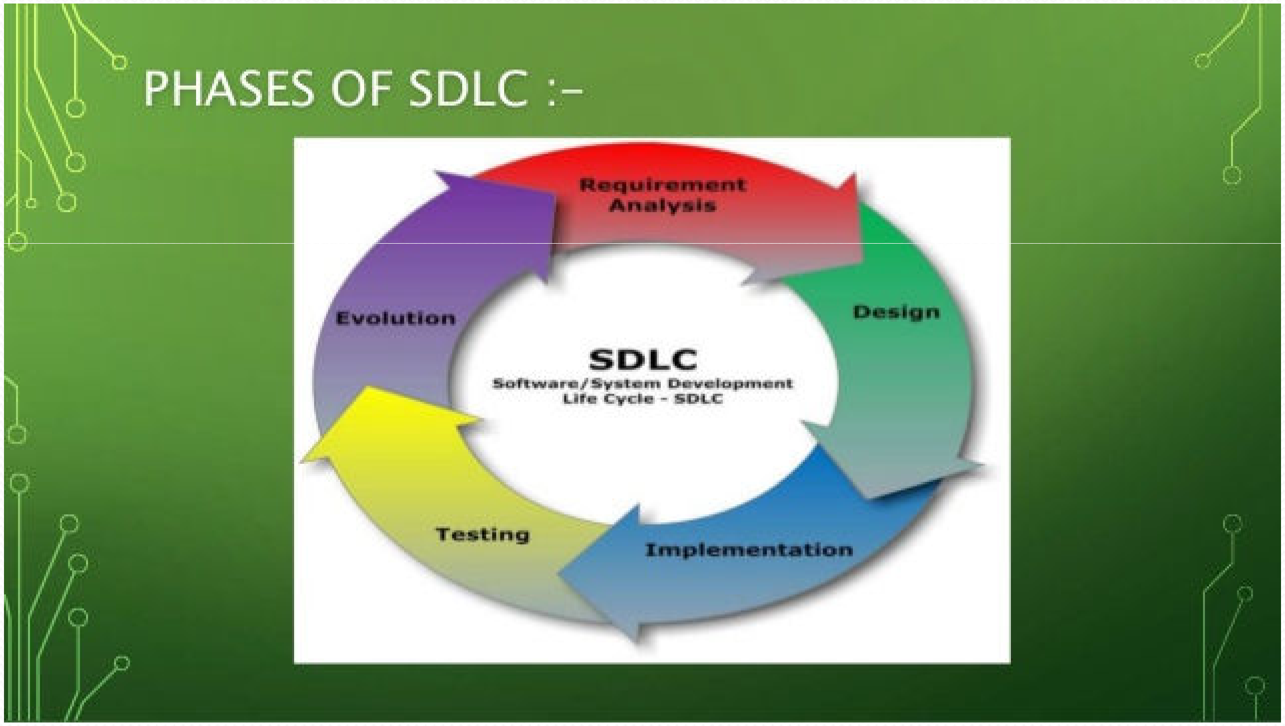
Software Testing

STLC-Software Testing Life Cycle

SDLC



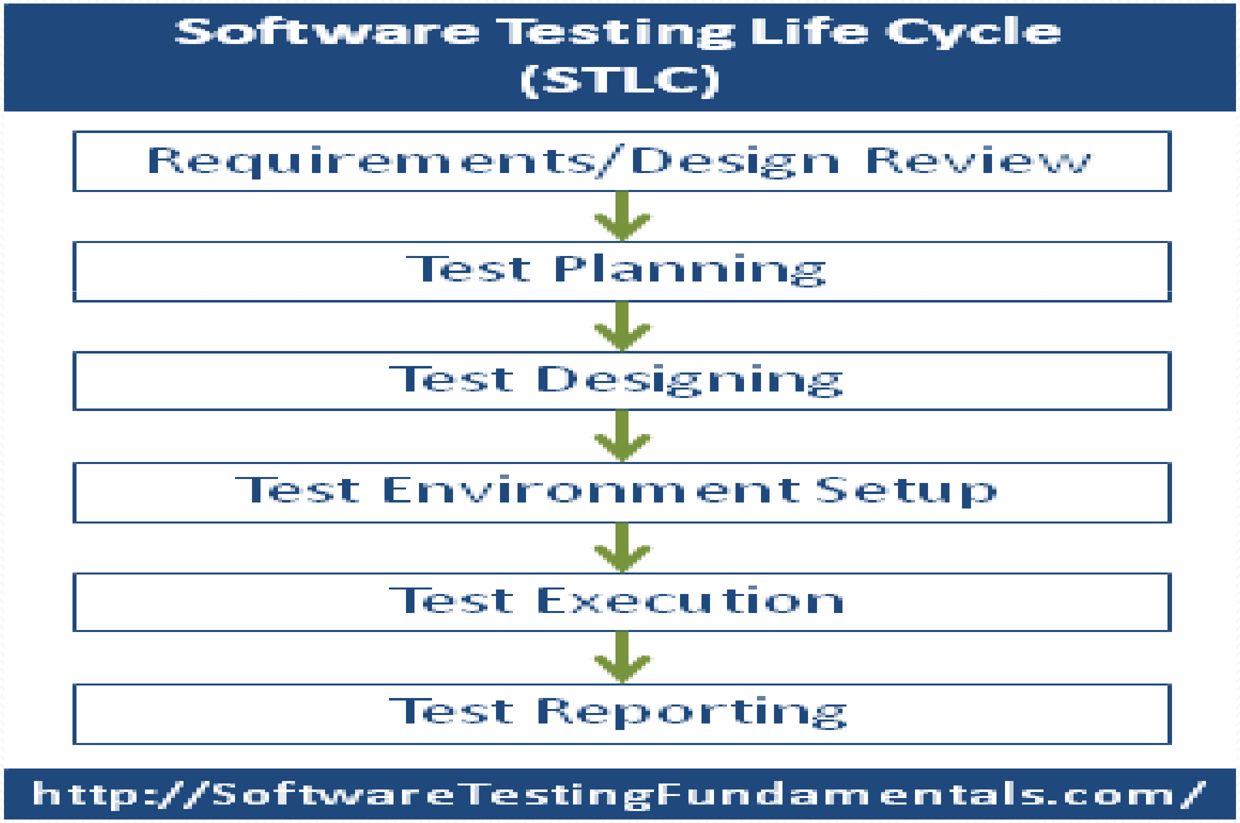
PHASES OF SDLC :-

Software Testing Lifecycle

Software Testing Life Cycle (STLC) defines the steps/ stages/ phases in testing of software. However, there is no fixed standard STLC in the world and it basically varies as per the following:

- Software Development Life Cycle
- Whims of the Management

Phases of STLC



Software Testing Life Cycle (STLC)

Requirements/Design Review

Test Planning

Test Designing

Test Environment Setup

Test Execution

Test Reporting

http://SoftwareTestingFundamentals.com/

Phases of STLC

Requirements Analysis: You review the software requirements/ design.

Test Plan: Once you have gathered a general idea of what needs to be tested, you ‘plan’ for the tests.

Test Design: Develop test cases for each part that needs to be tested based on the test requirements and design of the software.

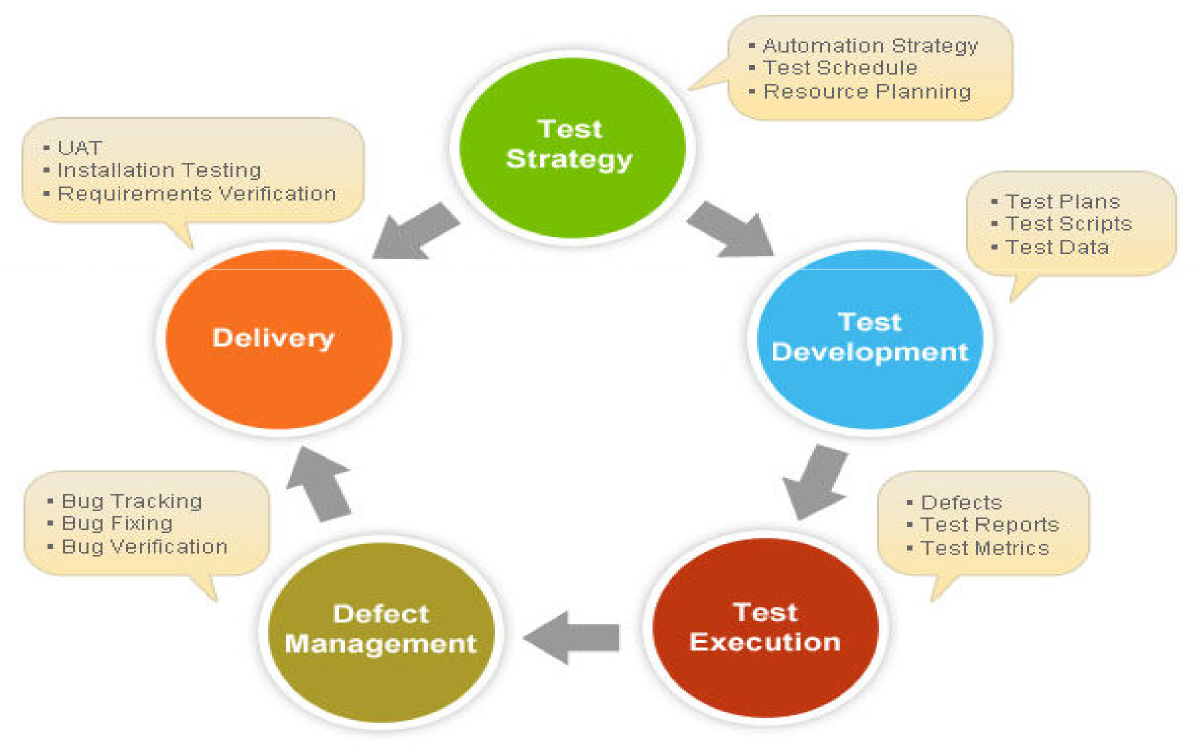
Cont..

Test Environment Setup: You setup the test environment (server/ client/ network, etc) with the goal of replicating the end-users' environment.

Test Execution: You execute your Test Cases/ Scripts in the Test Environment to see whether they pass.

Test Reporting: You prepare various reports like test evaluation summary, and a RTM if needed.

STLC deliverables



Component Testing

Component Testing: Component testing is a method where testing of each component in an application is done separately.

Example: Consider two web pages of a website. In one of the web pages there are certain fields like username, address, mobile no. etc in which data has to be entered. In the other (second) web page also there are certain fields which carry forward the data from the first page. Testing the functionality of these individual pages is called Component Testing.

Unit Testing

Unit Testing: Individual components are tested to ensure that they operate correctly. Each component is tested independently without other system components.

Unit tests are written and executed by software developers to make sure that code meets its design and requirements and behaves as expected.

Method Used for unit testing: White Box Testing method is used for executing the unit test.

When Unit testing should be done?

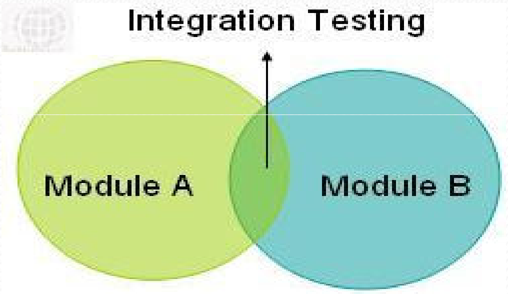
Unit testing should be done before Integration testing.

By whom unit testing should be done?

Unit testing should be done by the developers.

Integration Testing

Integration Testing: Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions.



For example you have to test the keyboard of a computer then it is a unit testing but when you have to combine the keyboard and mouse of a computer together to see its working or not than it is the integration testing. So it is prerequisite that for performing integration testing a system must be unit tested before.

System Testing

System Testing: System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards.

- System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose.
- System testing is carried out by testers.
- System testing should investigate both functional and non-functional requirements of the testing.

Functional Testing

Testing the application against business requirements. Functional testing is done using the functional specifications provided by the client or by using the design specifications like use cases provided by the design team.

Ex: Unit Testing, Integration Testing, Regression Testing

Requirement-based testing: In this type of testing the requirements are prioritized. This will ensure that the most important and most critical tests are included in the testing effort.

Business-process-based testing: In this type of testing the scenarios involved in the day-to-day business use of the system are described. It uses the knowledge of the business processes. For example, a personal and payroll system may have the business process along the lines of:

someone joins the company, employee is paid on the regular basis and employee finally leaves the company.

Non-Functional Testing

In non-functional testing the quality characteristics of the component or system is tested. Non-functional refers to aspects of the software that may not be related to a specific function or user action such as scalability or security.

Eg. Performance Testing, Security testing, Load testing

Non-functional testing is also performed at all levels like functional testing.

Performance Testing, refers to testing done to analyze and improve the performance of an application. The focus here is on optimization of resource consumption by analyzing data collected during testing. Performance Testing to a certain extent should be done by developers, or separate performance team.

In some organizations, the performance team is a part of the QA function.

Load Testing refers to the kind of testing usually done by QA/Perfomance test team to ensure that the application can handle a certain load level. Criteria are set to ensure that releases of a product meet certain conditions like the number of users they can support while delivering a certain response time.

Scalability Testing refers to performance testing that is focused on understanding how an application scales as it is deployed on larger systems and/or more systems or as more load is applied to it. The goal is to understand at what point the application stops scaling and identify the reasons for this. Scalability testing can be viewed as a kind of performance testing. focuses on the performance of your Web sites, hardware and software products.

Acceptance Testing

Acceptance testing: Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release i.e whether the system meets the specified business needs.

Example: When a mobile application is set to release, all the functionality is tested to match the specified business needs and requirements.

Alpha and beta testing

Alpha Testing: Alpha testing performed by Testers, who are usually internal employees of the organization, at the developer’s location.

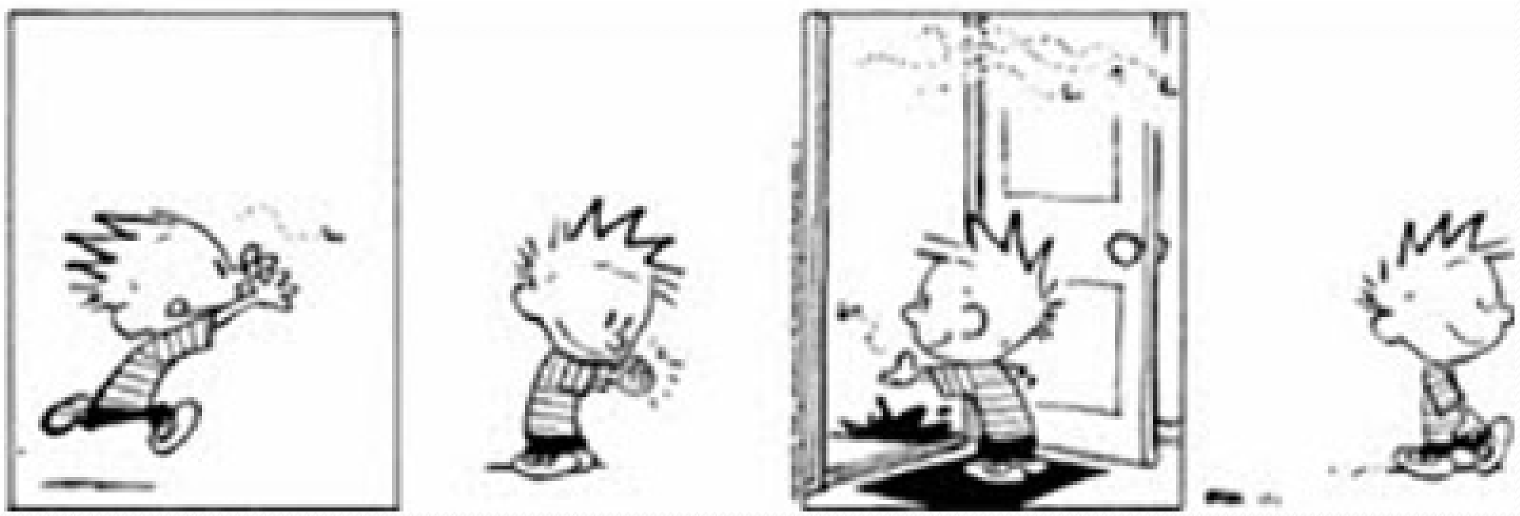
Alpha testing involves both the white box and black box techniques.

Beta Testing: Beta testing is performed by Clients or End users, who are not employees of the organization, at Client location. Beta Testing typically uses black box testing.

Regression testing

- Regression testing: Regression testing is a type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes, have been made to them.

Regression:
"when you fix one bug, you introduce several newer bugs."



Regression testing example

For Example there are three Modules in the Project named Admin Module, Personal Information, and Employment Module and suppose bug occurs in the Admin Module like on Admin Module existing User is not able to login with valid login credentials so this is the bug.

Now Testing team sends the above - mentioned Bug to the Development team to fix it and when development team fixes the Bug and hand over to Testing team then testing team checks that fixed bug does not affect the remaining functionality of the other modules (Admin, PI, Employment) and also the functionality of the same module (Admin) so this is known as the process of regression testing done by Software Testers.

Validation

Validation: Validation is the process of evaluating the final product to check whether the software meets the business needs.

Following item is evaluated during Validation: Actual product or Software under test.

Ex: Smoke testing, Functional testing, Regression testing

Verification

Verification: Evaluates the intermediary products to check whether it meets the specific requirements of the particular phase.

Intermediary products – requirement specifications, design documents, ER diagrams etc.
Following items are evaluated during Verification: Plans, Requirement Specifications, Design Specifications, Code, Test Cases etc,

Validation is the process of checking whether the specification captures the customer's needs, while verification is the process of checking that the software meets the specification.

User Story

A user story is a document in Agile software development terms to capture a description of a software feature from an end-user perspective.

The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement