

SDLC Models

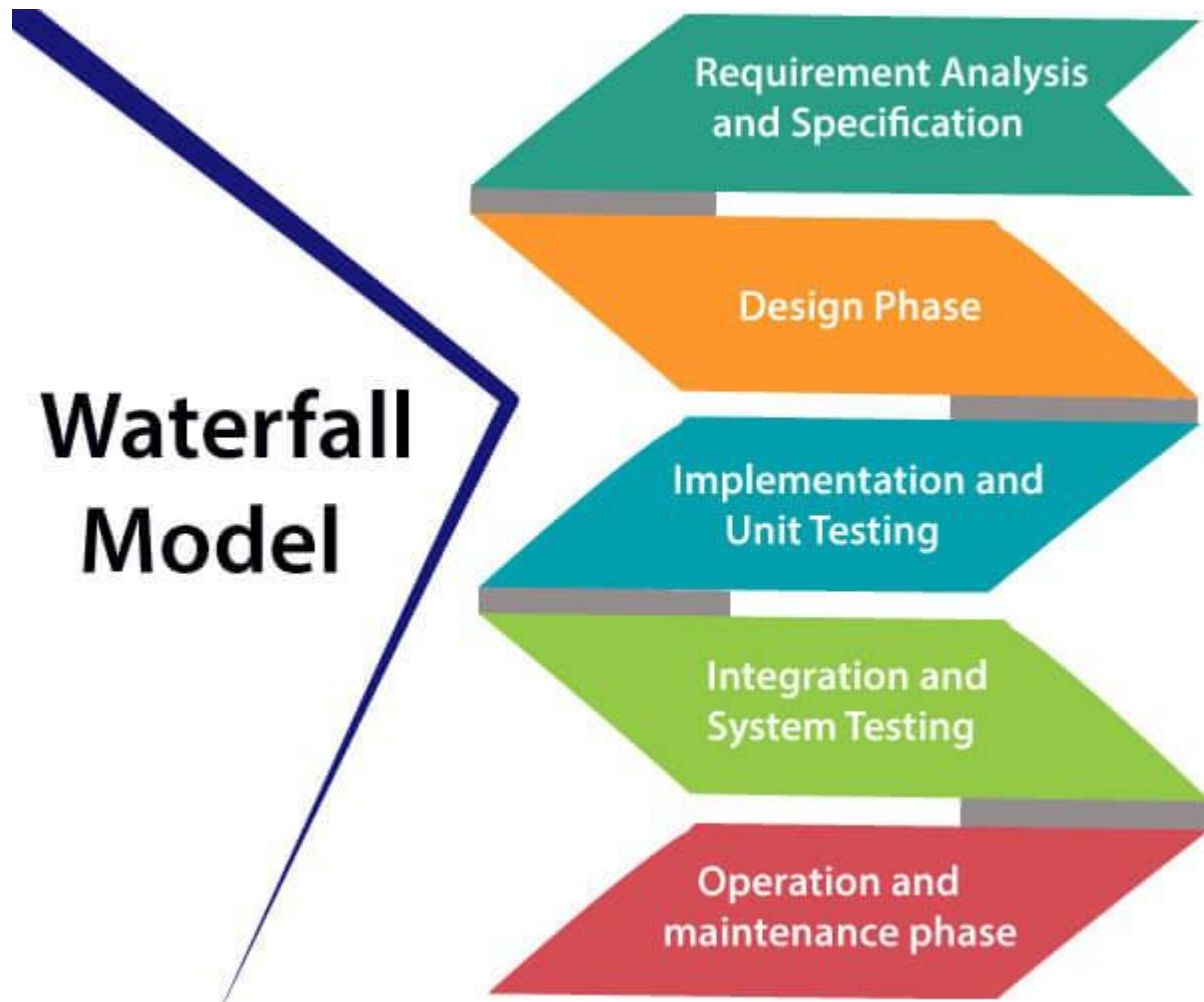
(1) Sequential development models

A sequential development model describes the software development process as a linear, sequential flow of activities.

This means that any phase in the development process should begin when the previous phase is complete.

(a) Waterfall Model

In the Waterfall model, the development activities (e.g., requirements analysis, design, coding, testing) are completed one after another. In this model, test activities only occur after all other development activities have been completed. The Waterfall model is the earliest SDLC approach that was used for software development.



When to use SDLC Waterfall Model?

Some Circumstances where the use of the Waterfall model is most suited are:

- When the requirements are constant and not changed regularly.
- A project is short
- The situation is calm
- Where the tools and technology used is consistent and is not changing
- When resources are well prepared and are available to use.

Advantages of Waterfall model

- This model is simple to implement also the number of resources that are required for it is minimal.
- The requirements are simple and explicitly declared; they remain unchanged during the entire project development.
- The start and end points for each phase is fixed, which makes it easy to cover progress.
- The release date for the complete product, as well as its final cost, can be determined before development.
- It gives easy to control and clarity for the customer due to a strict reporting system.

Disadvantages of Waterfall model

In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects. This model cannot accept the changes in requirements during development.

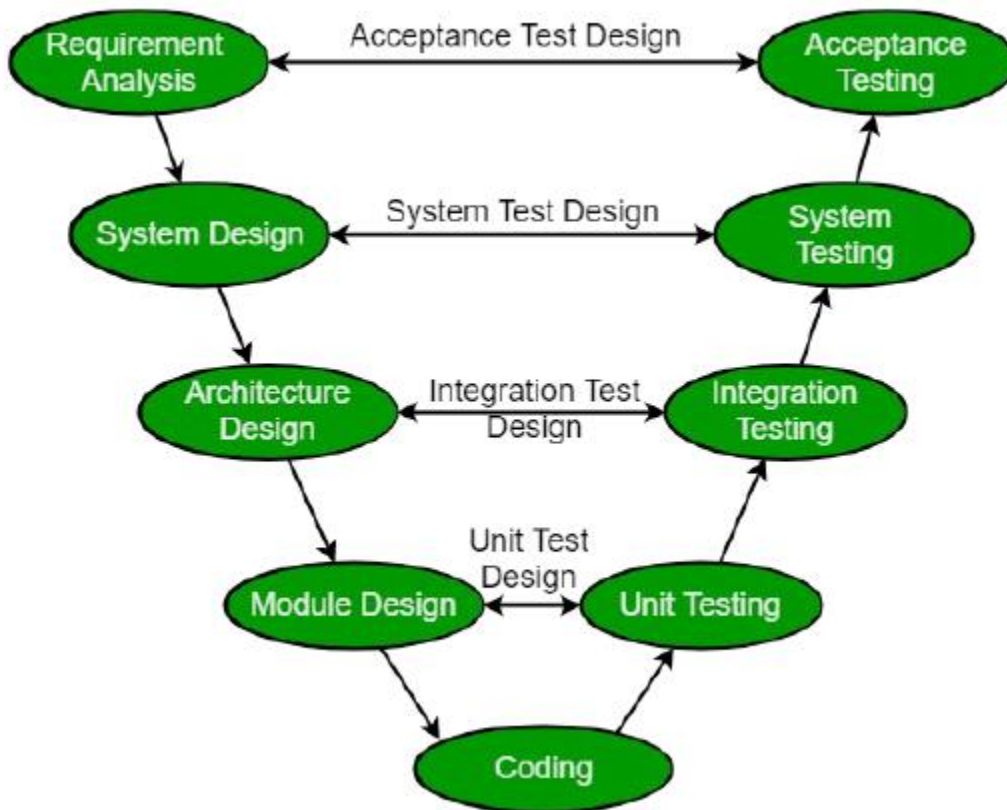
It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.

Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

(b)V-model

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly

associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.



When to use?

Where requirements are clearly defined and fixed.

The V-Model is used when ample technical resources are available with technical expertise.

Advantages:

This is a highly disciplined model and Phases are completed one at a time.

V-Model is used for small projects where project requirements are clear.

Simple and easy to understand and use.

This model focuses on verification and validation activities

early in the life cycle thereby enhancing the probability of building an error-free and good quality product. It enables project management to track progress accurately.

Disadvantages:

High risk and uncertainty.

It is not a good for complex and object-oriented projects.

It is not suitable for projects where requirements are not clear and contains high risk of changing.

This model does not support iteration of phases.

It does not easily handle concurrent events.

(2) Iterative and incremental development models.

Iterative and incremental development is a process that combines the iterative design method with the incremental build model. It is used by software developers to help manage projects.

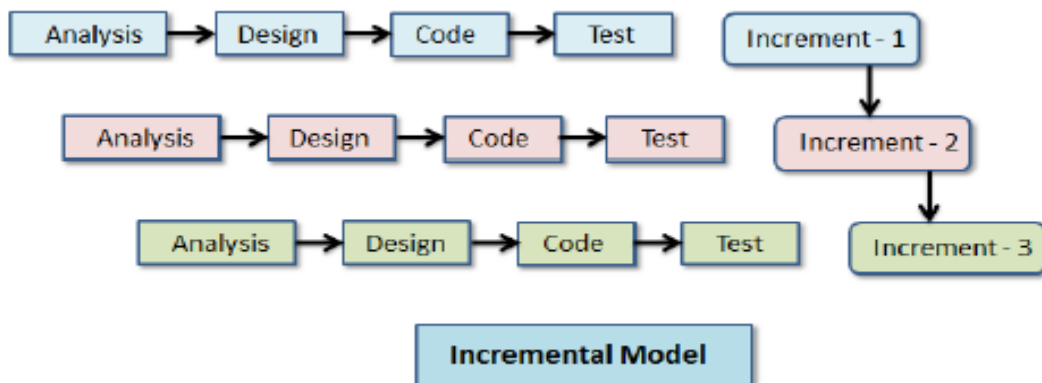
-incremental: An incremental approach breaks the software development process down into small, manageable portions known as increments. Each increment builds on the previous version so that improvements are made step by step.

In simple words Incremental development is a development approach that slices the product into fully working slices that are called increments. Each new increment builds on top of the existing released functionality.

Example: Ecommerce website

Consider a team building an ecommerce website using incremental development. The final target product has search, product information, a shopping basket, checkout, favourites, and customer reviews.

For the first released increment, the team builds the basic functionality to buy a product. It includes search, product information, adding products to a shopping basket and checkout. This first slice would only be released once it's complete. The second released increment builds on that basic functionality, and would add another capability such as favourites. The would be released when the favourites functionality is complete. The third released increment adds customer reviews once that is complete, and so on.



Iterative: An iterative model means software development activities are systematically repeated in cycles known as iterations. A new version of the software is produced after each iteration until the optimal product is achieved.

Iterative development is when teams gradually build up the features and functions but don't wait until each of these is complete before releasing. They release a basic version of each feature and then add to that feature in subsequent iterative releases, usually based on feedback from the basic version released.

Example: Ecommerce website

Assume a team building the same ecommerce website using an iterative process. The first release has a really stripped back version of all the required functionality; namely search, product information, a shopping basket, checkout, favourites, and customer reviews. For the second iterative release, the team would improve some of the existing basic functionality, taking into account feedback from stakeholders or customer, or other inputs such as analytics.

What is Agile project management?

Agile project management is an iterative approach to project management that focuses on breaking down large projects into more manageable tasks, which are completed in short iterations throughout the project life cycle. Teams that adopt the Agile methodology are able to complete work faster, adapt to changing project requirements, and optimize their workflow.

Rational Unified Process: Each iteration tends to be relatively long (e.g., two to three months), and the feature increments are correspondingly large, such as two or three groups of related features.

Scrum: Each iteration tends to be relatively short (e.g., hours, days, or a few weeks), and the feature increments are correspondingly small, such as a few enhancements and/or two or three new features

Kanban: Implemented with or without fixed-length iterations, which can deliver either a single enhancement or feature upon completion, or can group features together to release at once

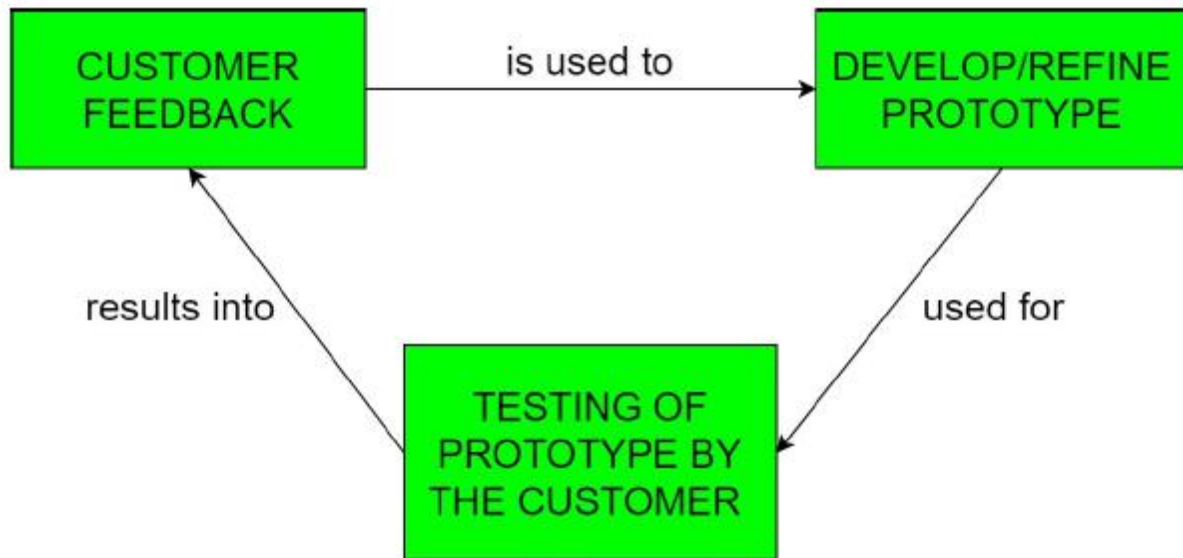
Spiral: Involves creating experimental increments, some of which may be heavily re-worked or even abandoned in subsequent development work

(3)The Prototyping Model

Prototyping Model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

In this process model, the system is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle. The process starts by interviewing the customers and developing the incomplete high-level paper model. This document is used to build the initial prototype supporting only the basic functionality as desired by the customer. Once the customer figures out the problems,

the prototype is further refined to eliminate them. The process continues until the user approves the prototype and finds the working model to be satisfactory



What is Manual Testing

Manual Testing is a process of finding out the defects or bugs in a software program. In this method the tester plays an important role of end user and verifies that all the features of the application are working correctly.

The tester manually executes test cases without using any automation tools.

The tester prepares a test plan document which describes the detailed and systematic approach to testing of software applications.

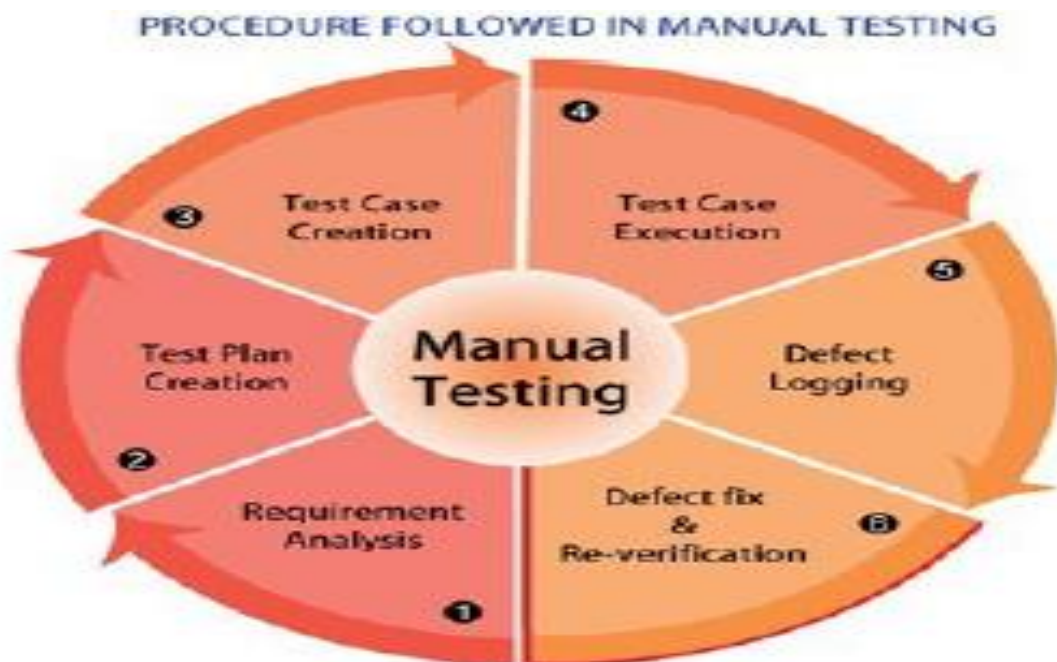
Test cases are planned to cover almost 100% of the software application. As manual testing involves complete test cases it is a time consuming test.

The differences between actual and desired results are

treated as defects. The defects are then fixed by the developer of software application. The tester retests the defects to ensure that defects are fixed. The goal of Manual testing is to ensure that application is defect & error free and is working fine to provide good quality work to customers.

Procedure of Manual Testing

- Requirement Analysis
- Test Plan Creation
- Test case Creation
- Test case Execution
- Defect Logging
- Defect Fix & Re-Verification



(1)Requirement Analysis

Some major task of requirement analysis:

To review the test basis. The test basis is the information on which test cases are based, such as requirements, design specifications, product risk analysis, architecture and interfaces

To identify test conditions

To design the tests

To design the test environment set-up and identify the required infrastructure and tools

(2)Test Plan Creation

Test planning involves producing a document that describes an overall approach and test objectives. It involves reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases and Designing the test environment. Completion or exit criteria must be specified so that we know when testing (at any stage) is complete.

To determine the scope and risks and identify the objectives of testing.

To determine the required test resources like people, test environments etc.

To schedule test analysis and design tasks, test implementation, execution and evaluation.

(3)Test case Creation

To develop and prioritize test cases by using techniques and create test data for those tests.

(4)Test case Execution

Test execution involves actually running the specified test on a computer system either manually or by using an automated test tool.It is a Fundamental Test Process in which actual work is done.

Test implementation has the following major task:

To create test suites from the test cases for efficient test execution. Test suite is a collection of test cases that are used to test a software program

To re-execute the tests that previously failed in order to confirm a fix.

To log the outcome of the test execution. A test log is the status of the test case (pass/fail).

How to do Manual Testing

Here's how to perform manual testing step by step:

- Analyze requirements from the software requirement specification document
- Create a clear test plan
- Write test cases that cover all the requirements defined in the document
- Get test cases reviewed by the QA lead
- Execute test cases and detect any bugs
- Report bugs, if any, and once fixed, run the failed tests again to re-verify the fixes

Manual testing can never be avoided entirely as it is a continuous process that requires human verification at regular intervals throughout the software development lifecycle. As a result, it's essential for teams to find the right balance between manual and automated tests.

Test Plan

Test plan can be defined as a document for a software project which defines the approach, scope, and intensity on the effort of software testing.

Test Strategy

The test strategy is a set of instructions or protocols which explain the test design and determine how the test should be performed.

The objective of the Test Strategy is to provide a systematic approach to the software testing process in order to ensure the quality, traceability, reliability and better planning.

Test Plan has the primary goal of how to test, when to test and who will verify whereas Test Strategy has the primary goal of what technique to follow and which module to check.

Test Report

Test Execution Report or Test Summary Report is a document which contains a summary of all test activities and final test results of a testing project.

Test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

What does a test report contain?

Project information	Test objective	Test summary	Defect
<ul style="list-style-type: none">• Project Name• Description	<ul style="list-style-type: none">• Test Type• Purpose	<ul style="list-style-type: none">• Test Passed• Test Failed• Test Blocked	<ul style="list-style-type: none">• Description• Priority• Status