Requirement Analysis and Specification

Software Requirements Document/Specification

The software requirements document (sometimes called the software requirements specification or SRS) is an official statement of what the system developers should implement. Requirements documents are essential when an outside contractor is developing the software system.

Who use SRS?

1. System Customers : Specify the requirements and read them to check that they meet their needs. Customers specify changes to the requirements.

2. Managers : Use the requirements document to plan a bid for the system and to plan the system development process.

3. System Engineers : Use the requirements to understand what system is to be developed.

4. System Test Engineers: Use the requirements to develop validation tests for the system.

5. System Maintenance Engineers: Use the requirements to understand the system and the relationships between its parts.

What includes in SRS?

Below table shows one possible organisation for a requirements document that is based on an IEEE standard for requirements documents (IEEE, 1998).

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organisation commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The non-functional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.

System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This should describe the functional and non-functional requirements in more detail. If necessary, further detail may also be added to the non-functional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components, the system, and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organisation of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

Requirements Specification

Requirements specification is the process of writing down the user and system requirements in a requirements document.

User requirements are almost always written in natural language supplemented by appropriate diagrams and tables in the requirements document. System requirements may also be written in natural language but other notations based on forms, graphical system models, or mathematical system models can also be used.

Example :- Natural Language Sentences

Priority Number	1
Function Name	Create Account
Description	Initially user should create account in the system
Input	Enter User's personal Data
Process	Application asks name, phone number, gender birthday from the user
Output	Create a new account by saving the data necessarily

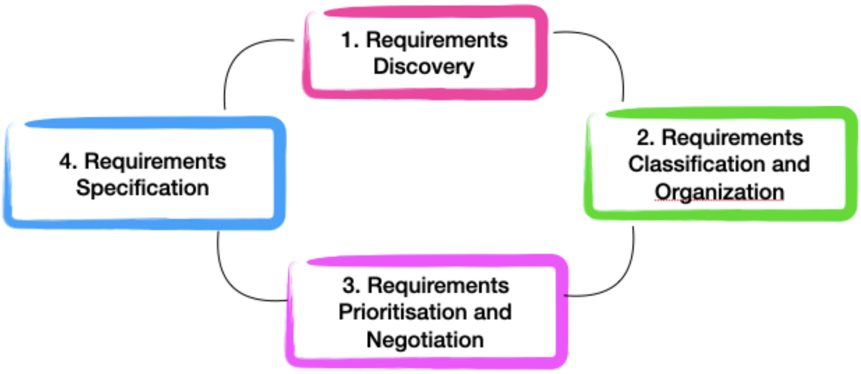
Requirements Engineering Process

requirements engineering processes may include four high-level activities.

1. Assessing if the system is useful to the business (feasibility study)
2. Discovering requirements (elicitation and analysis)
3. Converting these requirements into some standard form (specification)
4. Checking that the requirements actually define the system that the customer wants (validation).

Requirements Elicitation and Analysis

After an initial feasibility study, the next stage of the requirements engineering process is requirements elicitation and analysis. In this activity, software engineers work with customers and system end-users to find out about the application domain, what services the system should provide, the required performance of the system, hardware constraints, and so on.


1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritisation and Negotiation
4. Requirements Specification

The process activities are:

1. Requirements discovery - This is the process of interacting with stakeholders of the system to discover their requirements.

2. Requirements classification and organisation - This activity takes the unstructured collection of requirements, groups related requirements, and organises them into coherent clusters.

3. Requirements prioritisation and negotiation - Inevitably, when multiple stake- holders are involved, requirements will conflict. This activity is concerned with prioritising requirements and finding and resolving requirements conflicts through negotiation.

4. Requirements specification - The requirements are documented and input into the next round of the spiral. Formal or informal requirements documents may be produced, as discussed in lecture 04

Requirements Validation

Requirements validation is the process of checking that requirements actually define the system that the customer really wants. It overlaps with analysis as it is concerned with finding problems with the requirements. Requirements validation is important because errors in a requirements document can lead to extensive rework costs when these problems are discovered during development or after the system is in service.

The cost of fixing a requirements problem by making a system change is usually much greater than repairing design or coding errors. The reason for this is that a change to the requirements usually means that the system design and implementation must also be changed. Furthermore the system must then be re-tested.

During the requirements validation process, different types of checks should be carried out on the requirements in the requirements document.