

Software Requirement Specifications

Authors...

1 Introduction

1.1 Purpose

Purpose of the SRS and specify the audience of the SRS.

1.2 Scope

Identify the software products to be produced by name. Explain what the product will/will not do. Describe the application of software being specified; relevant benefits, objectives, and goals. Consistent with high level specs.

1.3 Definitions

Define all terms, acronyms and abbreviations. May refer to appendixes or other docs.

1.4 References

Complete list of docs referred to by the SRS. ID docs by title/report num/ date/ publisher. Specify sources from which ref can be obtained.

1.5 Overview

Describe the rest of the SRS. Explain how SRS is organized.

2 Overall description

Describe the general factors that affect the product and its requirements. Does not state specific requirements. Provide background for those requirements that are defined in section 3 and clarify them.

2.1 Product perspective

Put the product into perspective with other related products. State if product is independent/ totally self contained. If product is component of larger system, relate the requirements to the larger system and identify interfaces between system and the software.

2.1.1 System Interface

List each system interface and identify the functionality of the software requirements and the interface description to match the system.

2.1.2 User Interface

Logical characteristics of each interface between the software product and its users. Include configuration characteristics necessary to accomplish the software requirements.

Include aspects of optimizing the interface with the person who must use the system. Maybe list of do's and don'ts on how the system will appear to the user. Long/short error messages. Must be verifiable.

2.1.3 Hardware interfaces

Specify logical characteristics of each interface between the software and the hardware components. Include configuration characteristics. Cover what devices are supported, how they are supported and protocols.

2.1.4 Software interfaces

Specify use of other required software products and interfaces with other application systems. For each required software product, provide: Name, mnemonic, specification number, version number and source.

For each interface: discuss the purpose of interfacing software as related to this product. Define the interface in terms of message content and format.

2.1.5 Communication interfaces

Specify the various interfaces to communications such as local network protocols etc.

2.1.6 Memory constraints

Specify any applicable characteristics and limits on primary and secondary memory.

2.1.7 Operations

Specify special and normal operations required by the user such as: The various modes of operation in the user organization; user initiated operations. Periods of interactive operations and periods of unattended operations. Data processing support functions. Backup and recovery operations.

2.1.8 Site adaptation requirements

Define the requirements for any data or initialized sequences that are specific to a given site, mission or operational mode, ie grid values, safety limits etc. Specify the site or mission related features that should be modified to adapt the software to a particular installation.

2.2 Product functions

Provide summary of the major functions of software. Organize function list in a way that is understandable to customer. Text/graphical methods to show different functions and relationships. Not intended to show design of a product but to show logical representation among variables.

2.3 User characteristics

Describe general characteristics of the intended user of the product including education level, experience and technical expertise. Not used to state specific requirements but should provide the reason why certain specific requirements are in section 3.

2.4 Constraints

Provide general description of any other items that will limit the developer's options.

- (a) Regulatory Policies
- (b) Hardware Limitations
- (c) Interface to other applications
- (d) Parallel Operations
- (e) Audit functions

- (f) Control functions
- (g) Higher order language requirements
- (h) Single handshake protocols
- (i) Reliability requirements
- (j) Criticality of the application
- (k) Safety/ security considerations

2.5 Assumptions and dependencies

List each of the factors that affect the requirements stated in the SRS. These are not design constraints on the software but are any changes to them that can affect the requirements in the SRS. ie assume specific OS on certain hardware; may have to change SRS if specific OS is not available.

2.6 Apportioning of requirements

Identify requirements that may be delayed until future versions of the system.

3 Specific requirements

Contains all software requirements to a level of detail sufficient to enable the designers to design a system to satisfy those requirements and testers to test that the system satisfies those requirements. Every stated requirement should be externally perceivable by users, operators or other external systems. At minimum, a description of each input (stimulus) into the system, every output (response) from the system and all functions performed by the system in response to an input or in support of an output.

- (a) Specific requirements should be stated in conformance with all characteristics described in 4.3 of recommended practices.
- (b) Specific requirements should be cross-referenced with earlier documents that relate.
- (c) All requirements should be uniquely identifiable.
- (d) Careful attention given to organization of requirements for readability.

3.1 External interfaces

Detail description of all inputs into and all outputs from the system. It should complement the interface description of 5.2 and should not repeat information. It should include both content and format as follows:

- (a) Name of item
- (b) Description of purpose
- (c) Valid range, accuracy and/or tolerance
- (d) Units of measure
- (e) Timing
- (f) Relationship to other inputs/outputs
- (g) Screen formats/organization
- (h) Data formats

- (i) Command formats
- (j) End messages

3.2 Functions

Functional requirements should define the fundamental actions that take place in the software accepting and processing the inputs and in processing and generating the outputs. "The system shall"....

- (a) Validity checks on the input
- (b) Exact sequence of operations
- (c) Responses to abnormal situations, including
 - (a) Overflow
 - (b) Communication facilities
 - (c) Error handling and recovery
- (d) Effects of parameters
- (e) Relationship of outputs to inputs, including
 - (a) Input/Output sequences
 - (b) Formulas for input to output conversion

3.3 Performance requirements

Specify both the static and dynamic numerical requirements placed on the software or on human interaction with the software. Static numerical requirements may include:

- (a) The number of terminals to be supported
- (b) The number of simultaneous users to be supported
- (c) Amount and type of information to be handled

Dynamic numerical requirements may include the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions. Stated in measurable terms.

3.4 Logical database requirements

Specify logical requirements for any information that is to be placed in a database. Includes:

- (a) Types of information used by various functions
- (b) Frequency of use
- (c) Accessing capabilities
- (d) Data entities and their relationships
- (e) Integrity constraints
- (f) Data retention requirements

3.5 Design constraints

Specify the requirements that can be imposed by other standards, hardware limitations, etc.

3.5.1 Standards compliance

Specify the requirements derived from existing standards or regulations. May include:

1. Report format
2. Data naming
3. Accounting procedures
4. Audit tracing

ie could specify the requirements for software to trace processing activity.

3.6 Software system attributes

There are a number of attributes of software that can serve as requirements. Specify required attributes so their achievement can be objectively verified.

3.6.1 Reliability

Specify factors required to establish the required reliability of the software system at time of delivery.

3.6.2 Availability

This should specify the factors required to guarantee a defined availability level for the entire system such as check-point, recovery, and restart.

3.6.3 Security

Specify factors that would protect software from accidental or malicious access, use, modification, destruction, or disclosure. Could include:

- (a) Utilize cryptographical techniques
- (b) Keep log or history data sets
- (c) Assign certain functions to different modules
- (d) Restrict communications between some areas of the program
- (e) Check data integrity for critical variables

3.6.4 Maintainability

Specify attributes of software that relate to ease of maintenance of the software itself. May be requirements for modularity, interfaces, complexity, etc. Requirements should not be placed here because they are thought to be good design practice.

3.6.5 Portability

Specify attributes of software that relate to ease of porting to other host machines/OS.

- (a) Percentage of component with host dependent code
- (b) Percentage of code that is host dependent
- (c) Use of proven portable language
- (d) Use of particular compiler or language subset
- (e) Use of particular OS

3.7 Organizing the specific requirements