Severity:

- It is an indicator of the impact of the bug/defect on the software application called severity.

Or

- Impact of the bug/defect on customer business is called severity.
- Severity can be categorised as below:
    1. Critical
    2. Major
    3. Minor
    4. Low

1.Critical:

- This defect/bug indicates complete shut-down of the process, nothing can proceed further.
- A defect/bug that has completely blocked the functionality of an application where the user or the tester is unable to proceed or test anything. If the whole application is inaccessible or down because of a defect/bug, such a defect/bug is categorised as a critical defect/bug.
- When the login screen of an application is not working and the user is unable to login, the whole application becomes inaccessible to the user.

2.Major:

- When a bug isn't affecting the whole application but still prevents major functionalities of a system from working, it becomes a Major bug/defect.
- There won't be a complete shut-down of the system (which would be the case for a critical defect). Even so, it will prevent major, sometimes even basic functionalities of the application from working.
- Example:On Flipkart users are not able to add anything in the Cart.

3.Minor:

- The behavior of an application is not as expected, but this does not affect functionality.
- Usually, minor severity defects have a workaround, so they may not block a functionality completely (unlike major severity defects where there is no workaround).
- Sometime when bug occurring frequency is less then that bugs also categorised in minor
- The download link in the Help section of an application is not working. However, the user is still able to read the document online.

4.Low:

- Defects/bugs of cosmetic nature that don't affect the application functionality or UX directly. But they are valid bugs/defects nonetheless.
- For instance, spelling mistakes on the webpage. These are valid defects/bugs, but they can wait to be fixed since they're not affecting application functionality.
- Such minor defects/bugs can wait to be fixed until the next release because they are not restricting the functionality of the application.

Priority:

- Importance giving to the bug to fix it is called as priority.

or

- Priority is important for fixing the bug or which bug to be fixed first or how soon the bug should be fixed.
- Priority is defined as the order in which the bug should be resolved
- The Priority status is set based on end users requirement.
- Priorities can be categorised as below:

1.High
2.Medium
3.Low

1.High:

- It has a major impact on the customer application, and it has to be fixed first.
- These bugs may affect the whole application
- Usually, a high severity means a high priority as well. But this is not always the case.

2.Medium:

- The defects/bugs which don’t affect business and customers typically get Medium priority.
- They are not as urgent as the high priority defects/bugs and can be fixed when the development team has the bandwidth to take them up.
- Such bugs can be fixed either in the same release or the next.

3.Low:

- The defects/bugs which have the least priority for getting fixed, they are fixed after all the high and medium priority defects are fixed.
- The flow should be fixed if there is time, but it can be deferred with the next release.

High Severity and High Priority:

- In the flipkart application if a user is not able to search the product or not able to add the product to cart then it comes under the high severity and high priority.

High Severity and Low Priority:

- Privacy link is not opening
- If user not able to download yearly report that user is not using regularly, user use it’s once in year only

Low Severity and Low Priority:

- Alignment is not proper
- Spelling mistake on privacy page

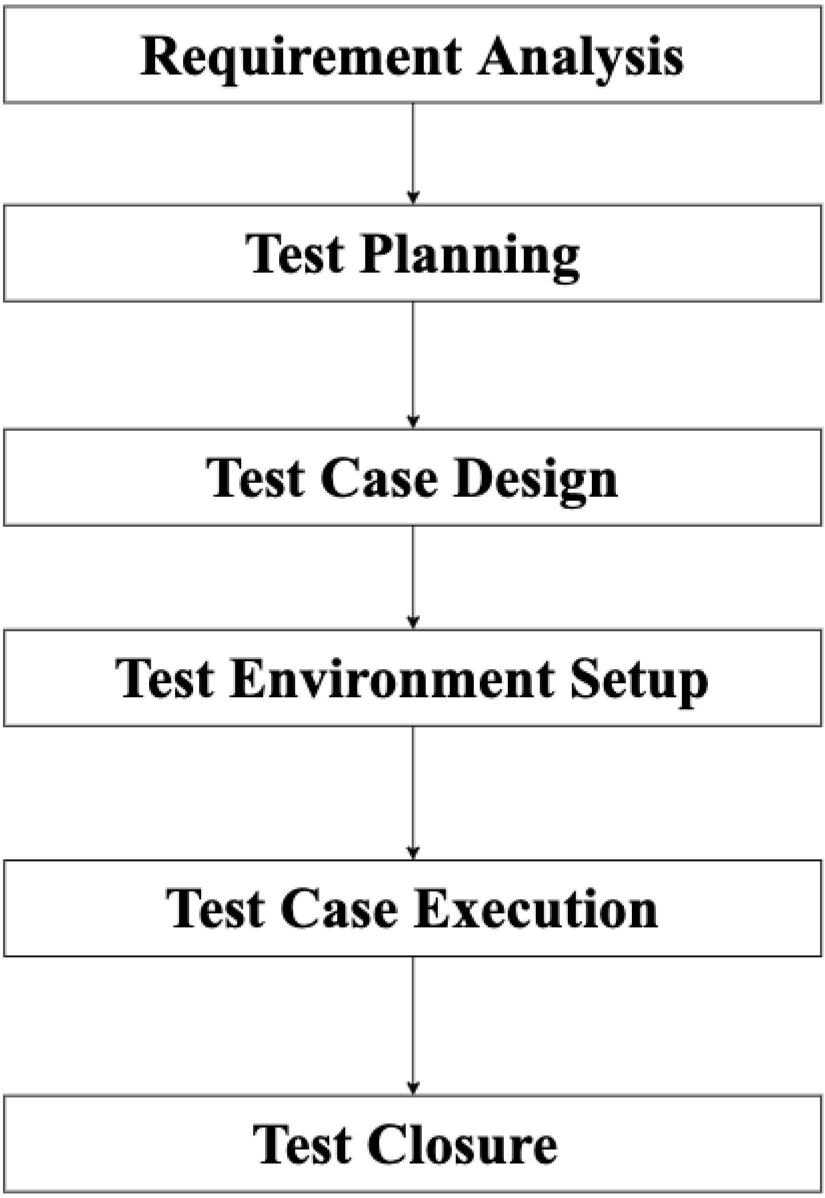
Low Severity and High Priority:

- Customer logo is not proper or spelling mistake in logo

STLC:

- STLC stands for Software Development Life Cycle
- STLC is a sequence of different activities performed by a testing team to ensure the quality of the software or product.
- STLC is a fundamental part of Software Development Life Cycle (SDLC) but STLC consists of only the testing phases.
- STLC starts as soon as requirements are defined or software requirement documents are shared by stakeholders.

Phases in STLC:



Requirement Analysis:

- Requirement Analysis is the first step of Software Testing Life Cycle (STLC)
- In this phase the 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 product owner/stakeholders to better understand the detailed knowledge of requirements.
- This phase helps to identify whether the requirements are testable or not. If any requirement is not testable, the test team can communicate with product owner and client.

Test Planning:

- Test Planning is most efficient phase of software testing life cycle where all testing plans are defined.
- In this phase QA manager/Lead make a plan for testing and it include Testing schedules,Testing Strategy,Job Allocation, Resource allocation and estimation.
- This phase gets started once the requirement gathering phase is Completed.

Test Case Design:

- The test case development phase gets started once the test planning phase is completed.
- In this phase testing team note down the detailed test cases
- A test case is set of actions executed to verify particular feature or functionality of your software application.
- When the test cases are prepared then they are reviewed by quality assurance team.

Test Environment Setup:

- Setup of the test environment is an independent activity and can be started along with Test Case Development.
- This is an essential part of the manual testing procedure as without environment testing is not possible.
- Environment setup requires a group of essential software and hardware to create a test environment.● The test environment setup is done based on the hardware and software requirement list. In some cases, the test team may not be involved in this phase

Test Execution:

- After the test case development and test environment setup test execution phase gets started.
- In this phase testing team start executing test cases based on prepared test cases in the earlier step.
- While executing the test cases tester found bugs and assign that bugs to developer
- After fixing the bug tester will do retest and after fixing all at the end tester will perform regression testing to check impact of that bug on application.

Test Closure:

- Test closure report prepared by Test Lead/QA Manager.
- Test closure report includes the No of test cases designed, Pass percentage, fail Percentage, Bug Percentage.
- Test lead will share this report to the client.
- Test Closure is a document that gives a summary of all the tests conducted
- Test closure report includes the No of test cases designed, Pass percentage, fail Percentage, Bug Percentage.
- Test lead will share this report to the client.