

- If you work with Waterfall methodology, you should read the SRS document several times and analyze it to make sure you understand the client's expectation.
- You should define "what" to be tested.
- In agile work environment, you should read the user stories instead of SRS.
- You should have a clear understanding of customer's need and what to be tested.
- Thus, you can test the software by comparing expected result with the actual result.

- Test plan is a document which consists of test schedules, testing tools, resources of the project, testing area and testers.
- In waterfall, team lead or a tester writes the test plan.
- All testers have access to test plan but the team leader has most of the responsibility.
- Test plan is written for whole project.
- In agile, team lead or testers write the test plan.
- Test plan is not written for whole project, just written for each release and for regression testing or team might not have any test plan document. They may talk verbally about the what type of testing will be done, testing tools.

- This document is most often prepared by manual testers if the team has a manual tester.
- In this document, test data is identified, test steps are written and testing environments are provided.
- It is used for both manual and automation testing.
- Test cases are written in different formats:
- Excel sheet (very old method)
- Issue Tracking software (Jira (most used), bitbucket)
- Gherkin language format in Cucumber

- There are at least 4 environments:
 - Dev Env
- QA/Testing Env
- Staging/preproduction Env
- Production Env
- Developers do Unit&Integration Testing in dev environment.
- QA/testers perform manual testing, functional testing, smoke testing, backend testing in QA environment.
- QA perform regression testing, UAT team perform Alpha testing, client or end users might perform Beta testing in staging environment.
- End user uses the software. No testing is performed in production environment.

- We perform manual and automated execution of test cases.
- We use some tools and IDEs such as Jira,
 Postman for manual testing. It's not about just opening the application and clicking buttons or sth else.
- Pre-conditions (test cases, test data, test script and environment) should be ready to execute the test.
- If the actual result !=
 expected result, execute
 the test script at least 3
 times with different data
 to ensure that there is a
 defect.

- Test reports are generated after each sprint.
- Reports are in Jira, Jenkins etc.
- Reports show who worked on how many user stories, who worked how many hours to do the manual testing, how many task is done, etc.
- These reports are evaluated by team lead, PO, BA and managers.
- Bug status is defined.

Test Execution:

If you find a bug when you perform manual testing, create a bug ticket.

Don't write code for automation until developers fix it. When the bug is fixed, test it manually again. If it passes, then write your code for automation.

Firstly, make sure it's not your code or data's issue, then create a bug ticket.

Bug/Defect Reporting:

Any mismatched functionality found between actual result and expected result is called as Defect/Bug/Issue.

Testers report mismatches as defects to developers through templates or using tools (Jira, ClearQuest, BugZilla).

Bug Life Cycle:

It's a systematic process for dealing with the defects. Bug goes through different stages on its cycle and it has status in each stage.

