Software Requirement Specification

What is a requirement?

- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.
- This is inevitable as requirements may serve a dual function
  - May be the basis for a bid for a contract – therefore must be open to interpretation; (there must not be room left for different interpretation)
  - May be the basis for the contract itself – therefore must be defined in detail;

Functional and non-functional requirements

- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
  - For example, try to open a file but it does not exists so how the system reacts to such a situation

- Non-functional requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc. (its everything, ..catch-all bucket to put in requirements that don't fit into the functional structure of the system)

Functional requirements

- Describe functionality or system services.
- Depend on the type of software, expected users and the type of system where the software is used.
- Functional user requirements may be high-level statements of what the system should do but functional system requirements should describe the system services in detail.
- The LIBSYS system
  - A library system that provides a single interface to a number of databases of articles in different libraries.
  - Users can search for, download and print these articles for personal study

Examples of functional requirements

- The user shall be able to search either all of the initial set of databases or select a subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store. (pdf, doc or post script format)
- Every order shall be allocated a unique identifier (ORDER_ID) which the user shall be able to copy to the accounts permanent storage area.

Requirements imprecision

- Problems arise when requirements are not precisely stated.
- Ambiguous requirements may be interpreted in different ways by developers and users.
- Consider the term ‘appropriate viewers’
  - User intention – special purpose viewer for each different document type;
  - Developer interpretation – Provide a text viewer that shows the contents of the documents

Requirements completeness and consistency

- In principle, requirements should be both complete and consistent.
- Complete
  - They should include descriptions of all facilities required. (specifying the requirements that what the systems exactly do and it should also specify along the same line what the system is not to do)
- Consistent
  - There should be no conflicts or contradictions in the descriptions of the system facilities.
- In practice, it is impossible to produce a complete and consistent requirements document.

Non-functional requirements

- These define system properties and constraints, e.g., reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.

- Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless.

  Example: Trading System – Real Time Systems

Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g., execution speed, reliability, etc.
- Organizational requirements
  - Requirements that are a consequence of organizational policies and procedures e.g. process standards used, implementation requirements, etc. (ISO 9001)
- External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements (confirm to certain standard guidelines and rules, example, Military applications) etc. (quality guidelines)

Non-functional requirements examples

- Product requirement

  1.1 The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.

- Organizational requirement

  2.1 The system development process and deliverable documents shall conform to the process and deliverables defined in ISO 9001.

- External requirement

  3.1 The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.

Guidelines for writing requirements

- Invent a standard format and use it for all requirements
- Use language in a consistent way. Use shall for mandatory requirements, should for desirable requirements.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon. (user may not understand certain computer terminology like frame or plain text format in webpage)

Ambiguity

“When temperature becomes high, start cooler”

Do you notice any problems?

- Above what threshold we consider the temperature to be high?

Inconsistent Requirement

- Some part of the requirement:
  - contradicts some other requirement.
- **Example:**
  - One customer says turn off heater and open water shower when temperature > 100°C
  - Another customer says turn off heater and turn ON cooler when temperature > 100°C

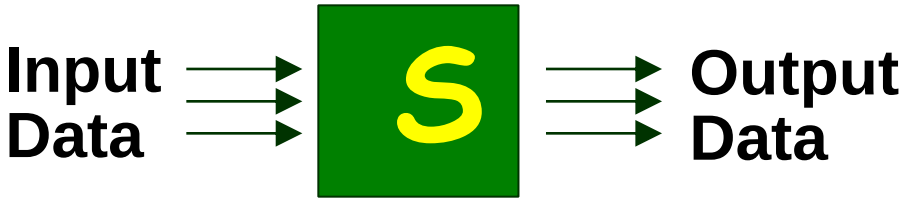
Incomplete Requirement

- Some requirements not included:
  - Possibly due to oversight.
- **Example:**
  - The analyst has not recorded that when temperature falls below 90° C :
    - heater should be turned ON
    - water shower turned OFF.

SRS Document

- As already pointed out--- useful in various contexts:
  - Statement of user needs
  - Contract document
  - Reference document
  - Definition for implementation

SRS Document (CONT.)



- SRS document is known as black-box specification:
  - The system is considered as a black box whose internal details are not known.
  - Only its visible external (i.e., input/output) behaviour is documented.

SRS Document (CONT.)

- SRS document concentrates on:
  - What needs to be done in terms of input-output behaviour
  - Carefully avoids the solution (“how to do”) aspects.

SRS Document (CONT.)

- The requirements at this stage:
  - Written using end-user terminology.
- If necessary:
  - Later a formal requirement specification may be developed from it.

SRS Document (CONT.)

- SRS is basis for subsequent design an implementation
- First and most important baseline
  - Defines contract with users
  - Basis for validation and acceptance
- Cost increases rapidly after this step; defects not captured here become 2 to 25 times more costly to remove later (defects which entered in the design and implementation which will be more costlier to remove it)

SRS Document (CONT.)

- It identifies all functional (inputs, outputs, processing) and performance requirements, and also other important constraints (legal, social, operational)
- Should be adequately detailed so that
  - Users can visualize what they will get
  - Design and implementation can be carried out

SRS Document (CONT.)

- Covers what and what at business level; e.g.,
  - What calculate take-home pay
  - How: procedure (allowances, deductions, taxes etc.)

What are the Uses of an SRS Document?

- Establishes the basis for agreement between the customers and the suppliers
- Forms the starting point for development.
- Provide a basis for estimating costs and schedules.
- Provide a basis for validation and verification.
- Provide a basis for user manual preparation.
- Serves as a basis for later enhancements.

Forms A Basis for User Manual

- The SRS serves as the basis for writing User Manual for the software:
  - **User Manual: Describes the functionality from the perspective of a user --- An important document for users.**
  - Typically also describes how to carry out the required tasks with examples.

SRS Document: Stakeholders

- Different levels of detail and formality is needed for each audience
- Different templates for requirements specifications used by companies:
  - Often variations of **IEEE 830**

Software Requirement Specification Format Document

- Baseline for the development, it's a contract document
- Based on IEEE Recommendation
- 1. **INTRODUCTION**
  - 1.1 **PURPOSE**:
    - clearly state purpose of this document (and what is covered in this document)
    - by whom and how it will be used for what purpose
  - 1.2 **SCOPE**: Overall context within which the software is being developed. What parts need to be automated.
  - 1.3 **Definitions**: Acronyms, Abbreviations as applicable
  - 1.4 **REFERENCES**: to other documents
  - 1.5 **Overview of Developer's Responsibilities**: In terms of development, installation, training, maintenance, etc.

Software Requirement Specification Format Document

- **2. GENERAL DESCRIPTION**

  2.1 **PRODUCT PERSPECTIVE**:

  - State whether it is a replacement to the existing system or it is a new software
  - relationship with other products and principle interfaces

  2.2 **PRODUCT FUNCTIONS OVERVIEW**: general overview of tasks; including data flow diagrams

  2.3 **USER CHARACTERISTICS**: who they are and what training they may need

  2.4 **GENERAL CONSTRAINTS:** about schedule, resources, cost, etc.

Software Requirement Specification Format Document

3. FUNCTIONAL REQUIREMENT

3.1 **INTRODUCTION**

3.2 **INPUTS**

3.3 **PROCESSING**

3.4 **OUTPUTS**

We give functional details, we define every function (ex: cancellation of a ticket function) by giving a brief introduction to this function, inputs, processing and outputs.

It is really the body of SRS Document.

3.5 …..(repeat similarly for each function)

Software Requirement Specification Format Document

4. External Interface Requirements

4.1 **User Interfaces**: a preliminary user manual giving commands, screen formats, outputs, error messages, etc. (logical contents of these components not the layout of the screen, output)

4.2 **Hardware Interfaces**: with existing as well as new or special purpose hardware, supported device type

4.3 **Software Interfaces**: with other software packages, operating systems, etc. (railway reservation system may be interfacing with the accounting packages so that all fund transfer may be handled)

Software Requirement Specification Format Document

5. Performance Requirements

- Capacity requirements (no of users, no of files(volume of data)), response time, throughput (in measurable terms)

  Ex: Bank Transactions – how long it takes, how many transactions will be possible over a given period
- Safety requirements – recovery after software failure, any damage or loss, handling of software and hardware failure
- Security and Privacy requirements

Software Requirement Specification Format Document

6. Design Constraints

6.1 **Standards Compliance**: software development standards as well as organizational standards (e.g., for reports- regulatory needs, auditing requirements)

6.2 **Hardware Limitations**: available machines, operating systems, storage capacities, etc.

Software Requirement Specification Format Document

7. Other Requirements

Possible future extensions

Note: All sections are not required for all projects.

- It has taken into account various aspects of the software. We can handover this SRS document to the development/design team. Then they can convert this specification into a design.

Software Requirement Specification Format Document

- SRS document needs to detailed and ensure we have collected all required data put it in the form of a document

- There should be a formal review meeting with the users and users should sign off that SRS document clearly defined what the software system needs to do.

- It is also ensured that the document contains enough design details required.

EXAMPLE 1: Withdraw cash from ATM

R.1: Withdraw cash

Description: The withdraw-cash function first determines the type of account that the user has and the account number from which the user wishes to withdraw cash. It checks the balance to determine whether the requested amount is available in the account. If enough balance is available, it outputs the required cash, otherwise it generates an error message.

EXAMPLE 1: Withdraw cash from ATM

R.1.1: Select withdraw amount option

Input: “Withdraw amount” option selected

Output: User prompted to enter the account type

R.1.2: Select account type

Input: User selects option from any one of the following—savings/checking/deposit.

Output: Prompt to enter amount

EXAMPLE 1: Withdraw cash from ATM

R.1.3: Get required amount

Input: Amount to be withdrawn in integer values greater than 100 and less than 10,000 in multiples of 100.

Output: The requested cash and printed transaction statement. Processing: The amount is debited from the user's account if sufficient balance is available, otherwise an error message displayed.

EXAMPLE 2 Search book availability in library

R.1: Search book

Description Once the user selects the search option, he would be asked to enter the keywords. The system would search the book in the book list based on the key words entered. After making the search, the system should output the details of all books whose title or author name match any of the key words entered. The book details to be displayed include: title, author name, publisher name, year of publication, ISBN number, catalog number, and the location in the library.

EXAMPLE 2 Search book availability in library

R.1.1: Select search option

Input: “Search” option

Output: User prompted to enter the key words

EXAMPLE 2 Search book availability in library

R.1.2: Search and display

Input: Key words

Output: Details of all books whose title or author name matches any of the key words entered by the user. The book details displayed would include—title of the book, author name, ISBN number, catalog number, year of publication, number of copies available, and the location in the library.

Processing: Search the book list based on the key words

EXAMPLE 2 Search book availability in library

R.2: Renew book

Description: When the “renew” option is selected, the user is asked to enter his membership number and password. After password validation, the list of the books borrowed by him are displayed. The user can renew any of his borrowed books by indicating them. A requested book cannot be renewed if it is reserved by another user. In this case, an error message would be displayed.

EXAMPLE 2 Search book availability in library

R.2.1: Select renew option

State: The user has logged in and the main menu has been displayed.

Input: “Renew” option selection.

Output: Prompt message to the user to enter his membership number and password

EXAMPLE 2 Search book availability in library

R.2.2: Login

State: The renew option has been selected.

Input: Membership number and password.

Output: List of the books borrowed by the user is displayed, and user is prompted to select the books to be renewed, if the password is valid. If the password is invalid, the user is asked to re-enter the password.

Processing: Password validation, search the books issued to the user from the borrower's list and display.

Next function: R.2.3 if password is valid and R.2.2 if password is invalid.

EXAMPLE 2 Search book availability in library

R.2.3: Renew selected books

Input: User choice for books to be renewed out of the books borrowed by him.

Output: Confirmation of the books successfully renewed and apology message for the books that could not be renewed.

Processing: Check if anyone has reserved any of the requested books. Renew the books selected by the user in the borrower's list, if no one has reserved those books.