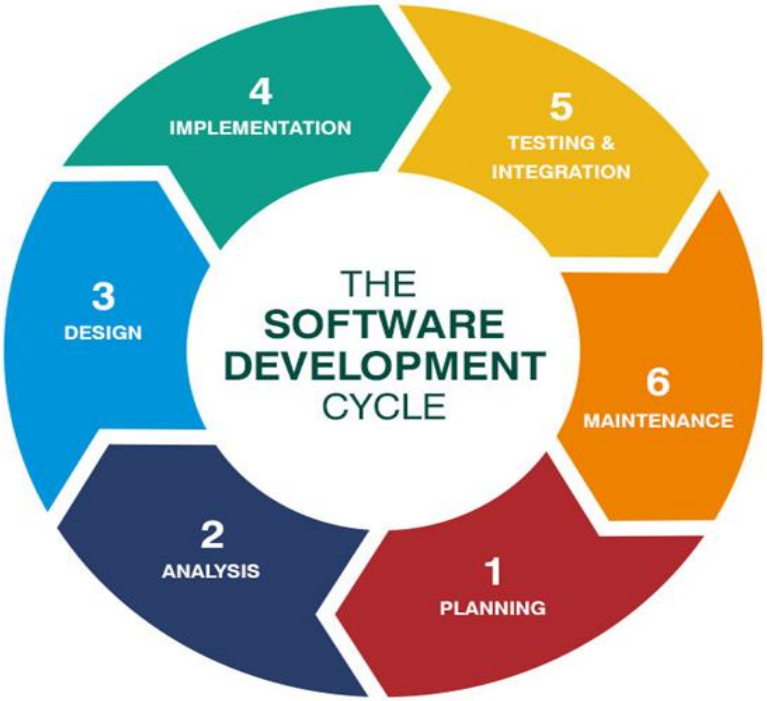
Contents

SDLC Vs STLC

SDLC	STLC
SDLC is mainly related to software development.	STLC is mainly related to software testing.
Besides development other phases like testing is also included.	It focuses only on testing the software.
In SDLC, more number of members (developers) are required for the whole process.	In STLC, less number of members (testers) are needed.
In SDLC, development team makes the plans and designs.	In STLC, testing team makes the plans and designs.
Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
It helps in developing good quality software.	It helps in making the software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.

1) What is SDLC?

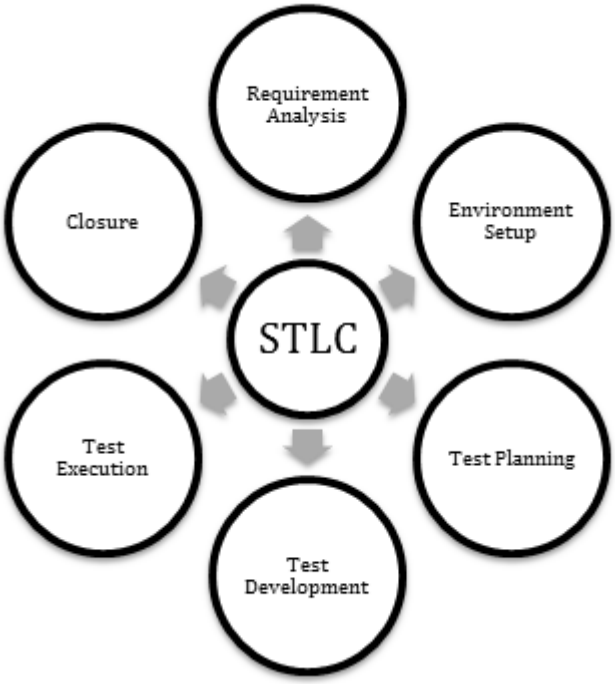
SDLC (Software Development Life Cycle) is the process of developing software through business needs, analysis, design, implementation and Release and maintenance.



2) What is STLC?

The process of testing software in a well planned and systematic way is known as software testing life cycle(STLC).

Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) consists of the following phases.



Software Testing Life Cycle - STLC explained

Requirement Analysis

This is the first phase in STLC where QA team will check the testing requirements for the whole project. For a depth understanding, QA team is generally following clients. It will be opted when only testing is needed not the development. The basic categories under this phase include – Entry criteria, activities performed, and deliverables too.

Test Planning

This is another crucial phase in the STLC. Here, time estimates are calculated before testing is started actually. The outcome of this phase is test plans or documentation. Once you have completed this phase successfully, the QA team can start with development activities.

Test Development

Here, the actual development of test cases will take place once the test planning is completed. Here, not only the test cases but a complete report that contains test data is documented. Once it is done, the report will be cross verified by QA leads.

Environment Setup

This is not practiced because the environment is generally decided in advanced in the SDLC phase only. The QA team will not make any changes to the environment. In case of Automation Environment setup.

Test Execution

Here, tests are executed based on test plans prepared earlier. If they are alright then marked as Pass otherwise Fail. QA team will also prepare a complete list of bugs in this phase that are further forwarded to the development team to fix issues.

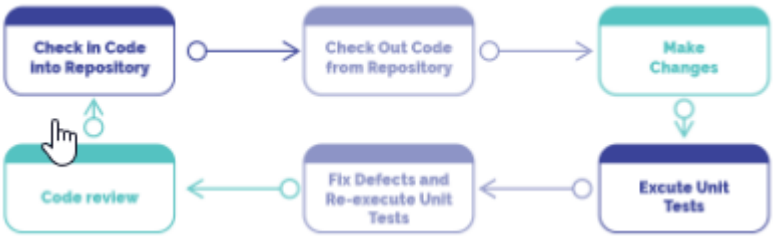
Closure

In this phase, discussion will be made where the team will decide what went right and what was wrong from test perspectives. If you are sure what to improve for the future then it will save time and money both.

Types of Testing

- Manual Testing
- Automation Testing
- Unit Testing
- Integration Testing
- System Testing
- Smoke and Sanity Testing
- What is Regression Testing?
- Non - Functional Testing

Q) Explain Unit Testing?
Unit testing is to verify the correctness of an individual component/module of the system/software. This type of testing is done during the development of a software.
Q) What are different names for Unit Testing?
Unit testing is also called Component Testing, Program Testing or Module Testing.



Q) What is integration Testing?

Individual modules/components of the software are integrated and tested as a group to ensure whether they are functioning as expected with their counterpart.

Q) What is System Testing?

System testing is testing the system as a whole after integrating all the components. It is performed on the entire system to ensure that it is working as end user expected.

Q) What is Acceptance Testing?

Acceptance testing is the testing done in the real time environment by the intended audience or business representatives.

Q) What is Alpha Testing?

A cross-section of potential users and members are invited to use the system. Developers observe the users and note problems.

Q) What is Beta Testing?

Beta testing or field testing sends the system to a cross-section of users who install it and use it under real-world working conditions. The users send records of incidents with the system to the development organizations where the defects are fixed.

Q) What is verification?
Verification is the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. It takes care of set of activities to address the question "**Are we building the product right**?"

Q) What is Validation?
Validation is the process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. It takes care of set of activities to address the question "**Are we building the right product**?"

What is Sanity Testing?
Sanity testing is done to ensure that basic functionality is working fine and whether it is reasonable to proceed with further testing or not. It can also call as "Build Verification test".

What is Smoke Testing in Software Testing?
Smoke Testing is done to make sure if the build we received from the development team is testable or not.
It helps not to waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

- Assignment – Difference between??

Q) What is Functional Testing?
Functional testing involves testing an applications' functionality and features based on analysis of the requirements specifications. It helps to verify what the system is supposed to do.

Q) What is Non-Functional Testing?

Non-functional testing involves testing the application's non-functional characteristics such as reliability, scalability etc. It helps to determine how well the system works.

Q) What is Re-Testing?
Re-Testing or Confirmation testing is the process of executing test cases that failed the last time when they were run, in order to verify the creativeness of the fixes provided.

Q) What is Regression Testing?
Regression testing is done to ensure that the changes made in the software for fixing defects or to enhance the functionalities does not affect the exiting functionality.

Q) What is the difference between Re-testing and Regression Testing?
Re-Testing is done to verify defect fixes and Regression testing is done to check if the defect fixes have disturbed existing functionality that was working fine before making the changes.

Test Case Development

- Equivalence Partitioning & Boundary Value Analysis
- Use Case Testing
- Testing Review
- Error Guessing

Test Management & Control

Test Estimation

Test Estimation is a management activity which approximates how long a Task would take to complete. Estimating effort for the test is one of the major and important tasks in Test Management.

Resources: Resources are required to carry out any project tasks. They can be people, equipment, facilities, funding, or anything else capable of definition required for the completion of a project activity.

Times : Time is the most valuable resource in a project. Every project has a deadline to delivery.

Human Skills : Human skills mean the knowledge and the experience of the Team members. They affect to your estimation. For example, a team, whose members have low testing skills, will take more time to finish the project than the one which has high testing skills.

Cost: Cost is the project budget. Generally speaking, it means how much money it takes to finish the project.

TEST PLAN

A TEST PLAN is a document describing software testing scope and activities.

A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used

Test Plan Identifier:

Provide a unique identifier for the document.

Introduction:

Provide an overview of the test plan.
Specify the goals/objectives.
Specify any constraints.

Features to be Tested:

List the features of the software/product to be tested.
Provide references to the Requirements and/or Design specifications of the features to be tested

Features Not to Be Tested:

List the features of the software/product which will not be tested.
Specify the reasons these features won't be tested.

Approach:

Mention the overall approach to testing.

Item Pass/Fail Criteria:

Test Environment:

Schedule:

Provide a summary of the schedule, specifying key test milestones, and/or provide a link to the detailed schedule.

Responsibilities:

List the responsibilities of each team/role/individual.

Risks:

List the risks that have been identified.

Assumptions and Dependencies:

List the assumptions that have been made during the preparation of this plan.
List the dependencies.

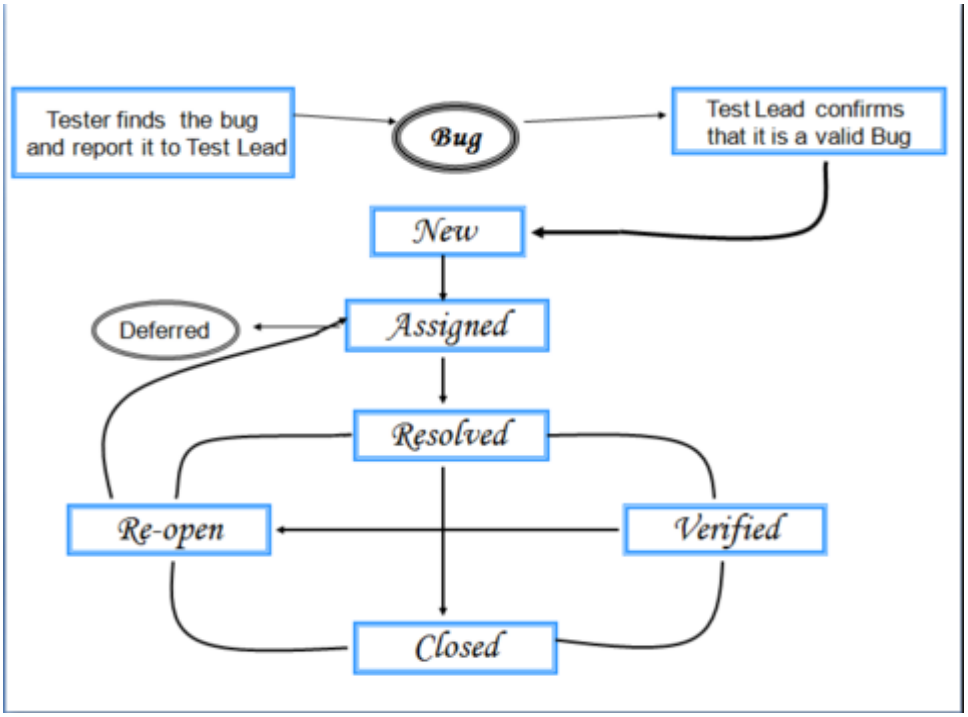
Defects

- Defects

- Defect Life Cycle

DEFECT: It can be simply defined as a variance between expected and actual. In other words Defect is the difference between expected and actual result in the context of testing. It is the deviation of the customer requirement.

Differentiate Error, Defect, and Failure?
Terminology vary from one phase to another in Software Application Life Cycle
Software Application Life Cycle has 3 phases;
i) Development phase
in Development phase if developers find any mismatch they call it as Error or mistake
ii) Testing phase
In Testing phase if Testers find any mismatch they call it as Defect or Bug or Fault
iii) Production phase
In Production phase if Customers or End Users find any mismatch they call it as Failure
Defect Life Cycle



List of Popular Defect Tracking Tools:

- ✓ JIRA
- ✓ HP ALM
- ✓ Bugzilla
- ✓ Mantis

Advanced Stuff

• Complete Web Application Testing Checklist

• Practical Tips and Tricks to Create Test Data

Importance of Checklist for Testing

1) Maintaining a standard repository of reusable test cases for your application will ensure that most common bugs will be caught more quickly.

2) A checklist helps to complete writing test cases quickly for new versions of the application.

3) Reusing the test cases help to save money on resources to write repetitive tests.

4) Important test cases will be covered always, thereby making it almost impossible to forget.

5) The testing checklist can be referred by developers to ensure if the most common issues are fixed in the development phase itself.

Back End Testing

- Importance of Backend
- SQL Statements
- Joins