Requirement Analysis

- Objective: determine what the system must do to solve the problem (without describing how)
- Done by Analyst (also called Requirements Analyst)
- Produce Software Requirement Specifications (SRS) document
- Incorrect, incomplete, inconsistent, ambiguous SRS often cause for project failures and disputes
- A very challenging task
  - -Users may not know exactly what is needed or how computers can bring further value to what is being done today
  - -Users change their mind over time
  - -They may have conflicting demands
  - -They can’t differentiate between what is possible and cost-effective against that is impractical (wish-list)
  - -Analyst has no or limited domain knowledge
  - -Often client is different from the users

SRS

- 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)
- It identifies all functional (inputs, outputs, processing) and performance requirements, and also other important constraints (legal, social, operational)

SRS ...

- Should be adequately detailed so that

  -Users can visualize what they will get

  -Design and implementation can be carried out
- Covers what and what at business level; e.g.,

  - What calculate take-home pay

  - How: procedure (allowances, deductions, taxes etc.)

Requirement Specification Format Document

- Baseline for the development, it’s a contract between the user and developed
- Based on IEEE Recommendation, IEEE 830 standard

Requirement Specification Format Document

- 1. **Introduction**

  1.1 **PURPOSE:** clearly state purpose of this document (and what is covered in this document). It should describe where the software would be deployed and how the software would be used.

  1.2 **SCOPE:** by whom and how it will be used for what purpose. Describe the overall context within which the software is being developed.

  Example: what parts of the problem are being automated and what parts of it would need to be automated in future.

  1.3 **Environmental characteristics:** Briefly outline the environment (hardware and other software) with which the software will interact.

  1.4 **Definitions:** Acronyms, Abbreviations as applicable

  1.5 **REFERENCES:** to other documents

  1.6 **Overview of Developer's Responsibilities:** In terms of development, installation, training, maintenance, etc.

Requirement Specification Format

2. **GENERAL DESCRIPTION**

2.1 **PRODUCT PERSPECTIVE**: relationship with other products and principle interfaces. Briefly state as to whether the software is a replacement for a certain existing systems, or it is a new software. It would be used as a component of a larger system, a simple schematic diagram containing the components of overall system

2.2 **PRODUCT FUNCTIONS OVERVIEW**: general overview of tasks; major ways in which the software would be used; including data flow diagrams

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

2.4 **Operating Environment:** hardware platform in which the software would run, operating system and other applications with which the software would interact.

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

Requirement Specification Format

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)

Requirement Specification Format

4. **External Interface Requirement**s

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

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)

4.4 Communication Interfaces: type of communications required by the software, such as e-mail, web access, network server communication protocols etc.

Any communication standards that will be used, such as TCP Sockets, FTP, HTTP etc. Communication security or encryption issues.

Requirement Specification Format

5. **Performance Requirements**

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

Ex: Bank Transactions – how long it takes, how many transactions will be possible over a given period

6. **Design and Implementation 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: Memory requirements, available machines, operating systems, database to be used, storage capacities, etc.

6.3 Specific programming language to be used, security considerations, programming standards etc.

Requirement Specification Format

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.
- 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.