Manual Testing

Testing

- Testing the application or a software whether it is working as per the client requirement or not. As a tester we are doing testing in order to find out bugs or defects in an early stage and get them rectified by the developers team.

- Once the developers team fixes the issues then testing team will do a retesting to ensure that the issue is been fixed and there are no further bugs.

Teams involved to develop a software or application

- **BA Team:** Business analyst team will collate the requirements from the client and prepare BRD and FRD documents and give a walk-through on the documents to all the teams.

- **Design Team:** Will prepare HLD and LLD documents and technical documents upon receiving the client requirements.

- **Development Team:** Will build the software as per the client requirements and will perform unit testing.

Testing Team

- Will go through the BRD and FRD documents and understand the functionalities and analyze the requirement.
- List out the test scenarios and prepare test case objectives.(Test Planning Phase).
- Details steps for every test case (Test design phase)
- Once the build is ready we have to start testing the cases (Test execution phase).
- Test Completion stage. We have to take sign off from the client upon completion of test case. (Need to give the status of how many test cases are passes, how many are failed, how many blocked.)
- Client will decide - release (GO/NO GO)

Software Development Life Cycle

- **Requirement gathering and analysis :** We have to go through the requirements analysis and understand the FRD and BRD documents and also we have to go through the functionality of each user stories short description and acceptance criteria and we will be involved in requirements walkthrough sessions by BA and Design team.
- **Designing:** Development and Testing teams will be involved in understanding the HLD and LLD documents and during this design phase development team will be involved in technical part and understanding from the technical documents. Testing team will be focused on the test scenario and test case preparation activities. Upon preparation of Test Case and scenarios we will give walk through sessions to BA, Design, Developers, Project stakeholders.

- **Coding:** Development team will be completely involved in coding and build preparation activities and they will do some kind of unit testing in development environment. Testing team will be focused on the test scenario and test case design effort & test scenario walk through sessions will be planned for BA and Design team, development team and required project stakeholders like that.

- **Testing:** Once build is ready, testing team will focus on smoke testing and sanity testing, progression and regression testing and all these testing activities will be performed and we can share test design & execution summary progress in daily dashboard report. Once SIT, FAT and system testing is completed, the testing team will give the test completion certificate about the overall test execution summary metrics & outstanding defect conclusion to all project team and stake holders. Then project team and stake holders will take decision for release to GO/NO GO discussion and accordingly further release action plans will be taken care.

Software Testing Life Cycle

- It is a sequence of specific activities conducted during the testinf process to ensure software quality objectives are met. STLC involves verification and validation activities.
- Phases involved are:

1. Requirement analysis
2. Test planning
3. Test Design
4. Test Environment setup
5. Test Execution
6. Test Cycle closure

- **Requirement Analysis:** We have to go through the requirement analysis and understand the BRD / FRD documents. QA team may interact with stakeholders to understand the requirements in detail.

- **Test Planning:**

- Understanding the in-scope and out-scope items for the current release.
- Test plan and test strategy document preparation activities.
- Analyzing test estimation size and calculation effort in person hours.
- Discussing on test schedules and test deliverables.
- Resource planning and determining roles and responsibilities.

- **Test Plan:** It is a document which contains the plan related to all testing activities which needs to be done to deliver a quality product. It is prepared after analyzing business requirements of the project.
- It is usually prepared by Test Lead or Senior QA in the Agile team
- The focus of the document is to describe what to test, what cannot be tested, tools used for testing, Environments / infrastructure required to test, staffing and training needs.
- Testing duration, risks and contingencies plan

- **Test Design**:
- Test Scenario preparation
- Querry resolution on requirements by BA/designers team.
- Test scenario walk through sessions to all projects team and taking formal approval on test scenario coverage requirements.
- Peer reviews and external reviews.
- Taking sign off and approvals by BA and clients on test scenarios and test case document.
- Uploading all test cases into JIRA
- Uploading all testing documents in sharepoint by release specific folders.

- This is the phase of STLC where testing team write down the detailed test cases. Along with test cases testing team also prepare the test data if any required for testing.3 Once the test cases are ready then these test cases are reviewed by peer members or QA lead.
- Also the requirement traceability matrix is been prepared. The RTMis an industry accepted format for tracking the requirements where each test case is mapped with the requirement. Using this RTM, we can track backward and forward traceability.

- **Test Environment Setup:**

- Checking and accessing test application URL.
- User setup, credentials checking & roles and responsibilities access to user.
- Test data preparation activities.
- Checking interface connectivity and readiness follow-up activities.

- Test environment decides the software and hardware conditions under which a work product is tested. Test environment set-up is one of the critical aspects of testing process.

- **Developer Code:** Deploy code into server -- view the application in browser by hitting that server URL. Products are stored in database. Application code will access the product and display.

Development team have their own environment --- Server+database. QA deploy the code into QA server -- database.

FAQ

- What is Testing ?

Testing the software or an application whether it is working fine as per the client requirement or not. As a tester we need to find out bugs/issues and inform the developers team to fix the issue at an early stage.

- What is the problem Faced in your project ?

Ans: We have only 2 or 3 QA environment in our project, but multiple teams working on it and everyone tries to push the code into the same environment, which is delaying to us to give sign-off on testing, we have to wait until someone has to complete the testing on QA environment that causes really a delay and we are not able to complete the task on time.

- **Test Execution:**

1. Executing test case in JIRA / HP ALM tool and sharing the daily dashboard test execution summary.
2. Logging defects in QC and defect follow-ups with the development team and other teams.
3. During this phase, the testers will carry out the testing bases on the test plans and test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed.

- **Test Cycle Closure:**

1. Complete all SIT and FAT test execution.
2. Publishing all outstanding defects and over all test execution summary metrics.
3. Getting confirmation on GO/NO Go decision by all the clients for current release deliverables or new features implemented.
4. Publishing test completion summary and taking sign off from client and project team.

- Test closure activities are mostly done after the product is delivered.
- Test closure activities mainly comprises of four types:

1. Ensure test completion (IF any test cases are pending and planning for next sprint)
2. Handover test artifacts (Submitting automation reports to the client)
3. Project retrospectives
4. Archive test work products.

Agile Scrum Methodologies

- Agile is a continuous iteration of development and testing the software in an incremental approach.
- Project release will be planned for short terms. Per month the sprint duration is 2 weeks.
- In Agile model, we are getting all the requirements in the form of user stories with acceptance criteria and story points, later we will also get BRD/FRD from BA's and designers.
- We will start our testing from requirement phase itself. Frequent requirement and design changes will be there in scrum methodology, so we have to update test scenarios and test cases accordingly and we need to give a walkthrough on test scenarios to BA and designers team. Final version of test scenarios and test cases will be updated in share point. All test cases will be uploaded in HP ALM tool.

- Daily we have scrum call meetings and these will be conducted by scrum master in that we will discuss about sprint backlog items, product backlog items and sprint grooming.

- **Sprint Backlog:** It is a list of user story tasks identified by scrum team to be completed in a sprint. During the sprint planning meeting, the team selects some no. of backlog items usually in the form of user stories, and identifies tasks necessary to be completed in each user story.
- **Sprint:** A product is built in a series of iteration called sprints that break down big, complex projects into bite sized pieces. Release time divided into no. of iterations.

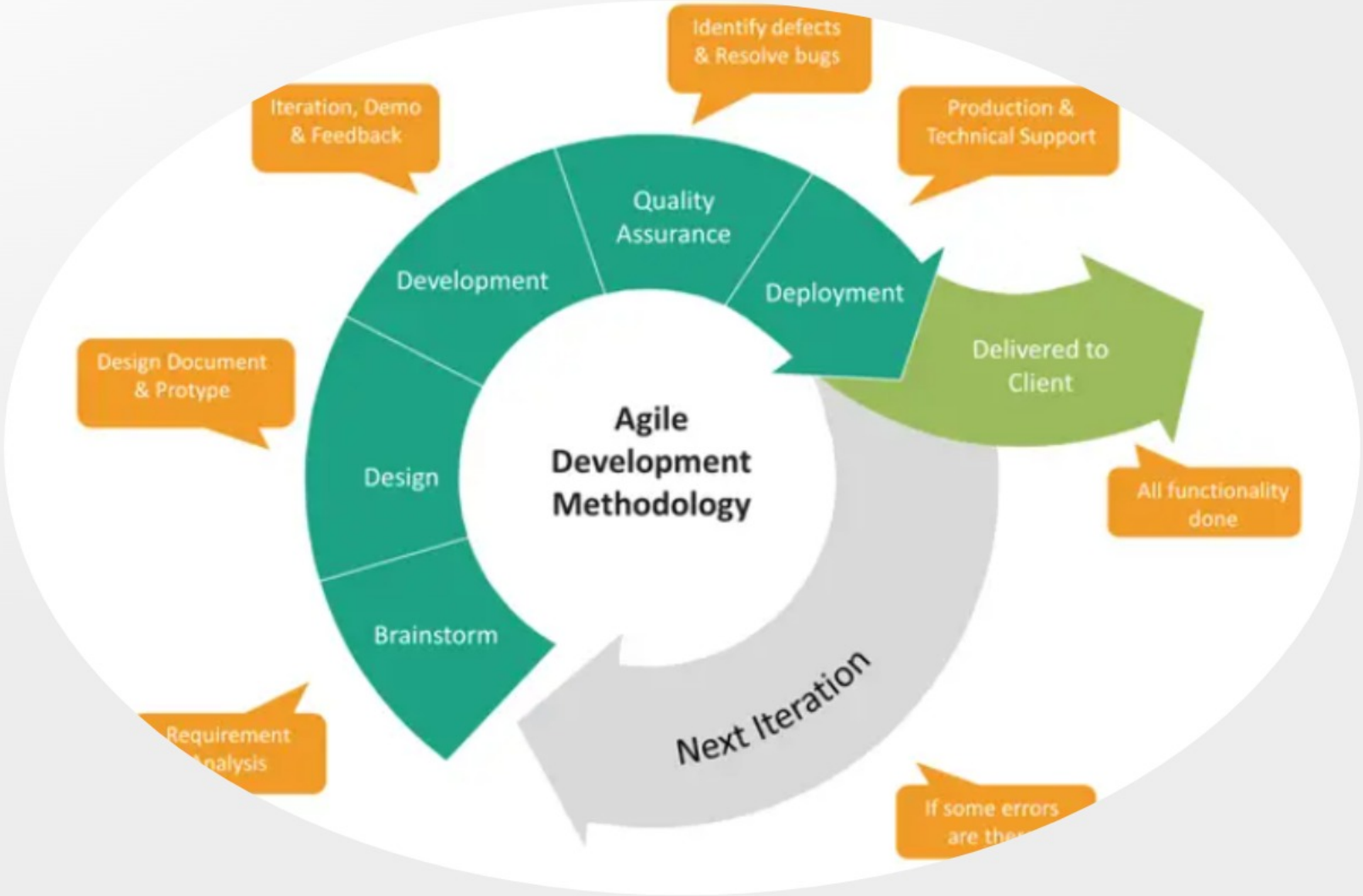
- **Sprint Planning:** If sprint would start from tomorrow then the sprint planning meeting will happen today before the sprint starts and would discuss on User Stories, No. of people required to handle the project, Time to complete the project.
- One week before the project ends backlog grooming is done to make user stories. In middle of the sprint backlog grooming for next sprint is discussed.

- **Product Backlog:** It is a list of prioritized deliverables (new features) that should be implemented as a part of project or product development. Product owner owns product backlogs, he is the one who prioritized it based on customers feedback or business value. Product owner prepare and maintains product backlog. It is prioritized by product owner, and anyone can add to it with approval of product owner.

- **Grooming:** It is a meeting of the scrum team in which the product backlog items are discussed and the next sprint planning is prepared. Product grooming is critical in product management because it means keeping the backlog up to date and getting backlog items ready for upcoming sprints.
- **Release Backlog:** The product owner co-ordinates with the scrum master to decide which user stories should be targeted for a release. Stories in the release backlog and targeted to be completed in a release.
- **Block List:** It is a list of blocks and unmade decisions owned by scrum master and updated daily.
- **BurnDown Charts:** Burn-down chart represents overall progress of the work in progress and work completed throughout the process. It represents in a graph format the stories and features not completed.

- **Scrum:** Is an agile development method which concentrates specifically on how to manage tasks within a team based development environment.
- **Scrum Master:** Master is responsible for setting up the team,sprint meetings and removed obstacles to progress.
- **Product Owner:** The product owner creates product backlog, prioritizes the backlog and is responsible for the delivery of functionality at each iteration.
- **Scrum Team:** Team manages its own work and organized the work to complete the sprint or cycle.

Agile - Scrum Framework



Scrum is based on 3 pillars

Roles
- Product Owner
- Scrum Master
- Team

Artifacts
- Product Backlog
- Sprint Backlog
- Burndown Charts

Ceremonies
- Sprint Planning
- Sprint Review
- Sprint Retrospective
- Daily Scrum Meeting

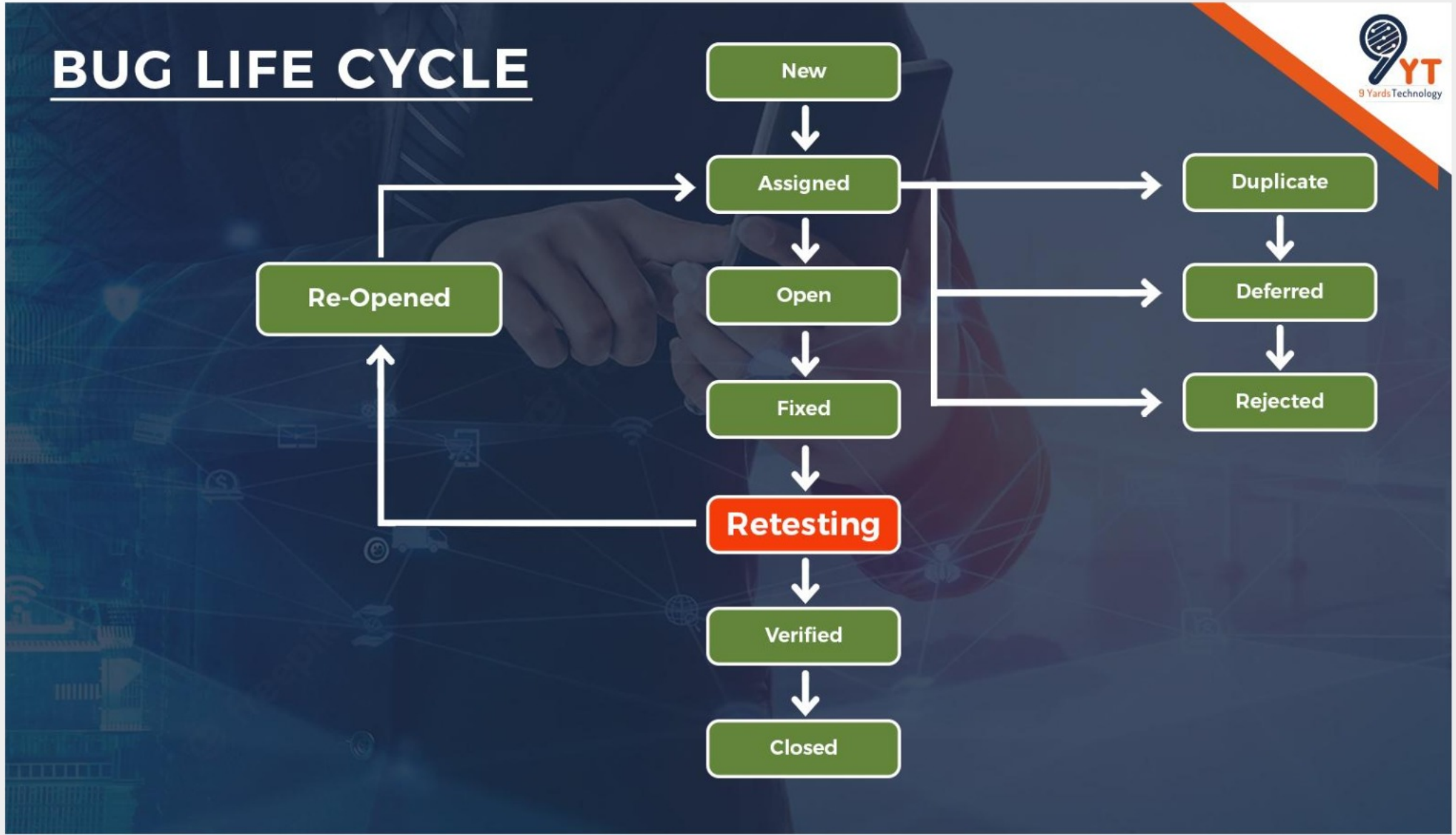
FAQ's

- **Story Points:** It is the estimation given by tester and dev for specific story. Size of the story depends on its complexity.
- **Epic:** An epic is a large that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories.
- **User Story:** A user story is the smallest unit of work in an agile framework.
- Epics are large work items broken down into a set of stories.
- The purpose of a user story is articulated how a piece of work will deliver a particular value back to the customer.
- User stories are a few sentences in simple language that outline the desired outcome.
- Requirements are added later, once agreed upon by the team
- User stories are often expressed in a simple sentences.

Requirement Traceability Matrix

- RTM is a mapping between requirements and test cases. It gives complete requirements coverage summary which of the requirements has been passed, failed, No Run, not completed and blocked progress etc. So that, it is easy to understand for client, the overall test execution or coverage progress by requirement wise.

Defect Life Cycle



- A defect life cycle is a process in which the reported defect goes through multiple stages in its life cycle. It starts as soon as a defect is found and closes when the tester is assured that it is fixed and not impacting any other functionality or areas of the software.
- There are two major persons required in any Software Development process–

1. **Software Developer –** One who designs and implements the code.
2. **Software Tester –** One who tests the software, detects as many defects as possible in the system, and reports them.

Defect States

- When a defect is reported, it has to go through different states as mentioned below:

1. **New:** It is the very first state of the Defect Life Cycle. It occurs when a new defect is found and reported. In this, the tester makes a proper document by mentioning how to reproduce the defect.

2. **Assigned:** In this state, the bug is assigned to a particular developer who takes full responsibility for fixing the defect.

3. **Open:** When the developer starts working on the defect, its state change from assigned to open. The developer will go through the details of the tester's information and will check if this is a valid defect or not. In this state, the defect can be moved to Duplicate, Deferred, or Rejected, based on analysis.

4. **Fixed:** When a defect is valid, the developer starts working on it. Once the developer fixes it by implementing the code changes and validates it on his end, the defect moves to the fixed state.

5. Pending Retest: Once the developer fixes the defect, he provides the latest code to the tester to retest the defect. The tester re-validates the defect and changes the defect status according to the results. It is changed to the 'Pending Retest' state when it is pending from the tester's end.

6. Retest: It is the tester's responsibility to retest the defect. If it is fixed, the tester closes the defect with proper evidence. If it is not fixed, the tester changes it to the re-open state. Once the defect is fixed, it moves to pending retest.

7. Re-Opened: Once a bug comes for a retest, the tester retests it. If the defect is still not fixed after the developer's fix, the tester re-opens the defect and assigns it back to the developer for the fix. In this case, the defects must go through the whole life cycle again.

8. Deferred: The bug can be marked as deferred also. When a bug is marked as deferred, it means that it will be fixed in the next release. There might be many reasons to mark the defect as 'Deferred'. It can be a low priority or a defect not affecting the software or functionality.

9. Duplicate: A defect can be marked as a duplicate if some other tester already reports the same defect. So, checking the reported defects before reporting any new defect is always advised to avoid duplicity.

10. Closed: When a defect comes for the retest, the tester retests the defect, and when he is confident that the defect is fixed properly, he marks the defect as close. It means the defect is retested, approved and closed, and it is not reproduced again.

11. Rejected: When the developer thinks the defect is not genuine, he marks the defect as rejected.

FAQ

- Once the bug is confirmed, what information we should provide to create a defect ?

Ans: Test Data, expected results, actual results, URL (QA environment), time stamp, troubleshoot thoughts, severity and priority.

What is Severity in Testing?

- One can define Severity as the extent to which any given defect can affect/ impact a particular software. Severity is basically a parameter that denotes the impact of any defect and its implication on a software's functionality. In other words, Severity defines the overall impact that any defect can have on a system.
- For instance, consider if a web page or an application crashes after clicking on a remote link. In such a case, a user would rarely click on the remote link. Yet, the overall impact of an app crashing is very severe. Hence, the severity gets high, and yet the priority gets low.

What is Priority in Testing?

- One can define Priority as a parameter for deciding the order in which one can fix the defect. In this, the defect with a higher priority first needs to get fixed. Priority basically defines the order in which one would resolve any given defect. The priority status defines if we should fix something or wait. The tester sets this priority status to the developer along with mentioning a time frame that can fix that defect. If they mention a higher priority, then the developer needs to fix it at the very earliest. Basically, the priority status comes into play according to the customer's requirements.
- For instance, let's consider a case where one misspells the company name on a website's home page. Thus, in this case, the Priority gets high while the Severity gets low for fixing it.

Difference Between Severity and Priority in Testing

Parameters	Severity in Testing	Priority in Testing
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameter	Severity is basically a parameter that denotes the total impact of a given defect on any software.	Priority is basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Change of Value	The value of Severity changes continually from time to time.	The value of Priority changes from time to time.
Who Decides the Defect	The testing engineer basically decides a defect's severity level.	The product manager basically decides a defect's priority level.
Types	There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.	There are 3 types of Priorities: High, Medium, and Low.

- **Smoke Testing:** When we get the build from development team first we will do smoke testing it is known as build verification testing.
- If the build is stable, we can continue the further testing. If the build is unstable we cannot continue the further testing. And we will inform the development team that the build is not stable with some justification/reasons.
- In smoke test we will do high level spot checks whether we are able to access every module or not, login etc.
- Verifies critical functionalities of software are working fine like application starts successfully or not.
- There wont be any planned test cases.
- Smoke test development and regression test development are related and similar. The only difference is depth scope and duration of running the tests.
- Typically runs after every build. And adds the most critical tests to the smoke test suite.

Sanity Testing

- It is also build verification testing.
- Sanity testing is a kind of a software testing performed after receiving the software build, with minor changes in code, or functionality. To ascertain that the bugs have been fixed and no further issues are introduced due to these changes.
- We will pick one or two test cases from every module and we will execute those planned test cases when we get the build.
- If the build is stable we can continue further testing. If the build is not stable, we will inform to development team that the build is not stable and share the test execution summary.
- Verifies new functionalities, bugs fixes in the build.

Regression Vs Retesting

Retesting: is a process to check specific test cases that are found with bug/s in the final execution. Generally, testers find these bugs while testing the software application and assign it to the developers to fix it. Then the developers fix the bug/s and assign it back to the testers for verification. This continuous process is called Retesting.

Regression Testing is a type of software testing executed to check whether a code change has not unfavorably disturbed current features & functions of an Application

Re-testing is carried out to confirm the test cases that failed in the final execution are passing after the defects are fixed	Regression Testing is carried out to confirm whether a recent program or code change has not adversely affected existing features
Re-testing is done on the basis of the Defect fixes	The purpose of Regression Testing is that new code changes should not have any side effects to existing functionalities
Defect verification is the part of re-testing	Defect verification is not the part of Regression Testing
Priority of re-testing is higher than regression testing, so it is carried out before regression testing	Based on the project and availability of resources, Regression Testing can be carried out parallel with Re-testing
You cannot automate the test cases for Retesting	You can do automation for regression testing, Manual Testing could be expensive and time-consuming
Re-testing is a planned testing	Regression testing is known as a generic testing
Retesting is done only for failed test cases	Regression testing is done for passed test cases
Re-testing executes a defect with the same data and the same environment with different inputs with a new build	Regression testing is only done when there is any modification or changes become mandatory in an existing project
Test cases for retesting cannot be obtained before start testing.	Test cases for regression testing can be obtained from the functional specification, user tutorials and manuals, and defect reports in regards to corrected problems

System Testing

- System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different type of tests with the purpose to exercise and examine the full working of an integrated software computer system against requirements.

- To check the end-to-end flow of an application or the software as a user is known as System testing. In this, we navigate (go through) all the necessary modules of an application and check if the end features or the end business works fine, and test the product as a whole system.

- It is **end-to-end testing** where the testing environment is similar to the production environment.

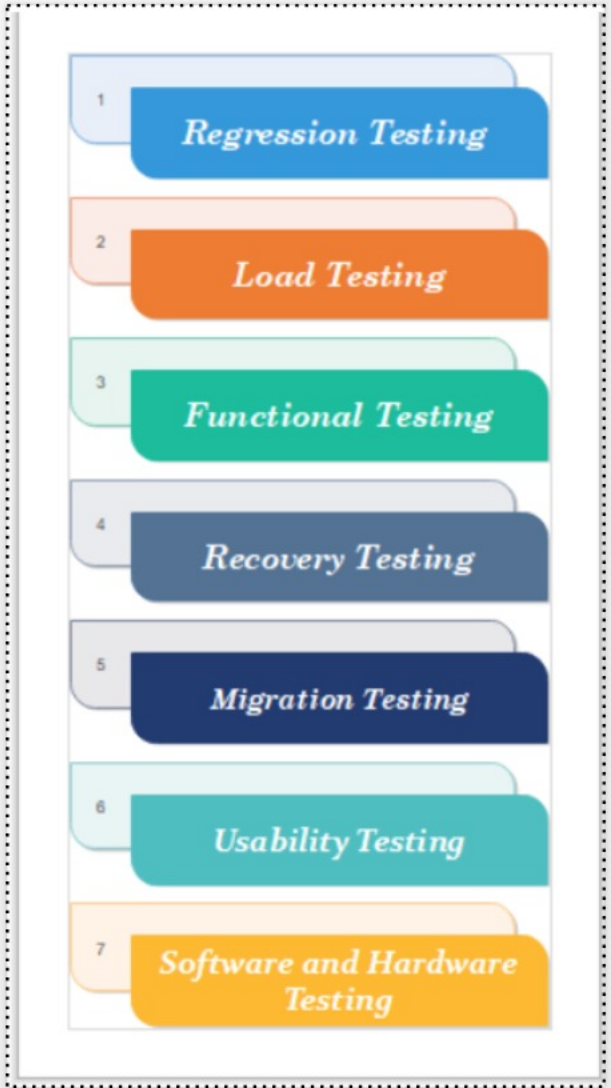
- There are four levels of **software testing**: **unit testing, integration testing, system testing and acceptance testing**, all are used for the testing purpose.
- **Unit Testing** used to test a single software
- **Integration Testing** used to test a group of units software
- **System Testing** used to test a whole system
- **Acceptance Testing** used to test the acceptability of business requirements. Here we are discussing system testing which is the third level of testing levels.

Hierarchy of Testing Levels

System Testing: It falls under black-box testing as it includes testing of external working of the software. Testing follows user's perspective to identify minor defects.

- System Testing includes the following steps.
- Verification of input functions of the application to test whether it is producing the expected output or not.
- Testing of integrated software by including external peripherals to check the interaction of various components with each other.
- Testing of the whole system for End to End testing.

Types of System Testing



- **Regression Testing:** Regression testing is performed under system testing to confirm and identify that if there's any defect in the system due to modification in any other part of the system. It makes sure, any changes done during the development process have not introduced a new defect and also gives assurance; old defects will not exist on the addition of new software over the time.
- **Load Testing:** Load testing is performed under system testing to clarify whether the system can work under real-time loads or not.
- **Functional Testing:** Functional testing of a system is performed to find if there's any missing function in the system. Tester makes a list of vital functions that should be in the system and can be added during functional testing and should improve quality of the system.
- **Recovery Testing:** Recovery testing of a system is performed under system testing to confirm reliability, trustworthiness, accountability of the system and all are lying on recouping skills of the system. It should be able to recover from all the possible system crashes successfully.

- In this testing, we will test the application to check how well it recovers from the crashes or disasters.

Recovery testing contains the following steps:

> Whenever the software crashes, it should not vanish but should write the **crash log message or the error log message** where the reason for crash should be mentioned. **For example**: **C://Program Files/QTP/Cresh.log**

> It should kill its own procedure before it vanishes. Like, in Windows, we have the Task Manager to show which process is running.

> We will introduce the bug and crash the application, which means that someone will lead us to how and when will the application crash. Or **By experiences**, after few months of involvement on working the product, we can get to know how and when the application will crash.

> Re-open the application; the application must be reopened with earlier settings.

- **For example**: Suppose, we are using the Google Chrome browser, if the power goes off, then we switch on the system and re-open the Google chrome, we get a message asking whether we want to **start a new session** or **restore the previous session**. For any developed product, the developer writes a recovery program that describes, why the software or the application is crashing, whether the crash log messages are written or not, etc.

- **Migration Testing:** Migration testing is performed to ensure that if the system needs to be modified in new infrastructure so it should be modified without any issue.
- **Usability Testing:** The purpose of this testing to make sure that the system is well familiar with the user and it meets its objective for what it supposed to do.
- **Why is System Testing Important?**

System Testing gives hundred percent assurance of system performance as it covers end to end function of the system.

- It includes testing of System software architecture and business requirements.
- It helps in mitigating live issues and bugs even after production.
- System testing uses both existing system and a new system to feed same data in both and then compare the differences in functionalities of added and existing functions so, the user can understand benefits of new added functions of the system.