

# SOFTWARE REQUIREMENTS SPECIFICATION

1

# Requirement Engineering

---

Requirements describe

What not How

Produces one large document written in natural language contains a description of what the system will do without describing how it will do it.

Crucial process steps

Quality of product  Process that creates it

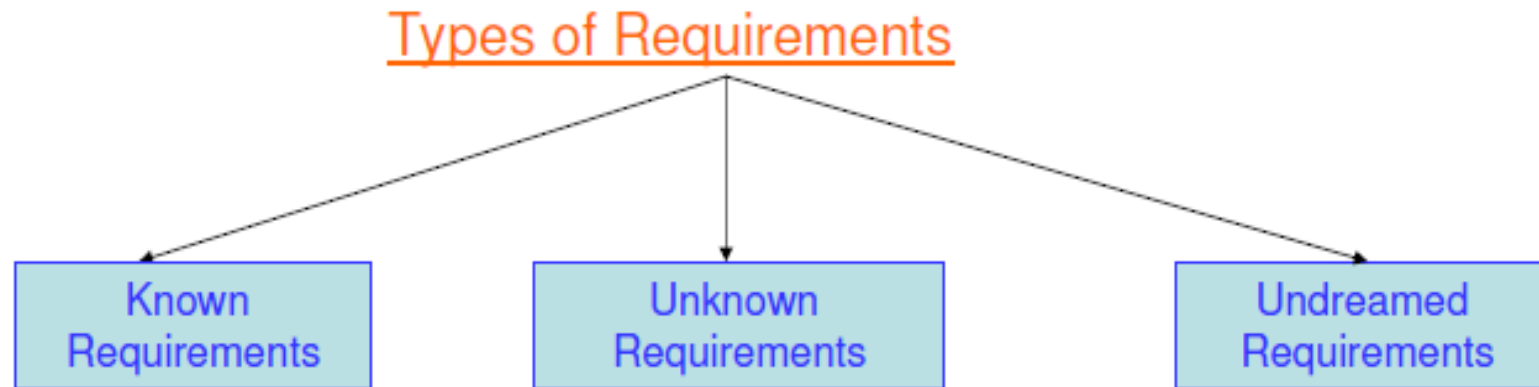
Without well written document

- Developers do not know what to build
- Customers do not know what to expect
- What to validate



# *Types of Requirements*

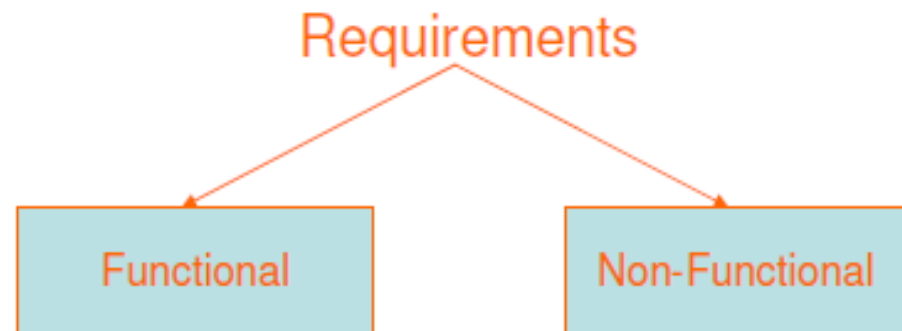
---



Stakeholder: Anyone who should have some direct or indirect influence on the system requirements.

--- User

--- Affected persons



# *Types of Requirements*

---

Functional requirements describe what the software has to do. They are often called product features.

Non Functional requirements are mostly quality requirements. That stipulate how well the software does, what it has to do.

Availability  
Reliability  
Usability  
Flexibility



For Users

Maintainability  
Portability  
Testability



For Developers



## *Feasibility Study*

---

### **Is cancellation of a project a bad news?**

As per IBM report, “31% projects get cancelled before they are completed, 53% over-run their cost estimates by an average of 189% & for every 100 projects, there are 94 restarts.

### **How do we cancel a project with the least work?**

**➡ CONDUCT A FEASIBILITY STUDY**



# FEASIBILITY REPORT TEMPLATE

1. Executive summary
2. Description of product/service
3. Technology considerations
4. Product/service marketplace
5. Marketing strategy
6. Organization/staffing
7. Schedule
8. Financial projections
9. Findings and recommendations




## *Requirements Documentation*

---

This is the way of representing requirements in a consistent format

SRS serves many purpose depending upon who is writing it.

- written by customer
  - written by developer
- 

Serves as contract between customer & developer.



# UNDERSTAND AND SPECIFYING REQUIREMENTS

- A problem of scale
  - **For small scale:** understand and specifying requirements is easy
  - **For large scale:** very hard; probably the hardest, most problematic and error prone
- The requirements task:
  - **Input:** User needs in minds of people (hopefully)
  - **Output:** precise statement of what the future system will do



# CHALLENGES

- Identifying and specifying requirements
  - Necessarily involves people interaction
  - Cannot be automated

# BACKGROUND..

- What is a Requirement?
  - A condition or capability that must be possessed by a system (IEEE)
- What is the work product of the Req. phase ?
  - A software requirements specification (SRS) document
- What is an SRS ?
  - A complete specification of what the proposed system should do!

# BACKGROUND..

- Requirements understanding is hard
  - Visualizing a future system is difficult
  - Capability of the future system not clear, hence needs not clear
  - Requirements change with time
  - ...
- Essential to do a proper analysis and specification of requirements

# PURPOSE OF SRS DOCUMENT?

- SRS establishes basis of agreement between the user and the supplier.
  - Users needs have to be satisfied, but user may not understand software
  - Developers will develop the system, but may not know about problem domain
- SRS is
  - the medium to bridge the communications gap, and
  - specifies user needs in a manner both can understand

## NEED FOR SRS...

- Helps user understand his needs.
  - users do not always know their needs
  - must analyze and understand the potential
  - The requirement process helps clarify needs
- SRS provides a reference for validation of the final product
  - Clear understanding about what is expected.
  - Validation - “ SW satisfies the SRS “

# NEED FOR SRS...

- High quality SRS essential for high Quality SW
  - Requirement errors get manifested in final sw
  - To satisfy the quality objective, must begin with high quality SRS
- Requirements defects cause later problems
  - 25% of all defects in one study; 54% of all defects found after user testing
  - defects often found in previously approved SRS.

# NEED FOR SRS...

- Good SRS reduces the development cost
  - SRS errors are expensive to fix later
  - Req. changes can cost a lot (up to 40%)
  - Good SRS can minimize changes and errors
  - Substantial savings; extra effort spent during req. saves multiple times that effort
- An Example
  - Cost of fixing errors in req. , design , coding , acceptance testing and operation are 2 , 5 , 15 , 50 , 150 person-months

# Requirements vs. Design

Requirements	Design
Describe <b>what</b> will be delivered	Describe <b>how</b> it will be done <sup>3</sup>
Primary goal of analysis: UNDERSTANDING	Primary goal of design: OPTIMIZATION
There is more than one solution	There is only one (final) solution
Customer interested	Customer not interested (Most of the time) except for external





## *Requirements Documentation*

---

SRS Should

- Correctly define all requirements
- not describe any design details
- not impose any additional constraints

Characteristics of a good SRS

An SRS Should be

- ✓ Correct
- ✓ Unambiguous
- ✓ Complete
- ✓ Consistent



## *Requirements Documentation*

---

- ✓ Ranked for important and/or stability
- ✓ Verifiable
- ✓ Modifiable
- ✓ Traceable



# *Requirements Documentation*

---

## Correct

An SRS is correct if and only if every requirement stated therein is one that the software shall meet.

## Unambiguous

An SRS is unambiguous if and only if, every requirement stated therein has only one interpretation.

## Complete

An SRS is complete if and only if, it includes the following elements

- (i) All significant requirements, whether related to functionality, performance, design constraints, attributes or external interfaces.



## *Requirements Documentation*

---

- (ii) Responses to both valid & invalid inputs.
- (iii) Full Label and references to all figures, tables and diagrams in the SRS and definition of all terms and units of measure.

### Consistent

An SRS is consistent if and only if, no subset of individual requirements described in it conflict.

### Ranked for importance and/or Stability

If an identifier is attached to every requirement to indicate either the importance or stability of that particular requirement.



# *Requirements Documentation*

---

## Verifiable

An SRS is verifiable, if and only if, every requirement stated therein is verifiable.

## Modifiable

An SRS is modifiable, if and only if, its structure and style are such that any changes to the requirements can be made easily, completely, and consistently while retaining structure and style.

## Traceable

An SRS is traceable, if the origin of each of the requirements is clear and if it facilitates the referencing of each requirement in future development or enhancement documentation.



# Organization of SRS

<b>1.</b>	
<b>Introduction</b>	Purpose
1.1	Scope
1.2	Definition, Acronyms and
1.3	abbreviations
1.4	References
1.5	Overview



# *Requirements Documentation*

---

## 2. The Overall Description

### 2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communication Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirements



## *Requirements Documentation*

---

- 2.2 Product Functions
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Assumptions for dependencies
- 2.6 Apportioning of requirements

### 3. Specific Requirements

- 3.1 External Interfaces
- 3.2 Functions
- 3.3 Performance requirements
- 3.4 Logical database requirements
- 3.5 Design Constraints
- 3.6 Software System attributes
- 3.7 Organization of specific requirements
- 3.8 Additional Comments.

### 4. Others Requirements/Information





# INTRODUCTION

- The Software Requirement Specification(SRS) document provides an overview of the entire SRS.

- **Purpose:**

- Identify the purpose of SRS and its intended audience.

- **Scope:**

- Identify the software products to be produced by name.
  - Explain what the software products will, and, if necessary, will not do.
  - Describe the application of the software being specified, including relevant benefits, objective and goals
  - Be consistent with similar statement in higher level specification if the exist.



# INTRODUCTION CONTD.....

- **Definition, Acronyms, and Abbreviations:**

- Provide definition of all terms, acronyms, and abbreviations required to properly interpret the SRS.
- This information may be provided by reference to an appendix.

- **References:**

- Provide complete list of all documents referenced elsewhere in the SRS.
- Identified each document by title, report number, date, and publishing organization.
- Specify the sources from which the reference can be obtained.



# INTRODUCTION CONTD.....

- **Overview:**

- Describe what the rest of the SRS contains.
- Explain how the SRS is organised.



# THE OVERALL DESCRIPTION

- Describe the general factors that affects the product and its requirements.
- **Product Perspective:**
  - Put the product into perspective with other related products.
  - If the product is independent and totally self-contained, it should be so stated here.
  - If the SRS defines a product that is a component of a larger system, as frequently occurs, then it relates the requirements of the larger system to functionality of the software and identifies interfaces between that system and the software.



# THE OVERALL DESCRIPTION

## CONTD.....

The following subsection describe how the software operates inside various constraints:

- **System Interface:**

- List each system interface and identify the functionality of the software to accomplish the system requirement.
- And the interface description to match the system.



# THE OVERALL DESCRIPTION

## CONTD.....

- **Interfaces:**

- It specifies the logical characteristics of each interface between the software product and its users.
- All the aspects of optimising the interface with the person who must use the system.

- **Hardware Interfaces:**

- Specify the logical characteristics of each interface between the software product and hardware component of the system.
- This also includes configuration characteristics.



# THE OVERALL DESCRIPTION CONTD.....

- **Software interfaces:**

- Specify the use of other software products and interface with other application system.
- It includes
  - Name
  - Specification number
  - Version number
  - Source



# THE OVERALL DESCRIPTION

## CONTD.....

- **Communication Interfaces:**
  - Specify the various interfaces to communications such as local network protocols, etc.
- **Memory constraints:**
  - Specify any applicable characteristics and limits on primary and secondary memory.





# THE OVERALL DESCRIPTION CONTD.....

- **Operations:**

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



# THE OVERALL DESCRIPTION

## CONTD.....

- **Site Adaption Requirements:**
  - Define the requirement for any data or initialization sequences that are specific to a given site, mission, operational mode.
  - Specify the site or mission related feature that should be modified to adapt the software to a particular installation.



# PRODUCT FUNCTION

- Provide a summary of the major functions that the software will perform.
  - The function should be organised in a way that makes the list of functions understandable to the customer or to any one else reading the document for the first time.
  - Textual or graphic methods can be used to show the different function and their relationship.



# USER CHARACTERISTICS

- Describe those general characteristic of the intended users of the product including educational level, experience, and technical expertise



# CONSTRAINTS

- Provide a general description of any other items that will limit the developer's options. These can include:
  - Regulatory Policies
  - Hardware Limitation
  - Interface to other Application
  - Parallel Operation
  - Audit Function
  - Control Function
  - Higher Order Language Requirements
  - Signal Handshake Protocols
  - Reliability Requirements
  - Criticality of the Application
  - Safety and Security Considerations



# ASSUMPTION AND DEPENDENCIES

- List each of the factors that affects the requirements stated in SRS
- These factors are not designed constraints on the software but, any change to them may that can affect requirements in SRS.



# APPORTIONING OF REQUIREMENTS

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



# SPECIFIC REQUIREMENTS

- Specific requirements should be stated with all the characteristics of the good SRS.
  - Correct
  - Unambiguous
  - Complete
  - Consistent
  - Ranked for importance and stability
  - Verifiable
  - Modifiable
  - Traceable
- Specific requirement should be cross-referenced to earlier documents that relate.
- All requirements should uniquely identifiable.
- Careful attention should be given to organising the requirement to maximize readability.





# EXTERNAL INTERFACE

- This contains a detailed description of all the input into and output from the software system.
- It contains both content and format as follow:
  - Name of item
  - Description of purpose
  - Source of input or destination of output
  - Valid range, accuracy and tolerance
  - Units of measure
  - Timing
  - Relationship to other Input/Outputs
  - Screen formats
  - Window formats
  - Data formats
  - Command formats
  - End message



# FUNCTIONS

- Functional requirements defines the fundamental action that must take place in software in accepting and processing the inputs and in processing and generating the outputs. These includes:
  - Validity checks on the inputs
  - Exact sequence of operations
  - Response to abnormal situation, including
    - Overflow
    - Communication facilities
    - Error handling and recovery
  - Effects of parameters
  - Relationship of output to inputs, including
    - Input/output Sequences
    - Formulas for input to output conversion.



# PERFORMANCE REQUIREMENTS

- This subsection specifies both the static and the dynamic numerical requirements placed on the software.
- Static numerical requirements may include:
  - The number of terminals to be supported.
  - The number of simultaneous users to be supported.
  - Amount and types of information to be handled.
- Dynamic numerical requirements may include:
  - Number of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.



# LOGICAL DATABASE REQUIREMENTS

- This section specifies the logical requirements for any information that is to be placed into a database. This may include:
  - Types of information used by various functions
  - Frequency of use
  - Accessing capabilities
  - Data entities and their relationship
  - Integrity constraints



# DESIGN CONSTRAINTS

- Specify design constraints that can be imposed by other standards, hardware limitations, etc.
- **Standard Compliance:**
  - Specify the requirements derived from existing standards or regulations.
  - Report format
  - Data naming
  - Accounting procedure
  - Audit tracing



# SOFTWARE SYSTEM ATTRIBUTES

- There are a number of quality attribute of software that can serve as requirements.
- **Reliability:**
  - Specify the factor required to establish the required reliability of the software system at time of delivery.
- **Availability:**
  - Specify the factors required to guarantee a defined availability level for the entire system such as check point, recovery and restart.



# SOFTWARE SYSTEM ATTRIBUTES CONTD.....

- **Security:**

- Specify the factor that would protect the software from accidental or malicious access, use, modification, destruction. This may include need to:
  - Utilizing certain cryptographic techniques
  - Keep specific log or history data sets
  - Assign certain functions to different modules
  - Restrict communication between some areas of the program
  - Check data integrity for critical variable



# SOFTWARE SYSTEM ATTRIBUTES

## CONTD.....

- **Maintainability:**

- Specify attribute of software that relate to the ease of maintenance of software itself.
- There may be some requirement for certain modularity, interface, complexity, etc.

- **Portability:**

- Specify attributes of software that relates to the ease of parting the software to other host machine and operating systems. This may include:
  - Percentage of component with host dependent code
  - Percentage of code that is host dependent
  - Use of a proven portable language
  - Use of a particular compiler
  - Use of a particular operating system.





# ORGANISING THE SPECIFIC REQUIREMENTS

- **System Model:**

- Some system behave quite differently depending on the mode of operation. There are two possible outlines.
- The choice depends on whether interfaces,
- Performance are dependent on mode.

- **User Class:**

- Some system provide different sets of functions to different class of users.

- **Objects:**

- Object are real world entities that have a counterpart within the system.
- Associated with each object is a set of attributes and functions.
- These functions are called services, methods, or processes.



# ORGANISING THE SPECIFIC REQUIREMENTS

## CONTD...

- **Feature:**

- A feature is an externally desired service by the system that may require a sequence of inputs to effect the desired result.
- Each feature is described as sequence of stimulus-response pairs.

- **Stimulus:**

- Some system can be best organised by describing their functions in terms of stimuli.

- **Response:**

- Some system can be best organised by describing their function in support of the generation of the response.



# **CHANGE MANAGEMENT PROCESS:**

- Specify the change management process to be used to identify, log, evaluate, and update the SRS to reflect changes in project scope and requirement.



# DOCUMENT APPROVAL

- Identify the approvers of the SRS documents. Approver's name, signature, and date should be used.



# SUPPORTING INFORMATION

- The supporting information makes the SRS easier to use. It includes:
  - Table of contents
  - Index
  - Appendices



# SRS

- **1.0 INTRODUCTION**

This document specifies all the requirements for

- **1.1 Purpose**

The purpose of the ...is to ....

The system should assist ....

The intended audience for this document is ...

This specification describes .....

- **1.2 Scope**

This document applies only to .....

This specification is not concerned with .....

# SRS

## ■ 1.3 Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specifications

IEEE - Institute of Electrical and Electronic Engineering

## ■ 1.4 Reference

[1] IEEE 830-1993: IEEE Recommended Practice for Software Requirements Specifications" IEEE Standards Collection, IEEE, 1997.

## ■ 1.5 Overview

In the following sections of this specification.....will be presented.

In Section 2, the general product and its functions will be introduced.

# SRS

In Section 3, all detailed requirements will be specified and grouped.

In Appendix .....

## **2.0 GENERAL DESCRIPTION**

### **2.1 Product Perspective**

This system allows stakeholders to.....

The system will display.....

The system will help .....

The system provides information about ....

### **2.2 Product Functions**

The system provides the following functions:



# SRS

## ▪ 2.3 User Characteristics

The users of the system are:

- ⑩ Level of Users' Computer Knowledge
- ⑩ Level of Users' Business Knowledge
- ⑩ Frequency of Use

## ▪ 2.4 General Constraints

The system will support ....

The system will not allow .....

## ▪ 2.5 Assumption and Dependencies

This system relies on .....

The system must have a satisfactory interface and .....

# SRS

- **SPECIFIC REQUIREMENTS**

- **3.1 Functional Requirements**

- **3.1.1 Unit Registration**

- The unit registration requirements are concerned with functions regarding unit registration which includes students selecting, adding, dropping, and changing a unit.
- **SRS-001 (3.1.1.1):**
  - The system shall allow the user to register a unit.
- **SRS-002 (3.1.1.2):**
  - System shall allow the user to delete a unit if the user has chosen to drop that unit.
- **SRS-003 (3.1.1.3):**
  - System shall check if a unit has been filled by enough registered students.

# SRS

- **SRS-004 (3.1.1.4):**
  - System shall allow the user to add his/her name to the unit waiting list if the user wants to register in a unit which has been filled already with enough registered students.
- **SRS-005 (3.1.1.5):**
  - System shall automatically register the unit for the user who is the first one on the waiting list if a vacancy appears for that unit.
- **SRS-006 (3.1.1.6):**
  - System shall allow the user to change practical session(s) within a unit.
- **SRS-007 (3.1.1.7):**
  - System shall allow the user to change tutorial session(s) within a unit.

# SRS

- **3.1.2 Retrieving and Displaying Unit Information**
- The retrieving and displaying requirements are concerned with how information is retrieved and presented to the user.
- **SRS-014 (3.1.2.1):**
- The system shall allow users to enter the following selection criteria to retrieve unit information: by unit code, by unit number, by title of unit, by weight of unit (credit points).
- **OR** by unit code (3.1.2.1.1) , by unit number (3.1.2.1.2) , by title of unit (3.1.2.1.3) , by weight of unit (credit points) (3.1.2.1.4).

# SRS

- **3.2 Design Constraints**
- **SRS-031 (3.2.1):**
  - System shall store and retrieve persistent data.
- **SRS-032 (3.2.2):**
  - System shall support PC and/or UNIX platforms.
- **SRS-033 (3.2.3):**
  - System shall be developed using the JAVA programming language

# SRS

- **3.3 Non-Functional Requirements**

- **SRS-034 (3.3.1):**

- System shall respond to any retrieval in less than 5 seconds.

- **SRS-035 (3.3.2):**

- System shall generate a report within 1 minute.

- **SRS-036 (3.3.3):**

- System shall allow the user to remotely connect to the system.

- **SRS-041 (3.3.8):**

- The system will be accompanied by a comprehensive user manual.

# SRS

## ■ 3.5.3 Security

- The security requirements are concerned with security and privacy issues.

### **SRS-029:**

- System shall provide staff ID and password verification protection to protect from unauthorised use of the system.

### **SRS-030:**

- System shall allow the store manager to add, remove and modify staff ID and passwords as required.