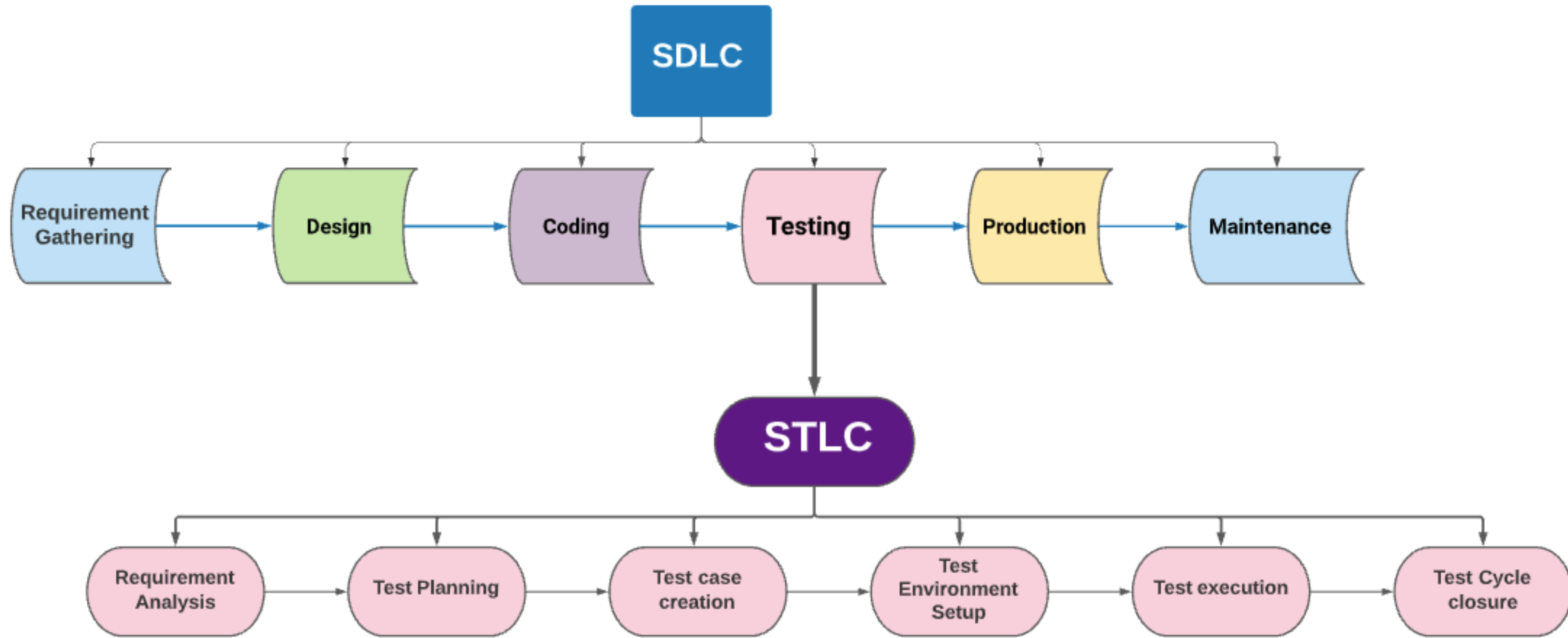


STLC - Software Testing Life Cycle

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



Step1: Requirement Analysis

- *In Agile, requirements are the user stories*
- 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 or back-end?

Example1:

As a user, I want to log in to the app_UI.

Analyze:

Which functionality am I testing? -> log in function

Is this UI/FrontEnd testing or BackEnd testing? -> UI/FrontEnd testing

Manually Test or automatically? or both? -> both (also depending on your role in the team)

Example2:

As a data consumer, I want the user credentials(username & password) stored in MySQL DB correctly in the user's table._DB

Analyze:

Which functionality am I testing? -> log in function

Is this UI/FrontEnd testing or BackEnd testing? -> BackEnd testing

Manually Test or automatically? or both? -> both (also depending on your role in the team)

Example1:

As a user, I want to log in to the app_UI.

Analyze:

Which functionality am I testing? -> log in function

Is this UI/FrontEnd testing or BackEnd testing? -> UI/FrontEnd testing

Manually Test or automatically? or both? -> both (also depending on your role in the team)

Example2:

As a data consumer, I want the user credentials(username & password) stored in MySQL DB correctly in the user's table._DB

Analyze:

Which functionality am I testing? -> log in function

Is this UI/FrontEnd testing or BackEnd testing? -> BackEnd testing

Manually Test or automatically? or both? -> both (also depending on your role in the team)

Example3:

As a manager, I want to use the Get method API to retrieve the list of user accounts. _API

Analyze:

Which functionality am I testing? -> log in function

Is this UI/FrontEnd testing or BackEnd testing? -> BackEnd testing

Manually Test or automatically? or both? -> both (also depending on your role in the team)

Step 2: Test Planning

- 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
- Many scrum teams plan testing for each release.
- It is common that there is a master test plan template in the companies that can be used for each project.
- In test planning, testers determine:
 - Who are the **testers**?
 - What is the role & **responsibility** of each tester?
 - What is the test **strategy**?
 - Which **features** will be tested by the next release?
 - Which **testing types** will be performed?
 - In which **Environment** that testing are performed?
 - What are the **testing tools** will be used? any tool needs to be updated or rollback?
 - What the is **due date**? /Schedule?
 - What are the software and hardware **resources**? any app - jira, aws, VM, etc
 - What is the **input** and **output**? or **entry** Criteria & **Exit criteria**?

Test plan doc sample: <https://docs.google.com/document/d/1xyeSwF7ap9G-sX89aXmYJfane9K94CLy/edit?usp=sharing&ouid=105312414068310747400&rtpof=true&sd=true>

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?
Resources	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

- Test case is a document
- QA creates test case doc for per requirement/user story
- There is at least one Test Case for one user story
- Test case document is used for both manual & automation testing
- A Test Case document has:
 - Test case ID
 - Test case description /summary
 - Test data
 - Test Environment
 - Test steps
 - Actual & expected results
 - Test pass / fail result
- Creating automated **test script** / code is also done in this step
- Map the test cases to the user stories in **RTM** / project management tool(jira)

User Story: As a user, I want to login to the Cydeo app_UI.

test case 1:

ID: 001

Description: [Teachers](#) login with valid credential

Steps: step 1. user go to login page
step 2. user enter valid user name & password
step 3. click "login" button
step 4. Verify user login successfully launched on the home page and [see 4 modules](#)

Data: [user name:](#) Teacher1 , teacher23, teacher46, teacher100
[password:](#) abc123

Environment: [http:// QA.cydeo.com](http://QA.cydeo.com)

test case 2:

ID: 002

Description: [students and mentors](#) login with valid credential

Steps: step 1. user go to login page

step 2. user enter valid user name & password

step 3. click "login" button

step 4. Verify user login successfully launched on the home page and [see 2 modules](#)

Data: [user name:](#) student1 , student20, student55, student100

mentor1 , mentor20, mentor55, mentor100

[password:](#) abc123

Environment: [http:// QA.cydeo.com](http://QA.cydeo.com)

test case 3:

ID: 003

Description: users fails to login with invalid username

Steps: step 1. user go to login page

step 2. user enter **invalid user name** & password

step 3. click "login" button

step 4. Verify the user see "Invalid username, please try again" message

Data: **user name:** asds, 123, muhtar, mug, etc

password: abc123

Environment: [http:// QA.cydeo.com](http://QA.cydeo.com)

test case 4:

ID: 004

Description: users fails to login with invalid password

Steps: step 1. user go to login page

step 2. user enter valid **user name** but **invalid password**

step 3. click "login" button

step 4. Verify the user see "invalid password" message

Data: **user name:** teacher1, student20, mentor22, teacher48, mentor65, student34
password: 9876

Environment: [http:// QA.cydeo.com](http://QA.cydeo.com)

Step 4 : Test Environment Setup

- Testers should make sure all the new codes are deployed from the Dev env to the QA/test environment
- Testers manually and automatically perform testings in test/qa environment.

Dev environment -> Unit & Integration for the developpers

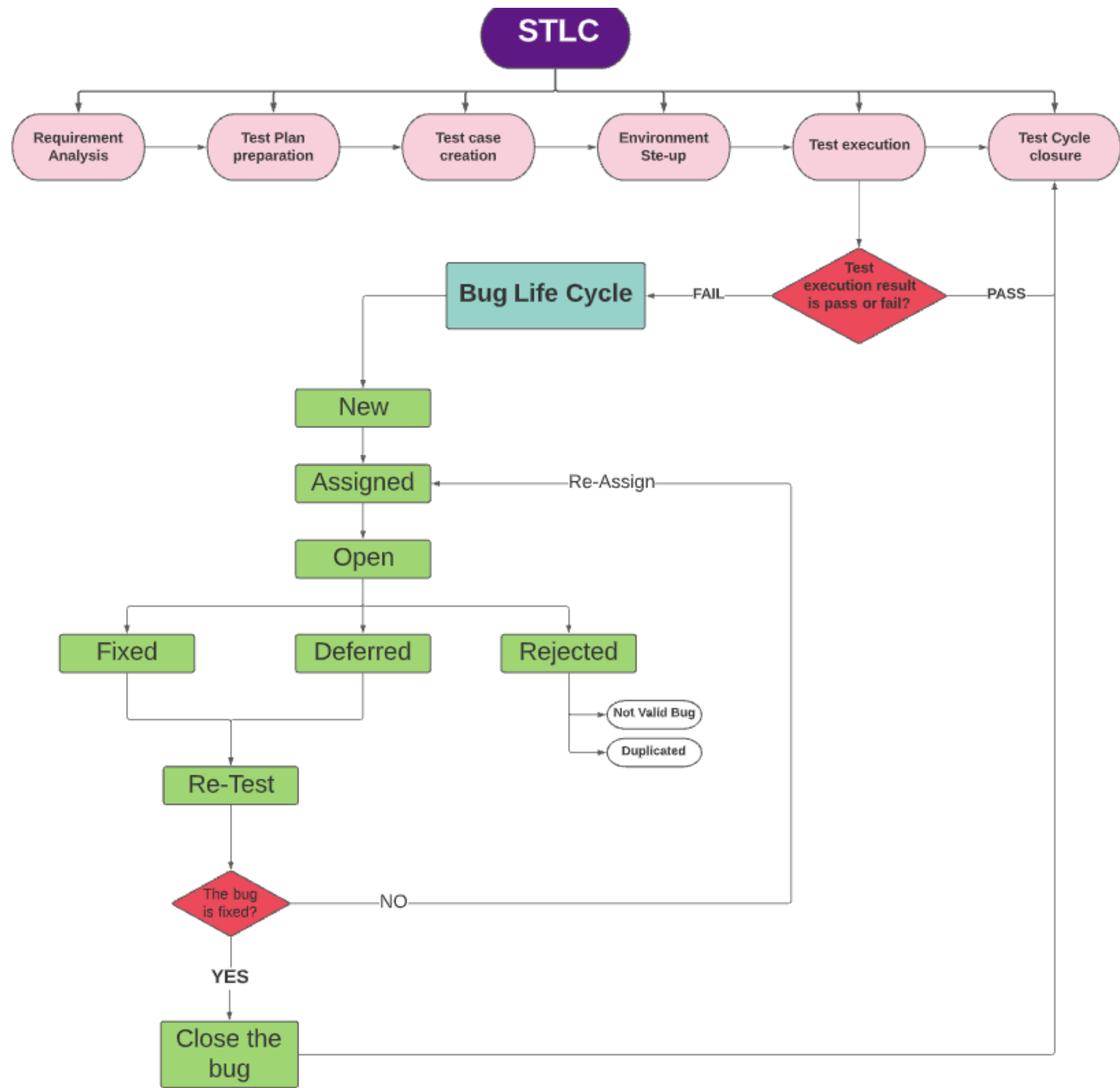
QA Environment --> Functional & Non-Functional -> for the QA and performance testers

staging Environment -> UAT -> the client, UAT team

Production Environment -> End users

Step 5 : Test Execution

- Once the user story is clear, test cases are written, test data & environments are, testers start manual testing and automation testing.
- Test execution means both **manual testing & automation executed**
- **Expected result vs actual result**
 - If expected and actual **results are not match, testers follow the Bug Life Cycle**
 - Testers write bug reports
 - Bugs should be fixed by the developers
 - Testers track the bugs, once a bug is fixed, a tester has to re-test to confirm the bug is removed
- Update the test cases with pass / fail results



Step 6 : Test Closure

- Evaluates each user story and its Test cases to ensure whether they are executed successfully or not, and manages reports for the future references.
- There are many reports are auto generated by project management app like Jira.
- **Test cases result reports includes:**
 - How many total number of test cases are created for a user story ?
 - How many test cases are executed? How many manually ? automatically?
 - Total number of bugs ? how many of them are fixed ? how many of them aren't fixed?
 - Total number of test cases per tester executed? sometimes shows the spent time as well
- In agile-Scrum, reviewing the test case reports can be done between the team lead and the business team, or among the testers in the retro to learn from the lesson.