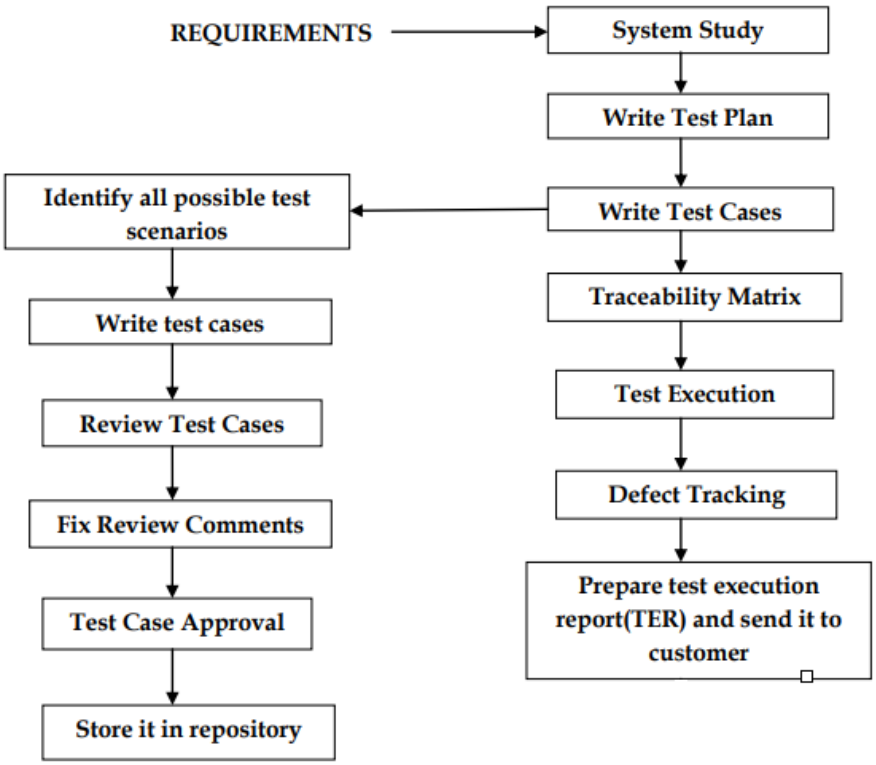
**SOFTWARE TEST LIFE CYCLE (STLC)**

- Testing itself has many phases i.e is called STLC.
- STLC is part of SDLC
- Defect Life Cycle is a part of STLC

Software Testing Life Cycle (STLC) is a process used to test software and ensure that quality standards are met. Tests are carried out systematically over several phases. During product development, phases of the STLC may be performed multiple times until a product is deemed suitable for release.
The testing process is executed in a well-planned and systematic manner. All activities are done to improve the quality of the software product.



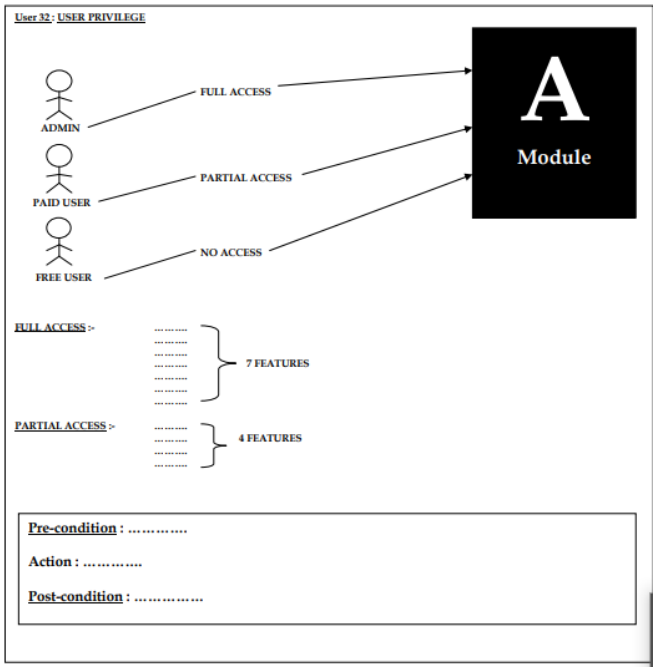

**REQUIREMENTS COLLECTION / SYSTEM STUDY**

**Requirement is the input for testing.**
In this phase, the tester analyses the requirement document of SDLC (Software Development Life Cycle) to examine requirements stated by the client. After examining the requirements, the tester makes a test plan to check whether the software is meeting the requirements or not

- The requirements can be in any of the following forms,
- CRS (Customer Requirement Specification)

- SRS (System Requirement Specification)
- FS (Functional Specification)
- If we don"t have requirements and if we are given only the application, then we do exploratory testing
- Use case

**Eg of Use Case:** Use case is a pictorial representation of requirements. It explains how the end user interacts with the application. It gives all possible ways of how the end user uses the application.




The above figure shows a sample use case of one of the requirements in the CRS.
For the module A of the application, there are 7 features.
Admin has access to all the 7 features.
For a paid user – access to 4 features, For a free user – no access to any of the features.

**Ex – for admin**
Precondition – admin must be created
Action – login as paid user , Post condition – 4 features must be there

**Ex – for free user**
Precondition – free user must be created
Action – login as free user , Post condition – no features

Customer gives the CRS for the application to be developed. The development team write the use case for the CRS and the use case is sent to the customer for review. If the customer

approves it, then the approved use case is sent to the development team for design and coding. The approved use case is also sent to the testing team who start writing test plan and later on start writing test cases for the features of the application

**Entry Criteria**—documented requirements, acceptance criteria, and intended product architecture.

**Deliverables:** List of all testable requirements, Automation feasibility report (if applicable)

**Test Plan –** is a document which derives all future activities of the project. All future testing activities are planned and put into document and this document is known as Test Plan.
It contains – number of engineers needed for the project, who should test which feature, how the defects must be communicated to the development team, when we should start and finish writing test cases, executing test cases, what are the types of testing we use to test for the application etc

**Entry Criteria:** Requirements Documents

**Deliverables:** Test Strategy, Test Plan, and Test Effort estimation document.

**Test Case Development / Write test cases –** we write test cases for each feature. These test cases are reviewed, and after all mistakes are corrected and once the test cases are approved – then they are stored in the test case repository.

- **Entry Criteria:** Requirements Documents (Updated version of unclear or missing requirement)
- **Deliverables:** Approved Test cases, Test Scripts (if automation), Test data.

**Traceability Matrix** – it is a document which ensures that every requirement has a test case . Test cases are written by looking at the requirements and test cases are executed by looking at the test cases. If any requirement is missed i.e, test cases are not written for a particular requirement, then that particular feature is not tested which may have some bugs. Just to ensure that all the requirements are converted, a traceability matrix is written.

**Traceability Matrix** is a document which has got the mapping between requirements and test cases. We write TM to make sure that every requirement has at least 1 test case.

Once the missing requirements are identified – we write the test cases for the requirements which we have missed – review it and get it approved – and then store the test cases in the repository and then fill in the name of the test case for which the requirements have been missed

| *TRACEABILITY MATRIX* | |
|---|---|

| Requirement Number | Test Case Name |
|---|---|
| 1 | … |
| 2 | … |
| 3 | |
| 4 | … |
| 5 | … |
| 6 | |
| 7 | … |

For the requirements (3 and 6) for which test cases are not written, the cells are marked in thick borders so that they are distinct and then test cases are written for them. The Traceability Matrix is also known as RTM(Requirement Traceability Matrix) or CRM(Cross Reference Matrix).

**Test Case Execution**
Here, we test the product. We test repeatedly for 40 – 60 cycles. We do all types of testing on the application. Test Execution is the phase where we spend 80% of our time on the project. Only 20% is spent on the remaining stages.

**Entry Criteria:** Requirement, Test Plan document, Test cases, Test data, Test Environment.
**Deliverables:** Test case execution report, Defect report, RTM

| Entry Criteria | Activity | Deliverable |
|---|---|---|
| Requirement Document | Creation of test cases.<br>Execution of test cases.<br>Mapping of test cases according to requirements | Test execution result.<br>List of functions with the detailed explanation of defects. |

**Defect Tracking –** any bug found by the testing team is sent to the development team. This bug has to be checked by the testing team if it has been fixed by the developers.
-how to communicate the defects found during testing to the development team and also how the development team should respond to it. We should also mention the priority of the defect – high, medium, low.

**Test cycle closure/ Test Execution Report :-**

The final STLC phase is test cycle closure. In this stage, the testing team provides a test closure report, which summarizes and communicates its findings with the rest of the team. This report typically includes summaries of the testing work and results, an assessment of the testing and the manager's approval.

During the test cycle closure, the testing team checks its deliverables, which include details relevant to the testing work, such as the test strategy, test case documents, automated test scripts and test results.

Send the Test Execution Report to customer – contains a list of bugs(major, minor and critical), summary of test pass, fail etc and when this is sent, according to the customer – the project is over. TER is prepared after every test cycle and sent to the development team, testing team, management and customer(depends if it is a fixed bid project or time & material bid project). The last TER of the last test cycle is always sent to the customer. And this means that the project is over according to the customer

**Entry Criteria:** Test Case Execution report (make sure there are no high severity defects opened), Defect report

**Deliverables:** Test Closure report, Test metrics

## STLC Phases (Entry and Exit Criteria)

| STLC Phase | Entry Criteria | Exit Criteria | Deliverables |
|---|---|---|---|
| Requirement Analysis | Requirements specification document, Acceptance criteria document, Application architectural document | Signed off RTM, Signed off Automation feasibility report | List of all testable requirements, Automation feasibility report (if applicable) |
| Test Planning | Requirements Documents, Automation feasibility report | Approved Test plan document, Approved Test strategy document, Signed off effort estimation document | Test Strategy, Test Plan, and Test Effort estimation document. |

| | | | |
|---|---|---|---|
| Test Design | Requirements Documents (Updated version of unclear or missing requirement), RTM, Test Plan, Test Estimation Document, Automation Analysis Report | Reviewed and approved test cases, test scripts, test data | Test cases, Test Scripts (if automation), Test data. |
| Test Environment Setup | Test Plan, Test environment setup plan, Smoke Test cases, Test Data | Working test environment setup, Valid test data setup, Successful smoke test | Test Environment. Smoke Test Results. |
| Test Execution | Test Plan document, Test cases, Test Scripts, Test data, Test Environment. | Execute all planned test cases, Log all defects found | Test case execution report, Defect reports, RTM |
| Test Closure | Testing has been completed, Test Case Execution report (make sure there are no high severity defects opened) | Signed off Test Closure report | Test Closure report, Test metrics |