

Software Requirement

According to IEEE standard 729, a requirement is defined as follows:

1. *A condition or capability needed by a user to solve a problem or achieve an objective*
2. *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents*
3. *A documented representation of a condition or capability as in 1 and 2*

A software requirement can be of three types:

1. Functional Requirement
2. Non-Functional Requirement
3. Domain Requirement

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract.

For example, in a class attendance management system, the teacher should be able to mark attendance, or view attendance of each individual student, etc. In another example, a patient record management system, the doctor should be able to search, view, edit and update patient's records.

Non-functional Requirements: These are basically the **quality constraints** that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

- **Portability**
- **Security**
- **Maintainability**
- **Reliability**
- **Scalability**
- **Performance**
- **Reusability**
- **Flexibility**

NFR's are classified into following types:

- Interface constraints
- Performance constraints: response time, security, storage space, etc.
- Operating constraints
- Life cycle constraints: maintainability, portability, etc.
- Economic constraints

Domain Requirements: Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain **must necessarily exhibit** come under this category.

For instance, in an academic software that maintains records of a school or college, the functionality of being able to access previous sessions and students enrolled or teachers teaching etc. are domain requirement. These requirements are therefore identified from that domain model and are not user specific

Requirement Engineering Lifecycle

Requirement Engineering is the process of defining, documenting and maintaining the requirements. It is a process of gathering and defining service provided by the system. Requirements Engineering Process consists of the following main activities:

1. Requirement Inception (Gathering of Requirements)
 - Walk-Throughs
 - Brain Storming
 - Model Analysis
 - Check Lists, CRC, etc
2. Requirements Elicitation
 - Scope
 - Understanding
 - Volatility
 - Incorrect Requirements
3. Requirement Elaboration
 - Prototyping
4. Requirement Negotiation
5. Requirements Specification
6. Requirements Verification and Validation
7. Requirements Management
8. Requirement Finalization (Contract Signing)

Requirements Inception:

A basic understanding of the problem is gained and the nature of the solution is addressed. Ask yourself questions that improve understanding of problem.

- Understanding of the problem.
- The people who want a solution.
- Nature of the solution.
- Communication and collaboration between the customer and developer.

Requirement Elicitation:

Elicitation means choosing or finding the right requirements.

There is **Scope issues**; what feature should be included in work and which should not, **Understanding issues**; there could be a difference of understanding between the developer and the customers, **Volatility**; Change is inevitable in any set of requirements,

which requirement is going to change in near future and which requirement will not change, **Incorrect Requirements**; the ability to select the right requirements for the project is important, leaving out faulty, incomplete, vague requirement is the main purpose of this task.

Requirement Elaboration:

Clarify any ambiguity in this phase. Explain requirement more with the help of experience, domain knowledge, business practices, standards, etc.

Requirement Negotiations:

- **Resolve conflicts**
- **Priorities work tasks**
- **Identify risks**
- **Estimations of resources**

With repeated discussions (inside organization & with customer) figure out what they want and how we will be able to deliver. What is required from customer and what we need for our team of developers?

- Availability of Resources.
- Delivery Time.
- Scope of requirements.
- Project Cost.
- Estimations on development

Requirement Specifications:

- Written document.
- A set of models.
- A collection of use cases.
- A prototype.

In the specification phase, the requirements engineer gathers all the requirements and develops a working model. This final working model will be the basis of any functions, features or constraints to be observed. The models used in this phase include

- **ER (Entity Relationship) diagrams,**
- **DFD (Data Flow Diagram),**
- **FDD (Function Decomposition Diagrams),**
- **Data Dictionaries.**

A software specification document is submitted to the customer in a language that he/she will understand, to give a glimpse of the working model.

Requirement Validation:

Focus on checking for errors and debugging. In validation phase, the developer scans the specification document and checks for the following:

- All the requirements have been stated and met correctly
- Errors have been debugged and corrected.
- Work product is built according to the standards.

Also known as the Technical Review of Requirements in some literature. Some of the validation techniques are the following-

- Requirements reviews/inspections.
- Prototyping.
- Test-case generation
- Automated consistency analysis

Requirement Management:

In this phase, the team is responsible for managing any changes that may occur during the project. New requirements emerge, and it is in this phase, responsibility should be taken to manage and prioritize as to where its position is in the project and how this new change will affect the overall system, and how to address and deal with the change. Based on this phase, the working model will be analyzed carefully and ready to be delivered to the customer

Requirement Finalization:

Contact signing, legal documentation and other specifications are listed in this phase. Both customer and the developing team representatives are responsible for preparation of documents and registering with authorities.