

What is STLC?

- STLC stands for **Software Testing Life Cycle**

- **Systematic** and **well-defined** steps performed during the testing of a software application

- STLC helps in a better understanding of the application which results in overall better testing

- STLC also helps to identify bugs from the app and report it

- STLC Steps:
1. Requirement Analysis

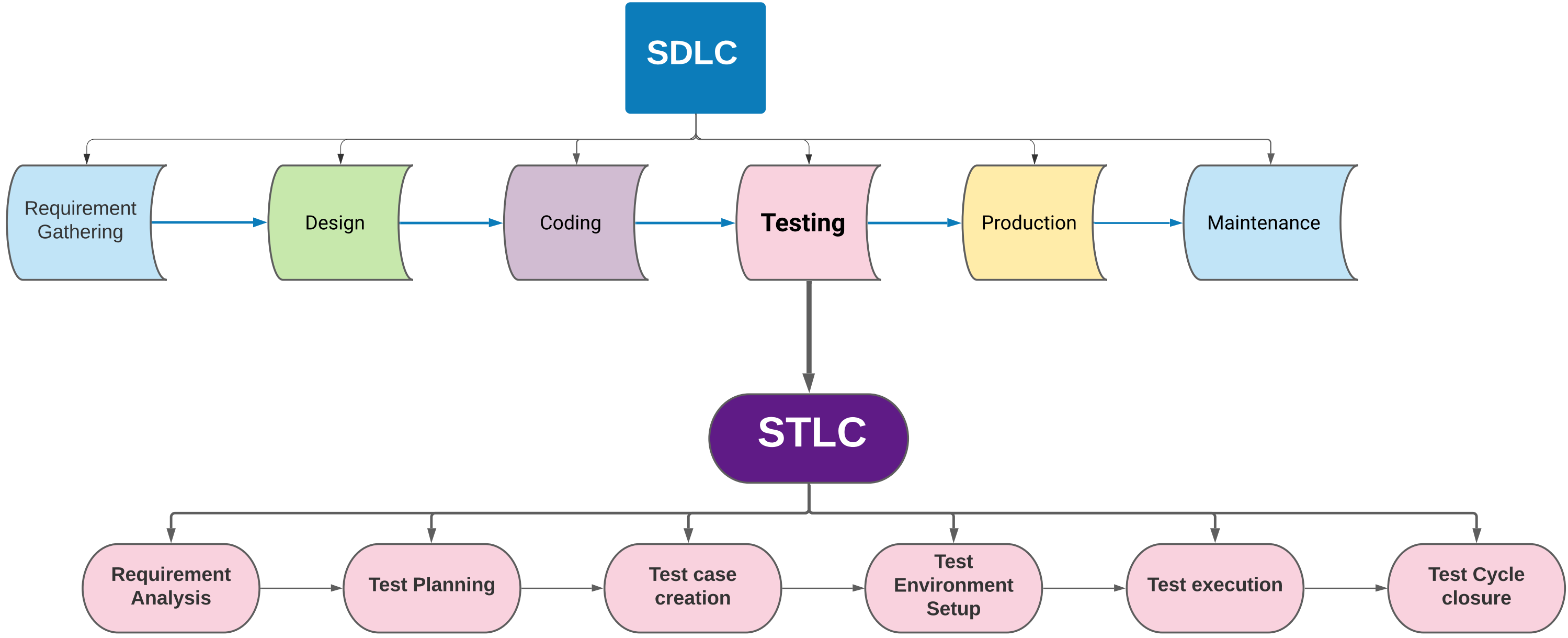
2. Test Planning

3. Test case creation

4. Test Environment setup

5. Test Execution

6. Test Cycle closure



- Step 1: Requirement Analysis
- **Testing team** studies the **requirement doc in Waterfall**

• **In Agile**, requirement is user story, each tester has one story at a time to analyze

• It is important to **understand** the client's request for the project

• Analyze if the story can be automated ? Is it about front-end functionality or back-end?

- Step 2: Test Planning
- In **Waterfall**, there is a **test plan document** must created

• **In Agile**, planning testing is handled through a meeting **with or without** a test plan document

• Test Plan document generally **prepared by the QA lead** or Senior QA **in Agile**

• Many scrum team creates **one test plan per release**

• Click here to see a sample test plan document

Components of A Test Plan	
Items in test plan template	What do they contain?
Introduction	Purpose of the test plan document Project brief introduction
Test strategy	Testing framework - BDD cucumber framework Scope of testing - Testing types list & overall info Environments
Schedule	Sprint cycles -starting & ending date
Roles & Responsibilities	Team members are listed How to track bugs? - e.g: jira Who is going to be in charge?
Recourses	Software lists - e.g: VM, jira, github Hardware list - e.g: HP
Risks & Mitigation	Potential / possible risks are listed What to do in case of problems?
Approvals	The Names and Titles of all persons who must approve this plan

- Step 3 : Test case creation
- **QA creates test case doc for per requirement/user stories**

- There is at least one Test Case for one user story

- **A Test Case document has : ID, test case summary, test data, environment, test steps , actual & expected results.**

- create automated **test script** / code also done in this step

Test data:

• **Input** to perform both positive and negative test cases

• Test Data can be Generated:

• Manually from dummy data generator website

• Mass copy of data from production

• Mass copy of test data from legacy client system

• Automated Test Data Generation Tools

Your functional test cases can have:

• **No data:** Check system response when no data is submitted

• **Valid data:** Check system response when Valid test data is submitted

• **Invalid data:** Check system response when *Invalid* test data is submitted

- Test case steps has 2 types: 1. **manual** test steps
2. **Cucumber** test steps -> use gherkin language

Manual test steps:

steps: step 1. user go to login page
step 2. user enter user name & password
step 3. click "login" button
step 4. Verify user login successfully launched on the home page

Cucumber test steps:

Given user go to login page
When user enter user name & password
And user click "login" button
Then Verify user login successfully launched on the home page

- Step 4 : Test Environment Setup
- testers should make sure all the new codes are deployed to the QA/test environment

- The different types of testings manually, automatically performed in test/qa environment.

Dev environment

QA Environment --> Functional & Non-Functional performed

staging Environment

Production Environment

- Step 5 : Test Execution
- test case execution means both **manual testing & automation executed**

- **expected result vs actual result**

- **create bug report** if expected and actual **results are not match**

- Step 6 : Test Closure
- Checking if all the **project deliverable are delivered**

- test lead and business team **generate reports** to **evaluate** and learn from the process

