

Software Requirements

- Requirements are descriptions of the services that a software system must provide and the constraints under which it must operate.
- Requirements can range from high-level abstract statements of services or system constraints to detailed mathematical functional specifications.
- Requirements engineering is the process of establishing the services that the customer requires from the system and the constraints under which it is to be developed and operated.

Software Requirements Characteristics

- Clear
- Correct
- Consistent
- Coherent
- Comprehensible
- Modifiable
- Verifiable
- Prioritized
- Unambiguous
- Traceable
- Credible source

Different Types of Software Requirements

I. 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. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Ex- Student management system's functional requirements as follows,

1. The system will help the organizations to conduct their student registration
2. Only authorized person can access related details.
3. Organizations can change their information regarding themselves.
4. The students can login through STUDENT-ID and PASSWORD.

II. Non-Functional requirements

It describes how the system works, while functional requirements describe what the system should do. Also this known as system's quality and which define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

Example: - Modified data in a database should be updated for all users accessing it within 2 seconds.

III. User requirements

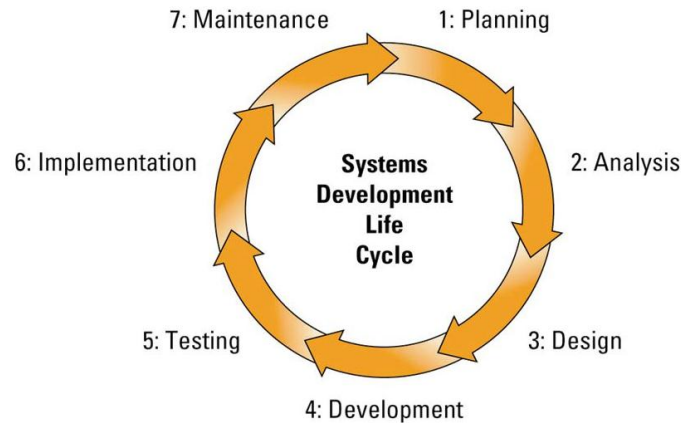
- It describes functional and non-functional requirements so that they are understandable by system users who don't have detailed technical knowledge.
- User requirements are defined using natural language, tables and diagrams

IV. Business requirements are usually captured by business analysts or product owners who analyze business activities in the proposed system. And it also emphasizes on the needs and expectations of the customer. It can indicate all the project deliverable and the inputs and outputs associated with each process function.

SDLC – Software Development Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the phases of a typical SDLC.



Phase – 1 Planning

- What do we want?” Define the system to be developed, the team defines the requirements of the new software and determines the cost and resources required.
- It also set the project scope, details the risks involved and provides sub-plans for reducing the risks. In this stage, a Software Requirement Specification document is created.

Phase – 2 Requirement Analysis

- Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for new software being built or modified.
- System study mainly focus
 - Study the organizational structure and problems and collect data from various source and stakeholders
 - Analyzing the current system, documenting, validating and managing and develop different possible solution
 - Selecting the best solution and finally make a *preliminary draft for feasibility study*.

Feasibility study:

- Feasibility is the determination (calculation) of whether a system or project can be performed successfully or not. The process followed in making this determination is called a feasibility study.
- It is the test of the proposed system in its workability, meeting user’s requirements, effective uses of resources and cost effectiveness with a help of SRS.

There are mainly five types of feasibilities checks:

- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/ compliances.
- **Operation feasibility:** Can we create operations which are expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

Phase – 3 Design

- Once the requirements are understood, software architects and developers can begin to design the software.
- It used to design the Architecture of the proposed system. Also it focuses on the physical construction, hardware, operating systems, system modules, programming, communications and security issues, and so on.

There are two kinds of design documents developed in this phase:

High-Level Design (HLD)	Low-Level Design(LLD)
<ul style="list-style-type: none">• Brief description and name of each module• An outline about the functionality of every module• Interface relationship and dependencies between modules• Database tables identified along with their key elements• Complete architecture diagrams along with technology details	<ul style="list-style-type: none">• Functional logic of the modules• Database tables, which include type and size• Complete detail of the interface• Addresses all types of dependency issues• Listing of error messages• Complete input and outputs for every module

Phase – 4 Development

- This phase produces the software under development. It is used to build technical architecture, databases and programs (developers start coding) according to the requirements and the design discussed in previous phase.
- It can be conducted Depending on the methodology which we proposed to develop the system.
- This development phase used to ensure that the stakeholders’ expectations are being met.

Database admins create the necessary data in the database, front-end developers create the necessary interfaces and GUI to interact with the back-end all based on guidelines and procedures defined by the company.

Phase – 5 Testing

- The testing phase is one of the most important. During testing, experienced testers start to test the system against the requirements.
- The testers aim to find defects within the system as well as verifying whether the application behaves as expected and according to what was documented in the requirements analysis phase.

Phase – 6 Implementation

- The project takes shape during the implementation phase. It involves the construction of the actual result of the project when we use the build in the proposed environment. Actually, the system is ready to use by the users.
- This phase is important to maintain the momentum.
- At the end of the implementation phase, the result is evaluated according to the list of requirements that was created in the beginning phase.

Phase – 7 Maintenance

- This phase is used to ensure proper operation of the system which was implemented by the team.
 - If an issue is encountered during the maintenance, the team must take actions depending on how severe the issue. It might either require a hot-fix which is created and shipped in a short period of time or if not very severe, it can wait until the next version of the software
 - Once a version of the software is released to production, there is usually a maintenance team that looks after any post-production issues.
-

Software Requirement Specification

- SRS is a document created by system analyst after the requirements are collected from various stakeholders.
- SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.
- The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.
- SRS should come up with following features:
 - User Requirements are expressed in natural language.
 - Technical requirements are expressed in structured language, which is used inside the organization.
 - Design description should be written in Pseudo code.
 - Format of Forms and GUI screen prints.
 - Conditional and mathematical notations for DFDs etc.