**SOFTWARE TESTING**

1. **What is testing?**

**Testing is a process to check whether the product or software meets the customer requirements or not.**

**Main goal of testing is to produce the high quality software,Without any bugs.**

1. **Types of Testing:**

* **Manual Testing:**
* **Testing of a software is done manually without the use of automated tools or applications that are available in the market.**
* Manual testing is the process in which QA analysts execute tests one-by-one in an individual manner. The purpose of manual testing is to catch bugs and feature issues before a software application goes live.
* **Automation testing:**
* Automated testing is the process in which testers utilize tools and scripts to automate testing efforts.
* Automated testing helps testers execute more test cases and improve test coverage. When comparing manual vs. automation testing,
* Manual takes longer. Automated testing is more efficient.

**Levels of testing (CISA)**

* COMPONENT TESTING
* INTEGRATION TESTING
* SYSTEM TESTING
* ACCEPTANCE TESTING

**STLC – SOFTWARE TESTING LIFE CYCLE**

**What is software testing life cycle?**

* The Software Testing Life Cycle (STLC) is a systematic approach to testing a software application to ensure that it **meets the requirements** and is **free of defects.**
* The STLC is used to ensure that the **software is of high quality, reliable, and meets the needs of the end-users.**

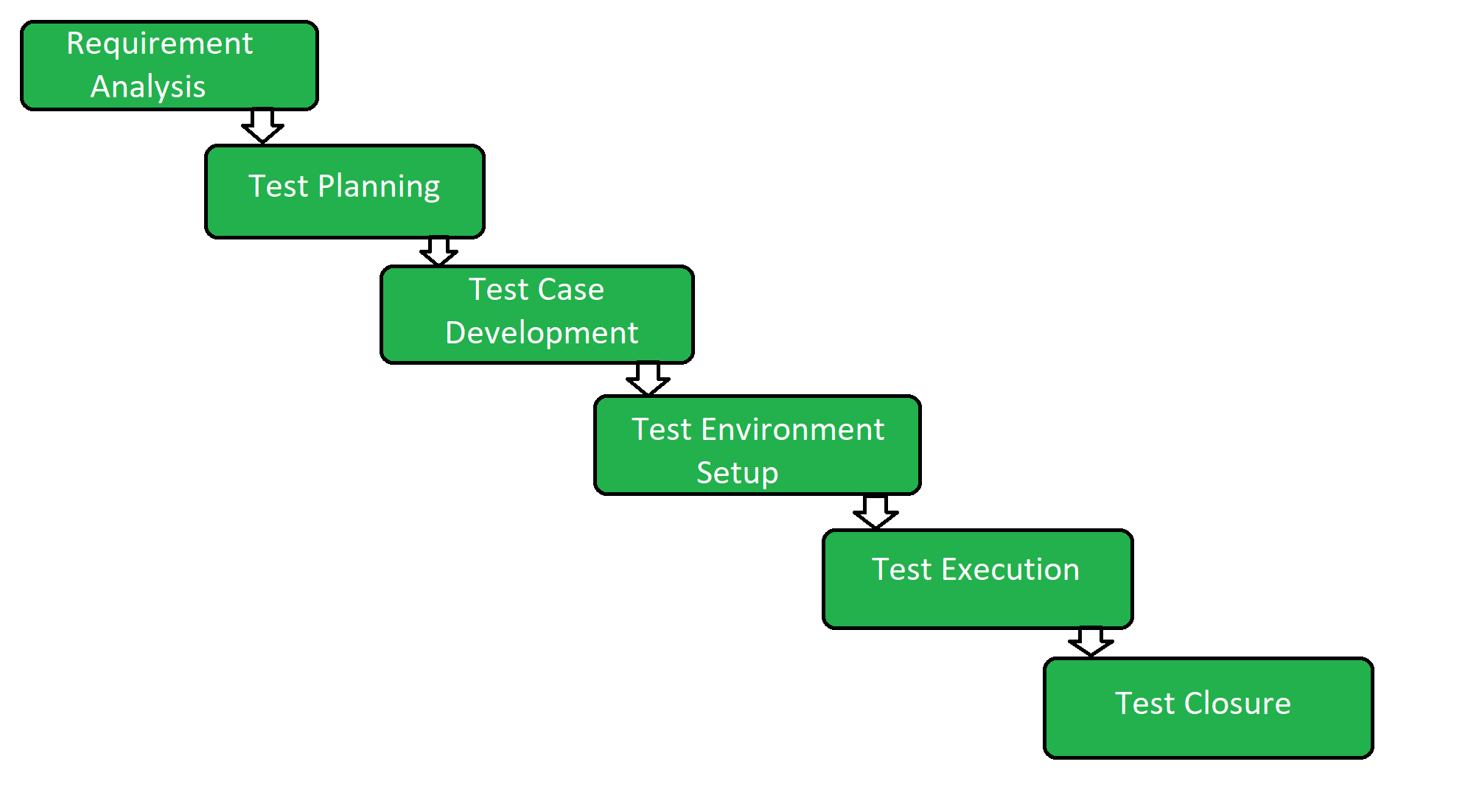
**Main goal of STLC**

* Identify and document **any defects or issues in the software application as early as possible in the development process**.
* This allows for **issues to be addressed and resolved** before the software is released to the public.

**Characteristics of STLC**

* STLC is a fundamental part of the [**Software Development Life Cycle (SDLC**)](https://practice.geeksforgeeks.org/problems/software-development-life-cycle) but STLC consists of only the testing phases.
* STLC starts as soon as requirements are defined or software requirement document is shared by stakeholders.
* STLC yields a step-by-step process to ensure quality software.

**Phases of STLC**

****

1. **Requirement Analysis:**

* Requirement Analysis is the first step of the Software Testing Life Cycle (STLC).
* In this phase quality assurance team understands the requirements like what is to be tested.
* If anything is missing or not understandable then the quality assurance team meets with the stakeholders to better understand the detailed knowledge of requirements.

**The activities that take place during the Requirement Analysis stage include:**

* Reviewing the software requirements document (SRD) and other related documents
* Interviewing stakeholders to gather additional information
* Identifying any ambiguities or inconsistencies in the requirements
* Identifying any missing or incomplete requirements
* Identifying any potential risks or issues that may impact the testing process
* Creating a **requirement traceability matrix (RTM)** to map requirements to test cases
* At the end of this stage, the testing team should have a clear understanding of the software requirements and should have identified any potential issues that may impact the testing process.
* This will help to ensure that the testing process is focused on the most important areas of the software and that the testing team is able to deliver high-quality results.

**2. Test Planning:** Test Planning is the most efficient phase of the software testing life cycle where all testing plans are defined. In this phase manager of the testing, team calculates the estimated effort and cost for the testing work. This phase gets started once the requirement-gathering phase is completed.

**The activities that take place during the Test Planning stage include:**

* Identifying the testing objectives and scope
* Developing a test strategy: selecting the testing methods and techniques that will be used
* Identifying the testing environment and resources needed
* Identifying the test cases that will be executed and the test data that will be used
* Estimating the time and cost required for testing
* Identifying the test deliverables and milestones
* Assigning roles and responsibilities to the testing team
* Reviewing and approving the test plan

At the end of this stage, the testing team should have a detailed plan for the testing activities that will be performed, and a clear understanding of the testing objectives, scope, and deliverables. This will help to ensure that the testing process is well-organized and that the testing team is able to deliver high-quality results.

**3. Test Case Development:**The test case development phase gets started once the test planning phase is completed. In this phase testing team notes down the detailed test cases. The testing team also prepares the required test data for the testing. When the test cases are prepared then they are reviewed by the quality assurance team.

**The activities that take place during the Test Case Development stage include:**

* Identifying the test cases that will be developed
* Writing test cases that are clear, concise, and easy to understand
* Creating test data and test scenarios that will be used in the test cases
* Identifying the expected results for each test case
* Reviewing and validating the test cases
* Updating the requirement traceability matrix (RTM) to map requirements to test cases

At the end of this stage, the testing team should have a set of comprehensive and accurate test cases that provide adequate coverage of the software or application. This will help to ensure that the testing process is thorough and that any potential issues are identified and addressed before the software is released.

**4. Test Environment Setup:** Test environment setup is a vital part of the STLC. Basically, the test environment decides the conditions on which software is tested. This is independent activity and can be started along with test case development. In this process, the testing team is not involved. either the developer or the customer creates the testing environment.

**5. Test Execution:**After the test case development and test environment setup test execution phase gets started. In this phase testing team starts executing test cases based on prepared test cases in the earlier step.

**The activities that take place during the test execution stage of the Software Testing Life Cycle (STLC) include:**

* **Test execution:** The test cases and scripts created in the test design stage are run against the software application to identify any defects or issues.
* **Defect logging:** Any defects or issues that are found during test execution are logged in a defect tracking system, along with details such as the severity, priority, and description of the issue.
* **Test data preparation:** Test data is prepared and loaded into the system for test execution
* **Test environment setup:** The necessary hardware, software, and network configurations are set up for test execution
* **Test execution:**The test cases and scripts are run, and the results are collected and analyzed.
* **Test result analysis:** The results of the test execution are analyzed to determine the software’s performance and identify any defects or issues.
* **Defect retesting:** Any defects that are identified during test execution are retested to ensure that they have been fixed correctly.
* **Test Reporting:** Test results are documented and reported to the relevant stakeholders.

It is important to note that test execution is an iterative process and may need to be repeated multiple times until all identified defects are fixed and the software is deemed fit for release.

**6. Test Closure:**Test closure is the final stage of the Software Testing Life Cycle (STLC) where all testing-related activities are completed and documented. The main objective of the test closure stage is to ensure that all testing-related activities have been completed and that the software is ready for release.

At the end of the test closure stage, the testing team should have a clear understanding of the software’s quality and reliability, and any defects or issues that were identified during testing should have been resolved. The test closure stage also includes documenting the testing process and any lessons learned so that they can be used to improve future testing processes

Test closure is the final stage of the Software Testing Life Cycle (STLC) where all testing-related activities are completed and documented. The main activities that take place during the test closure stage include:

* **Test summary report:** A report is created that summarizes the overall testing process, including the number of test cases executed, the number of defects found, and the overall pass/fail rate.
* **Defect tracking:** All defects that were identified during testing are tracked and managed until they are resolved.
* **Test environment clean-up:**The test environment is cleaned up, and all test data and test artifacts are archived.
* **Test closure report:** A report is created that documents all the testing-related activities that took place, including the testing objectives, scope, schedule, and resources used.
* **Knowledge transfer:** Knowledge about the software and testing process is shared with the rest of the team and any stakeholders who may need to maintain or support the software in the future.
* **Feedback and improvements:** Feedback from the testing process is collected and used to improve future testing processes

It is important to note that test closure is not just about documenting the testing process, but also about ensuring that all relevant information is shared and any lessons learned are captured for future reference. The goal of test closure is to ensure that the software is ready for release and that the testing process has been conducted in an organized and efficient manner.

DIFFERENCE BETWEEN BUG, DEFECT, ERROR, FAILURE

### What is a Bug?

In [**software testing**](https://qacraft.com/)**,** a bug is a deviation from the customer requirement, in simple language, we can say deviation between the expected result and the actual result in an application or in a module that is found by the [**testing team**](https://qacraft.com/qacraft-team/) during the testing period.

### What is a Defect?

If the functionality of an application is not working as per the customer’s requirement is known as a defect, It is found during the development phase while unit testing. Giving wrong input may lead to a defect or Any code error may lead to a defect.

### What is an Error?

An error in software testing refers to a slip-up, misunderstanding, or mistake made by a software engineer. In the category of developer, we include software engineers, analysts, programmers, and testers. For example, a developer may misunderstand a design notation, or a programmer might type a variable name incorrectly – leading to an Error. It is generated because of the incorrect login, loop, or syntax.

### What is a Failure?

Once the software is completed and delivered to the customer and if the customer found any issue in the software then it is the condition of failure of the software.

In other words, if an end user finds an issue in the software then that particular issue is called a failure.

## Difference between Bug and Defect

|  |  |
| --- | --- |
| **BUG** | **DEFECT** |
| **A bug is a deviation from the customer’s requirement.** | **The functionality of an application not working as per the customer’s requirement is known as a defect.** |
| **The bug is found by the testing team.** | **The defect is found by the development team.** |
| **Reason Behind Bug:**   * **Missing Coding** * **Wrong Coding** * **Extra Coding** | **Reason Behind Defect:**   * **Giving wrong input may lead to a defect.** * **Any code error may lead to a defect.** |
| **The following list includes different types of bugs:**   * **Logic bugs** * **Algorithmic bugs** * **Resource bugs** | **The following list includes different types of Defect: Based on priority:**   1. **High** 2. **Medium** 3. **Low**   **And based on the severity:**   * **Critical** * **Major** * **Minor** * **Trivial** |

**Conclusion:**

**A simple diagram makes you understand.**



**BUG LIFE CYCLE:**

**What is bug lifecycle?**

**Defect Life Cycle** or Bug Life Cycle in software testing is the specific set of states that defect or bug goes through in its entire life. The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.

**Defect Status**

**Defect Status** or Bug Status in defect life cycle is the present state from which the defect or a bug is currently undergoing. The goal of defect status is to precisely convey the current state or progress of a defect or bug in order to better track and understand the actual progress of the defect life cycle.

**Defect States Workflow**

The number of states that a defect goes through varies from project to project. Below lifecycle diagram, covers all possible states

* **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
* **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
* **Open:** The developer starts analyzing and works on the defect fix
* **Fixed:** When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
* **Pending retest**: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”
* **Retest**: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
* **Verified:** The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”
* **Reopen**: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.
* **Closed**: If the bug is no longer exists then tester assigns the status “Closed.”
* **Duplicate**: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”
* **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
* **Deferred**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs
* **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.



**Defect/Bug Life Cycle Explained**



* 1. Tester finds the defect
  2. Status assigned to defect- New
  3. A defect is forwarded to Project Manager for analyze
  4. Project Manager decides whether a defect is valid
  5. Here the defect is not valid- a status is given “Rejected.”
  6. So, project manager assigns a status **rejected**. If the defect is not rejected then the next step is to check whether it is in scope. Suppose we have another function- email functionality for the same application, and you find a problem with that. But it is not a part of the current release when such defects are assigned as a **postponed or deferred**status.
  7. Next, the manager verifies whether a similar defect was raised earlier. If yes defect is assigned a status **duplicate**.
  8. If no the defect is assigned to the developer who starts fixing the code. During this stage, the defect is assigned a status **in- progress.**
  9. Once the code is fixed. A defect is assigned a status **fixed**
  10. Next, the tester will re-test the code. In case, the[Test Case](https://www.guru99.com/test-case.html)passes the defect is **closed.** If the test cases fail again, the defect is **re-opened** and assigned to the developer.

11. Consider a situation where during the 1st release of Flight Reservation a defect was found in Fax order that was fixed and assigned a status closed. During the second upgrade release the same defect again re-surfaced. In such cases, a closed defect will be **re-opened.**

**------------------------------------\*\*\*--------------------------------------------**