Software Engineering

Goal: Systematic and cost effective techniques for software development

The Science and Art

- Past experience is used:
    - Theoretical basis has been provided wherever possible, otherwise experience is used
    - (On the contrary, scientific solutions are constructed through rigorous application of provable principles)
- Several alternate solutions are tried. Iteration and backtracking are used. (Science, on the other hand uses only unique solutions)
-

Exploratory Software development: Programmer uses his intuition, experience etc.

Software Engg.

Principles Deployed -> Abstraction (E.g. Modeling)
-> Decomposition (E.g. Divide and conquer)

Software requirements specification (SRS): **(Ref. IEEE 830-1998)**

TITLE PAGE

Software requirements specification
<name of project>
<author>
<date>

Version	Release Date	Responsible Party	Major changes

Table of Contents

SECTION	DESCRIPTION	PAGE
1	Introduction	2
2	General Description	3
3	Specific Requirements	6
4	Supporting information	13

.

Introduction

1.1. Purpose: Identifying the purpose of this SRS and its intended audience

1.2. Scope:

- Identify the software product to be produced, by name
- Explain what the software product will, and if necessary, will not do
- Describe the application of the software

As part of this,

- Describe relevant benefits, objectives and goals as precisely as possible
- Be consistent with similar statements in higher level specification, if they exist

1.3. Definitions, acronyms and abbreviations:

- May be provided by reference to an appendix

1.4. References:

- Provides a complete list of all documents, referenced elsewhere in the SRS
- Identify each document by title, report no. date and publishing organization
- Specify the sources from where the reference can be obtained

1.5. Overview: Describes the rest of the SRS and how it is organized

2. General Description:

(Does not state specific requirements, that is to be done in section 3

2.1. Product perspective:

This subsection relates the product to other products or projects

If the SRS defines a product that is component of a larger system or project

i. Describe the function of each component of larger system or project, and identify interfaces
ii. Identify the principal external interfaces of the software product
iii. Describe the complete hardware and peripheral equipment to be used (Overview only)
iv. [Block diagram is helpful]

2.2. Product function: Provide a summary of the functions that the software will perform [List of functions should be understandable to the customer reading the document for the first time]

2.3. User Characteristics: Describe the general characteristics of the end-users of the product that will affect the specific requirements (Eg. education, experience, technical expertise)

2.4. General Constraints

Example: Regulatory policies
Hardware limitation (e.g., Signal timing requirement)
Interface to other applications
Parallel operation
Audit function
Criticality of the application

2.5. Assumptions and dependencies

E.g. A specific operating system

3. Specific Requirements:

(LARGEST AND MOST IMPORTANT PART)

3.1. Functional Requirements: For each function, specify requirements on input, processing and output

- Purpose of the function
- Inputs: Sources, valid ranges of values, any timing concerns, operator requirement, special interface
- Operation to be performed: validity check, response to abnormal condition, types of processing required, etc
- Outputs: Destination, valid range of values, timing concerns, handling of illegal values, error messages, interfaces required.

3.2. External interface requirements:

3.2.1. User Interfaces

a. Required screen formats
b. page layout and content of any report or menus
c. relative timing of inputs and outputs
d. availability of some form of programmable function keys

3.2.2. Hardware interfaces

- Specify the logical characteristics of each interface between the software product and the hardware components of the system
- Include such matters as what devices are to be supported, how they are to be supported and protocols
- A block diagram showing the relationship among the hardware blocks and the software function hosted in each block is essential here

3.2.3. Software interfaces:

- Specify the use of other required software product (for example, a data management system, an operating system, or a mathematical package) and interfaces with other application systems.
- For each required software product, the following should be provided
  1. Name
  2. Mnemonic
  3. Specification no
  4. Version no

5. Source

- For each interface:
  1. Discuss the purpose of the interfacing software as related to this software product
  2. Define the interface in terms of message content and format
     (Reference to the document defining the interface is required)

3.2.4. Communication interfaces: Specify the various interfaces to communication, such as local network protocols

3.3. <u>Performance Requirements</u>

3.3.1. <u>Static numerical requirements: may include:</u>

- The number of terminals to be supported
- The number of simultaneous users to be supported
- The number of files and records to be handled
- The sizes of the files and tables

3.3.2. <u>Dynamic numerical requirements</u>: may include, for example, the number of transactions and tasks and the amount of data to be processed within certain periods for both normal and peak workload conditions.
E.g., 95% of the transactions shall be processed in less than 1 second