQUIREMENTS SPECIFICATION **77**

(*ii*) *Hardware Limitations*. The software may have to operate on some existing

or predetermined hardware, thus imposing restrictions on the design.

Hardware limitations can include the type of machines to be used,

operating system available on the system, languages supported, and

limits on primary and secondary storage.

(*iii*) *Reliability and Fault Tolerance*. Fault-tolerance requirements can place a

major constraint on how the system is to be designed. Fault-tolerance

requirements often make the system more complex and expensive.

Recovery requirements must specify what the system should do if some

fault occurs. Recovery requirements are often an integral part of the design

constraints.

(*iv*) *Security*. Security requirements are particularly significant in defense

systems and many database systems. Security requirements place

restrictions on the use of certain commands, control access to data, provide

different kinds of access requirements for different people, require the

use of passwords and cryptography techniques, and maintain a log of

activities in the system. Given the current security needs even of common

systems, they may also require proper assessment of security threats,

proper programming techniques, and use of tools to detect flaws, such as

buffer overflow.

4. **External Interface Requirements.** All the interactions of the software with

people, hardware, and other software should be clearly specified. For the

user interface, the characteristics of each user interface of the software product

should be specified. User interface is becoming increasingly important and

must be given proper attention. A preliminary user manual should be created

with all user commands, screen formats, an explanation of how the system will

appear to the user, and feedback and error messages. Like other specifications

these requirements should be precise and verifiable. So, a statement like

“the system should be user friendly” should be avoided and statements like

“commands should be no longer than six characters” or “commands names

should reflect the function they perform” should be used.

For hardware interface requirements, the SRS should specify the logical

characteristics of each interface between the software product and the

hardware components. If the software is to execute on existing hardware or

on predetermined hardware, all the characteristics of the hardware, including

memory restrictions, should be specified. In addition, the current use and load

characteristics of the hardware should be given.

The interface requirement should specify the interface with other software the

system will use or that will use the system. This includes the interface with the

operating system and other applications. The message content and format of

each interface should be