Requirement engineering is the disciplined application of proven principles, methods, tools, and notation to describe a proposed system's intended behavior.

It is a four-step process, which includes - Feasibility Study, Requirement Elicitation and Analysis, Software Requirement Specification and Software Requirements Validation.

Feasibility Study: The objective behind the feasibility study is to create the reasons for developing the software.

Technical Feasibility: The technical feasibility evaluates the current technologies.

Operational feasibility assesses the range in which the required software performs a series of levels to solve business problems and customer requirements.

Economic feasibility: Economic feasibility decides whether the necessary software can generate financial profits for an organization.

Software requirement specification is a kind of document which is created by a software analyst.

The requirements are analyzed to identify inconsistencies, defects, omission, etc.

We describe requirements in terms of relationships and resolve conflicts if any.

Data Flow Diagrams (DFDs) are used widely for modeling requirements.

DFD shows the flow of data through a system.

Data Dictionaries are simply repositories to store information about all data items defined in DFDs.

Data dictionary should at least define customer data items, to ensure that the customer and developers use the same definition and terminologies.

Another tool for requirement specification is theentity-relationship diagram, often called an "E-R diagram" Requirement management is the process of managing changing requirements during the software engineering process.

Largely software requirements must be categorized into two categories: 1.

Software requirements are the basis of the entire software development project.

functional requirements define a function that a system or system element must be qualified to perform and must be documented in differentforms.

Non-functional requirements are divided into two main categories: Execution qualities like security

and usability, and Evolution qualities like testability, maintainability, extensibility.

SRS is a specification for a specific software product, program, or set of applications.

It serves several goals depending on who is writing it.

SRS is used to define the needs and expectation of the users.

A good SRS is said to be perfect if it covers all the needs that are truly expected from the system.

The second case is written for various purposes and serves as a contract document between customer and developer.

There are three types of possible conflict in the SRS.

The specified characteristics of real-world objects may conflicts.

The format of an output report may be described in one requirement as tabular but in another as textual.

There may be a reasonable or temporal conflict between the two specified actions.

SRS is unambiguous when every fixed requirement has only oneinterpretation.

SRS is ranked for importance and stability if each requirement has an identifier to indicate either the significance or stability of that particular requirement.

SRS is correct when the specified requirements can be verified with a cost-effective system to check whether the final software meets those requirements.

SRS is traceable if the origin of each of the requirements is clear and can be referenced in future development or enhancement.

The forward traceability of the SRS is especially crucial when the software product enters the operation and maintenance phase.

An SRS should be written in such a method that it is simple to generate test cases and test plans from the report.

The SRS report should be concise and at the same time, unambiguous, consistent, and complete.

Verbose and irrelevant descriptions decrease readability and increase error possibilities.

A well-structured document is simple to understand and modify.

The SRS report is also known as the black-box specification of a system.

It should only define what the system should do and refrain from stating how to do these.

Requirement analysis is significant and essential activity after elicitation.

This activity reviews all requirements and may provide a graphical view of the entire system.

The various steps of requirement analysis are shown in fig.

Prototype helps the client to visualize the proposed system andincrease the understanding of the requirements.

When developers and users are not sure about some of the elements, a prototype may help both the parties to take a final decision.

Prototype should be built quickly and at a relatively low cost.

It will always havelimitations and would not be acceptable in the final system..