

**Phase-1: Requirement Phase**

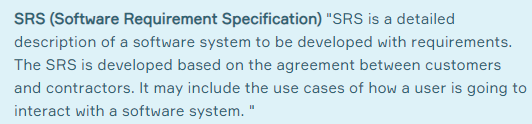
The requirement is the first and **the most critical phase of SDLC** for both the developing team and the project manager. During this phase, the client specifies requirements, specifications, expectations and any other special requirement related to the product or software. The business manager or project manager gathers all of this information and also prerequisites.

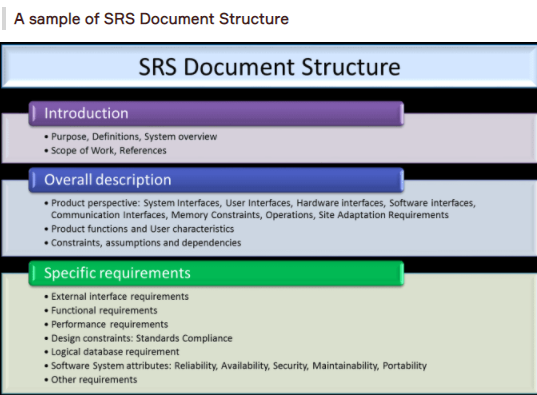
All the information gathered from this phase is crucial to developing the product as per the customer requirements.

To develop the software system we should have a clear understanding of the desired product/software. To achieve this we need to continuous communication with customers to gather all requirements.

Once all the information gathered, the next step is to clearly define and document the product requirements and get them approved by the customer or the market analysts.

This is done through an **SRS (Software Requirement Specification)** document. It consists of all the necessary requirements to be designed and developed during the project life cycle.





### Phase-2: Design Phase

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

This is the high priority phase in a system's development life cycle because the **logical designing** of the system is converted into **physical designing**. The output of the requirement phase is a list of things that are required and the design phase gives the way to accomplish these requirements. The decision of all required essential tools such as **programming language** like Java, .NET, PHP; **database** like Oracle, MySQL; **a combination of hardware and software** to provide a platform on which software can run without any problem is taken in this phase.

There are several tools and techniques used for describing system design, such as Flowchart, Data flow diagram (DFD), Data dictionary, Structured English, Decision table, and Decision tree.

### Phase-3: Build/Development Phase

After the successful completion of the requirement and design phase, the next step is to implement the design into the development of a software system.

This phase is also known as **coding phase**.

Developers start to build the entire system by writing code using the chosen programming language.

Work/task is divided into small units or modules, and coding starts by the team of developers according to the design and the requirements of the client to produce the desired result.

**Coding Phase is the longest phase of the SDLC process**, and it requires a more focused approach for the developer.

### Phase-4: Testing Phase

Once the software is complete, it is the time for the testing phase.

This phase is where you focus on investigation and discovery. The testing team starts testing the functionality of the entire system. This is done to verify that the software works and gives the result as per the requirements addressed in the requirement phase or not.

The development team makes a **test plan** to start the test. This test plan includes all types of essential testing such as integration testing, unit testing, acceptance testing, and system testing.

If there is a bug/defect detected in the software, or it is not working as expected. The testing team gives detailed information to the development team about the issue. If the defect is valid or worth fixing, it will be fixed and the development team replaces it with the new one. It also needs to be verified.

### Phase-5: Deployment/Deliver Phase

When software testing is completed with a satisfying result and there are no remaining issues in the working of the software, it is delivered to the customer.

As soon as customers receive the product, they are recommended first to do the beta testing. In beta testing, customers can require any changes which are not present in the software but mentioned in the requirement document to make it more user-friendly.

Besides this, if any type of defect is encountered while a customer using the software, the development team will be informed to fix this problem. If it is a critical defect, the development team solves it in a short time. Otherwise, it will wait for the next version.

After the solution of all types of bugs and changes, the software finally deployed to the end-user.

### Phase-6: Maintenance Phase

The **last** phase of the process SDLC is the maintenance phase where the process continues until the software's life cycle comes to an end. When a customer starts using software, actual problems start to show up. At that time, there's a need to solve these problems.

Maintenance Phase also includes making changes in hardware and software to maintain its operational effectiveness like to improve its performance, enhance security features and address customer's requirements.

## **SDLC Models**

A SDLC model describes the types of activities performed in a software development project at each stage, and how the activities relate logically and chronologically to each other.

There are many different SDLC models, each of which requires different approaches to testing.

### Software Development and Software Testing

It is an important part of a tester's role to be familiar with the common SDLC models so that appropriate test activities can take place. In any SDLC model, there are several characteristics of good testing:

* For every development activity, there is a corresponding test activity
* Each test level has test objectives specific to that level
* Test analysis and design for a given test level begin during the corresponding development activity
* Testers participate in discussions to define and refine requirements and design, and they are involved in reviewing work products as soon as drafts are available

No matter which SDLC model is chosen, test activities should start in the early stages of the life cycle, adhering to the testing principle of early testing.

### Verification & Validation

In every development life cycle, a part of testing is focused on **verification** testing, and another part is focused on **validation** testing.

Verification is concerned with evaluating a work product, component, or system to determine whether it meets the requirements set. Verification focuses on the question **Is the deliverable built according to the specification?**.

Validation is concerned with evaluating a work product, component, or system to determine whether it meets the user needs and requirements. Validation focuses on the question **Is the deliverable fit for purpose, and does it provide a solution to the problem?**.

### SDLC Models

There are various Software development models or methodologies. They are as shown below:

