

# **T3 - IDENTIFY SOFTWARE TEST LIFE CYCLE**

---

## **SOFTWARE TESTING**

**Module Code - K72C001M05**


**ICT in NVQ Level 5**

**BY:- MS.P.Yasotha**

# Objectives

- This lesson will provide learner
  - Ability to identify Software Test life Cycle
  - Ability to use different Test models
  - Understandability of Test Methods
  - Ability to identify different levels of Testing
  - Ability to review the Test Process.


# Contents

1. Software Testing Life Cycle (STLC)
  2. Software Testing Models
  3. Method of Software Testing
  4. Levels of Software Testing
  5. Review and Test Process.
- 



01

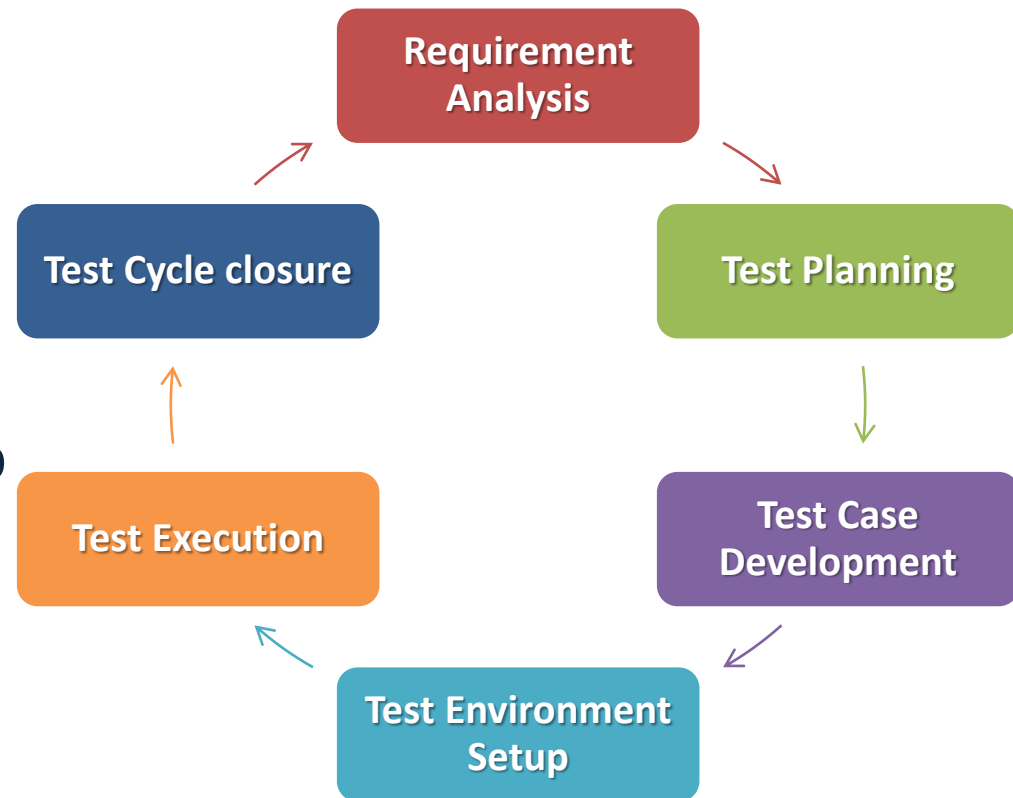
# **SOFTWARE TESTING LIFE CYCLE (STLC)**



Software Testing Life Cycle (STLC) is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product.

### Phases of STLC:

1. Requirement Analysis
2. Test Planning
3. Test Case Development
4. Test Environment Setup
5. Test Execution
6. Test Cycle closure



# Phase 1. Requirement Analysis

- This is the very first phase of Software testing Life cycle (STLC). Testing team goes through the Requirement document with both Functional and non- functional requirements details in order to identify the testable requirements.
- In case of any confusion the QA team may setup a meeting with the clients and the stakeholders in order to clarify their doubts.
  - Example- Technical Leads, Business Analyst, System Architects and Client etc.


# Phase 1. Requirement Analysis

## Activities

- Analyzing the System Requirement specifications from the testing point of view
- Identifying the testing techniques and testing types
- Prioritizing the feature which need focused testing
- Identifying the details about the testing environment where actual testing will be done

# Phase 1. Requirement Analysis

## What is Traceability Matrix? (TM)

- A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship.
  - It is used to track the requirements and to check the current project requirements are met.
- 



# Phase 1. Requirement Analysis


## What is Traceability Matrix? (TM)

Requirement Traceability Matrix												
	Test Case ID →	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8	TC_9	TC_10	# Test Cases for respective Requirement
Req. ID ↓												
Req_1		✖		✖			✖					3
Req_2			✖			✖						2
Req_3				✖								1
Req_4					✖		✖					2
Req_5						✖		✖				2
Req_6							✖					1
Req_7						✖		✖				2
Req_8									✖			1
Req_9										✖		1
Req_10											✖	1

## Phase 2. Test Planning


- It starts soon after the completion of the Requirement Analysis phase. The QA manager or QA Lead will prepare the Test Plan and Test Strategy documents.
- Usually IEEE 829 test plan template is used for test planning.

### Activities

- Estimation of testing effort
  - Selection of Testing Approach
  - Preparation of Test Plan, Test strategy documents
  - Resource planning and assigning roles and responsibility to them
  - Selection of Testing tool
- 

## Phase 2. Test Planning

### What is a Test Plan?

- It is a detailed document that describes the test strategy, objectives, schedule, estimation and deliverables and resources required for testing.
  - Test Plan helps us determine the effort needed to validate the quality of the application under test.
  - The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.
- 

# Phase 2. Test Planning


## How to write a Test Plan

- You already know that making a **Test Plan** is the most important task of Test Management Process.
  - Analyze the product
  - Design the Test Strategy
  - Define the Test Objectives
  - Define Test Criteria
  - Resource Planning
  - Plan Test Environment
  - Schedule & Estimation
  - Determine Test Deliverables

# Phase 3. Test case Development

- In this phase the QA team writes test cases for testing the product and they write scripts for automation if required.
- Verification of both the test cases and test scripts are done by peers. Creation of Test Data is done in this phase.


## Activities

- Creation of test cases
  - Creation of test scripts if required
  - Verification of test cases
  - Creation of Test Data in testing environment
- 

# Phase 4. Test Environment Setup

- This phase includes the setup or installation process of software and hardware which is required for testing the application to test the build.
- Once the installation of build is successful and complete then the Test Data is generated.

## Activities

- As per the Requirement and Architecture document the list of required software and hardware is prepared
  - Setting up of test environment
  - Creation of test data
- 

# Phase 5. Test Execution

- In Test Execution phases the test cases are executed when the Test Environment setup is ready.
- QA team executes the test cases to find the bugs and which will be reported against the requirements.
- This bug is fixed by the developer and is retested by the QA.

## Activities

- Execution of Test Cases
- Reporting test results
- Logging defects for the failed test cases
- Verification and retesting of the defect
- Closure of defects



# Phase 5. Test Execution

## Defect report

It is a document that identifies and describes a defect detected by a tester. The purpose of a defect report is to state the problem as clearly as possible so that developers can replicate the defect easily and fix it.

Note:- While reporting the bug to developer, your Bug Report should contain the following information


- |                       |                |                |
|-----------------------|----------------|----------------|
| 1. Defect             | 6. Date Raised | 11.Fixed by    |
| 2. Defect Description | 7. Reference   | 12.Date Closed |
| 3. Version            | 8. Detected by | 13.Severity    |
| 4. Steps              | 9. Status      | 14.Priority    |



# Phase 6. Test Cycle Closure

- In order to start the Test Cycle Closure activity the Test Execution phase should be completed. In Test Cycle phase the QA team will meet and discuss about the testing artifacts.
- The whole intent of this discussion is to learn lessons from the bad practices. This will help in future projects.

## Activities

- To evaluate the test completion on the basis of Test Coverage and Software Quality
  - Documentation of the learning from the project
  - Analyzing the test results
  - Test Closure Report preparation
- 

**YOUR TASK!!**

**TRY TO DISCUSSES THE  
DIFFERENCES OF  
“SDLC and STLC”**




**02**

# **SOFTWARE TESTING MODELS**


# What is Software Testing Models?

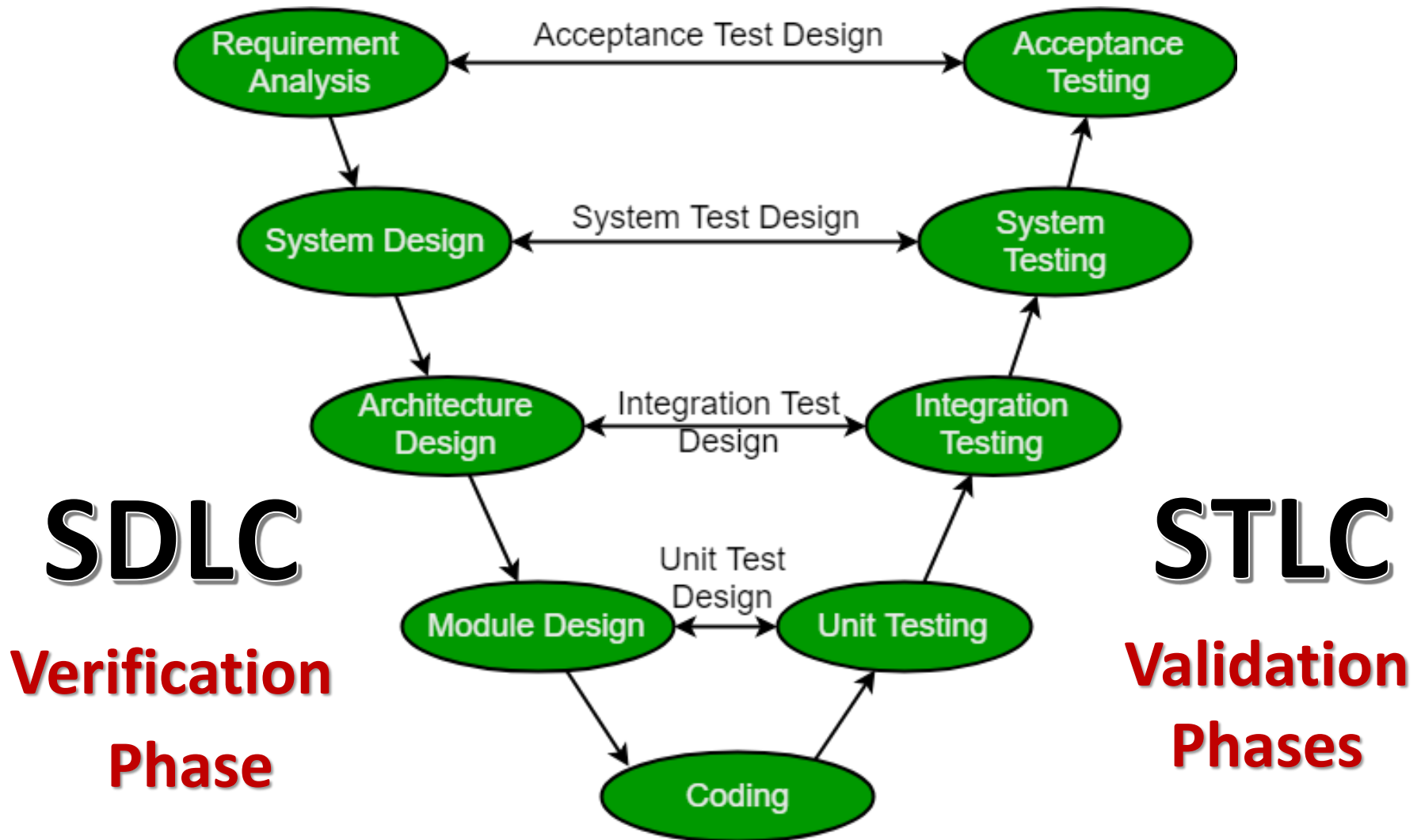
- Software Testing Model refers as Approaches, are used in the software development process where each model has its own advantages and disadvantages.
- This define the strategies and testing types used to certify that the “**Application Under Test meets client expectations**”.

## Example

- Agile methodology
  - Waterfall model
  - Iterative development
  - V-Model and etc.
- 

# What is 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 based on the association of a testing phase for each corresponding development stage.
  - Development of each step directly associated with the next phase and each development activity, there is a testing activity corresponding to it.
- 



The entire figure looks like a V, hence the name **V - model**


# Verification Phase

- Verification is **Static Testing** and the process of checking that a software achieves its goal without any bugs.
- It is the process to **ensure whether the product that is developed is right or not.**
- It verifies whether the developed product fulfills the requirements that we have.

## Phases

### Requirement Analysis

This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.



# Verification Phase

## **System Design**


This phase contains the system design and the complete hardware and communication setup for developing product.

## **Architectural Design**

System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside system.

## **Module Design**

In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).





# Validation Phase

- Validation is the **Dynamic Testing** process and checking whether the software product has high level requirements.
- It is used to **checks what we are developing is the right product**. it is validation of actual and expected product.

## Phases

### Unit Testing

Unit Test Plans are developed during module design phase. And it is executed to eliminate bugs at code or unit level. When it completed then the integration testing is performed.

# Validation Phase

## **Integration testing**

In this testing, the modules are integrated and the system is tested. It is performed on the Architecture design phase and it verifies the communication of modules among themselves.

## **System Testing**

It tests the complete application with its functionality (both functional and non-functional requirements ), inter dependency, and communication.


## **User Acceptance Testing (UAT)**

UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets user's requirement and system is ready for use in real world.


# Coding Phase

- System modules designed in the design phase is taken up in the Coding phase and which is performed based on the guidelines and standards.
- The best suitable programming language is decided based on the system and architectural requirements.
- The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

# 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.
- 
- The bottom of the slide features a decorative graphic consisting of several overlapping, wavy lines in shades of blue, teal, and light grey, creating a modern, abstract background element.

# **YOUR TASK!!**

**Explain the V Model process?  
Why is it different than the  
Waterfall model?**

