

Testing types:

1. Functional

- Unit Testing
- Integration Testing
- System testing
- Regression testing.
- UAT(User Acceptance Testing/ Acceptance Testing)
- Smoke
- Sanity
- White box
- Black box

2. Non Functional Testing

- Documenttation
- Preformance testing
 - Stress Testing
 - Load Testing
 - Spike Testing
 - Volume Testing
 - Compatibility Testing
 - Security Testing.
 - Installation testing.

*******Functional Testing** : *****

1. **FUNCTIONAL TESTING** is a type of software testing that validates the software system against the functional requirements/specifications.

2. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

3. Functional testing mainly involves black box testing.
4. The testing can be done either manually or using automation.

Mainline functions: Testing the main functions of an application

Error Conditions: Usage of testing techniques to check for error conditions. It checks whether suitable error messages are displayed.

How to do Functional Testing ?

1. Understand the Functional Requirements
2. Identify test input or test data based on requirements
3. Compute the expected outcomes with selected test input values
4. Execute test cases
5. Compare actual and computed expected results

Difference between functional and non- functional Testing

<https://www.guru99.com/functional-testing.html>

***** Unit Testing *****

Unit Testing :

1. UNIT TESTING is a type of software testing where individual units or components of a software are tested.
2. The purpose is to validate that each unit of the software code performs as expected.

3. Unit Testing is done during the development (coding phase) of an application by the developers
4. Unit testing is first level of testing done before integration testing.
5. Unit testing is a WhiteBox testing technique that is usually performed by the developer

Why Unit Testing?

1. if proper unit testing is done in early development, then it saves time and money in the end.
2. inappropriate unit testing leads to high cost Defect fixing during System Testing, Integration Testing.

Very IMP Testing Sequence.

Unit Testing > Component Testing > Integration Testing > System Testing > Acceptance testing.

*******Integration Testing*******

Integration Testing :

1. INTEGRATION TESTING is defined as a type of testing where software modules are integrated logically and tested as a group.
2. A typical software project consists of multiple software modules, coded by different programmers.
3. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated

IMP :Where you Used Integration testing in your project , give me a example.

1.Create a Account : -- It is part of Login Component

Fills a forms: (First name , Last name , Email ID , Photo, mobile number , Education address)

2. Home Page- Data is populated from Create Account page , no need to set up data again. so Data is flowed from one component to another this is called Integration testing.

Approaches, Strategies, Methodologies of Integration Testing

1. Big Bang Approach : is an Integration testing approach in which all the components or modules are integrated together at once and then tested as a unit.

2. Incremental Approach: which is further divided into the following

<https://www.guru99.com/integration-testing.html>

i. Top Down Approach

ii. Bottom Up Approach

iii. Sandwich Approach - Combination of Top Down and Bottom Up

How to do Integration Testing?

1. Prepare the Integration Tests Plan

2. Design the Test Scenarios, Cases, and Scripts.

3. Executing the test Cases followed by reporting the defects.
4. Tracking & re-testing the defects.
5. Steps 3 and 4 are repeated until the completion of Integration is successful.

Brief Description of Integration Test Plans:

1. Methods/Approaches to testing (as discussed above).
2. Scopes and Out of Scopes Items of Integration Testing.
3. Roles and Responsibilities.
4. Pre-requisites for Integration testing.
5. Testing environment.
6. Risk Plans.

Entry and Exit Criteria of Integration Testing

Entry Criteria:

1. Unit Tested Components/Modules
2. All High prioritized bugs fixed and closed
3. All Modules to be code completed and integrated successfully.
4. Integration tests Plan, test case, scenarios to be signed off and documented.

5. Required Test Environment to be set up for Integration testing.

Exit Criteria:

1. Successful Testing of Integrated Application.
2. Executed Test Cases are documented
3. All High prioritized bugs fixed and closed
4. Technical documents to be submitted followed by release Notes.

******* System Testing *******

System Testing :

1. SYSTEM TESTING is a level of testing that validates the complete and fully integrated software product.
2. The purpose of a system test is to evaluate the end-to-end system specifications.
3. Ultimately, the software is interfaced with other software/hardware systems.
4. System Testing is Blackbox

What do you verify in System Testing?

1. Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.

2. Verify thorough testing of every input in the application to check for desired outputs.

3. Testing of the user's experience with the application.

Different Types of System Testing

1. Regression Testing

2. Hardware/Software Testing

3. Usability Testing : mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives

4. Migration testing : is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

*******Regression Testing*******

Whenever developers change or modify their software, even a small tweak can have unexpected consequences. Regression testing is testing existing software applications to make sure that a change or addition hasn't broken any existing functionality.

When we raise a bug , and devloper fix that bug and assign to tester back , then tester ReTest the same bug. and also test impacted area of the module called Regreesion testing. because while fixing bug by bt devloper, it may have chance breaking exsiting working functionality. so we need to do Regression testing.

***** Re Testing *****

When you raise a bug , devloper fixed that bug and assign back to the tester. then Tester Retest that bug called Retesting.

***** Difference between Re Testing and regression testing *****

ReTesting is sub set Regression testing.

Regression testing = ReTesting of bug + Impacted area test brcause of this bug fix.

*****Smoke Testing *****

1. Identifying the business critical functionalities that a product must satisfy.
2. Designing and executing the basic functionalities of the application.
3. Ensuring that the smoke test passes each and every build in order to proceed with the testing.
4. Smoke Tests enables uncovering obvious errors which saves time and effort of test team.
5. Smoke Tests can be manual or automated.

***** Sanity Testing *****

1. Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code
2. It also validates the accuracy of the newly added features and components.
3. The execution of sanity testing makes sure that new modifications don't change the software's current functionalities.

*******Difference between Smoke and Sanity*******

<https://www.guru99.com/smoke-sanity-testing.html#:~:text=Sanity%20testing%20is%20a%20kind,functionality%20works%20roughly%20as%20expected.>

*******Acceptance Testing *******

1. Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications

2. There are various forms of acceptance testing:

User acceptance Testing

Business acceptance Testing

Alpha Testing

Beta Testing

3. Acceptance Criteria

Acceptance criteria are defined on the basis of the following attributes

Functional Correctness and Completeness

Data Integrity

Data Conversion

Usability

Performance

***** UAT *****

1. User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment.

<https://www.guru99.com/user-acceptance-testing.html>

***** Alpha Testing *****

1. Alpha Testing is a type of acceptance testing; performed to identify all possible issues and bugs before releasing the final product to the end users
2. Alpha testing is carried out by the testers who are internal employees of the organization.

***** Beta Testing *****

1. Beta Testing is performed by "real users" of the software application in "real environment" and it can be considered as a form of external User Acceptance Testing.
2. It is the final test before shipping a product to the customers.

***** Difference between Alpha testing and Beta testing*****

<https://www.guru99.com/alpha-beta-testing-demystified.html#:~:text=Alpha%20Testing%20is%20a%20type,internal%20employees%20of%20the%20organization.>

***** **Black box testing** *****

<https://www.guru99.com/black-box-testing.html>

1. Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths.
2. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

***** **White box testing** *****

1. White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security.

2. White box testing involves the testing of the software code for the following:

- Internal security holes

- Broken or poorly structured paths in the coding processes

- The flow of specific inputs through the code

- Expected output

- The functionality of conditional loops

- Testing of each statement, object, and function on an individual basis

***** **Difference between white box and black testing*******

<https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/>

Ad-hoc Testing: Testing performed without planning and documentation - the tester tries to 'break' the system by randomly trying the system's functionality. It is performed by the testing team

Benchmark Testing: Testing technique that uses representative sets of programs and data designed to evaluate the performance of computer hardware and software in a given configuration

Big Bang Integration Testing: Testing technique which integrates individual program modules only when everything is ready. It is performed by the testing teams.

Boundary Value Testing: Software testing technique in which tests are designed to include representatives of boundary values. It is performed by the QA testing teams.

Edit Box Maximum 100

99 : Should Accept values

100 : Should Accept values

100.1 or more than : Should not Accpet value.

Password Reset :

1 digit.

minimum 8 maximum 20

7 alpha Numric password. : Should not accept.

8 alpha Numric password. : Should Accept.

IMPS :

Minimum : 1 rs

0.99 paise

1 RS

Max limit : 200000

1 rs Boumdary

0.99 : not Accept

1 : Accept

1.1 > Accept

2,00,000 :

1,99,000.99 > Should Accept

2,00,000 : Should Accept

2,00,000.1 : Not Accept

Compatibility Testing: Testing technique that validates how well a software performs in a particular hardware/software/operating system/network environment.

Component Testing: Testing technique similar to unit testing but with a higher level of integration - testing is done in the context of the application instead of just directly testing a specific method. Can be performed by testing or development teams.

Static testing :

Scenario preparation , What will be expected output , Test case Documentation , TC Writing , Test Case Review : Static Testing.

Without Executing Test cases. : called Static testing.

Dynamic Testing : Actual Test Case Execution called dynamic testing.

End-to-end Testing: Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as

interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate. It is performed by QA teams.

Functional Testing: Type of black box testing that bases its test cases on the specifications of the software component under test. It is performed by testing teams

Gray Box Testing: A combination of Black Box and White Box testing methodologies: testing a piece of software against its specification but using some knowledge of its internal workings.

Gray Box Testing: A combination of Black Box and White Box testing methodologies: testing a piece of software against its specification but using some knowledge of its internal workings.

Integration Testing: The phase in software testing in which individual software modules are combined and tested as a group.

Interface Testing: Testing conducted to evaluate whether systems or components pass data and control correctly to one another.

Non-functional Testing: Testing technique which focuses on testing of a software application for its non-functional requirements. : Security Testing , Load testing , Stress testing.

Negative Testing: Also known as "test to fail" - testing method where the tests' aim is showing that a component or system does not work.

Passive Testing: Testing technique consisting in monitoring the results of a running system without introducing any special test data.

Regression Testing: Type of software testing that seeks to uncover software errors after changes to the program (e.g. bug fixes or new functionality) have been made, by retesting the program

Security Testing: A process to determine that an information system protects data and maintains functionality as intended. It can be performed by testing teams or by specialized security-testing companies.

1. Third person should not accept.
2. GMail installed. if you try from another device , you will get notification to acknowledge it , if you are that person.
3. 2 Step Verification.
4. Password that should be Encrypted format when it flow from Client machine to Server.(base 64 bit Encoder)
5. At A time only one Sesion should be open.

Sanity Testing: Testing technique which determines if a new software version is performing well enough to accept it for a major testing effort.

Smoke Testing: Testing technique which examines all the basic components of a software system to ensure that they work properly. Typically, smoke testing is conducted by the testing team, immediately after a software build is made.

System Testing: The process of testing an integrated hardware and software system to verify that the system meets its specified requirements. It is conducted by the testing teams in both development and target environment

System integration Testing: Testing process that exercises a software system's coexistence with others. It is usually performed by the testing teams.

Unit Testing: Software verification and validation method in which a programmer tests if individual units of source code are fit for use. It is usually conducted by the development team

Defect Life Cycle:

1. Defect Raise Status : New>
2. You Assigned to developer : Open
3. when doveloper starts working on it > In progress.
4. When devloper fix that bug : Fixed.
5. Devloper assign back to QA : Reteting / Ready for Testing.
6. QA test the Bug :

```
if {functionality works as expected }  
    then Status > closed  
else {  
    Status > Reopen (Goes to Step 2)  
}
```

Developer : Not a bug : As per Requirements it works fine.

Duplicate : already same issue/ bug Raised.

Deferred : will not fix in Current sprint / will be fixed in next Sprint.

Scenarios : What you will do , if Developer not Accepting a bug.

- > You Will provide Screen shot / Steps to Reproduce.
- > Still he is not Ready to Accept a bug , then you will provide Requirement Docs for supporting for a bug.
- > Still Developer not ready to Accept a bug
then I will raise a concern in the Daily Scrum meeting to take decision from Scrum Master.

Scrum master arrange a meeting for Respective Dev , Respective QA Member and Stakeholders. the they discuss , and Scrum master finalized the decision what to do.

Scenario 2 : What you will do , if it works fine at Dev side but not on your side.

- > Then you Should ask questions to Developer.
 - > On which Browser you have try to reproduce.
 - > on which build you tried.
 - > On machine Configuration.
 - > Have you tried with deleting cookies.

If all developer followed all above steps. still exist issue.

then show him to by sharing Screen.

then Developer takes a call what to do. but he has to fix that issue.

Scenario 3 : If bug is reproducible randomly. but when for developer it is not reproducible .

then developer mark as Not a bug. then What you will do.

- > We will provide Snap shots and try to provide him.
- > still he is not agree to accept a bug.
 - > then you will provide Application Logs to the developer when that bug is introduced.

Scenario 4 : Assume , tomorrow is Live production. (code is going to be)

Assume : Tomorrow's code goes to Production and you found One critical bug. what will be your Approach

1. You will See Impact of it. (How severe bug is)

2. Inform to the Team Lead/ Manager and expelling about the Sceanrio.

3. Inform to Devloer with Dev Team Lead ,Discussed wtih them how Quick they can fix a bug.

4. if dev says it takes too much time to fixed , then what you will do ?

We will discussed with client to about delaying a build / Release.

if Client is not agree for delaying a relaease then what you will do ?

5. You will calculate all risk about the bug

Give them a sign Off with condition
(Specifying Bug details and Impaced area / sceanrios.)

6. client will Mointor more on that bugs scenario's specified by you.

Jira :

Defect management tool.

you can Rasie create / modify /Link / unlink Bug or stories.

- Can Create a bug /Story/Epic/Enhancement/Issue
- can track a status of bug /Story/Epic/Enhancement/Issue
- can Link / unlink bug /Story/Epic/Enhancement/Issue to each other.

When you Create bug what fields you will fills.

1. Login to Jira.

2. Click on Create.

(https://www.google.com/search?q=how+to+link+a+bug+to+story+in+jira&rlz=1C1SQJL_enIN920IN920&hl=en-GB&sxsrf=ALeKk030S2_0k4veSAu3JLLQ__vjX7eVw:1616215193271&source=Inms&tbm=isch&sa=X&ved=2ahUKEwjVsObjhr7vAhWXzjgGHe-zDXsQ_AUoAnoECAkQBA&biw=1745&bih=852#imgrc=DV4_g4lHAuK4eM)

3. Then Choose a project.

4.

1.Choose Issue type :

When you choose Issue type : what you will observe in the dropdown.

()

1. Bug

2. Story

3. Epic

4. Enhancement

5. Task

2. Summary (Summary of bug : login to the application is failed.

3. Reporter : Automatically who logged in , it display his name.

4. Component : Login / Log out / H0me Page / Profile /Setting

5. Environment : SIT / QA / UAT /PROD

6. Afftected Version : Sprint Number (21.1.1)

Sprint 1. Menaing on 21.1.1 : year 2021 . Release 1(Jan to Mar) and

Sprint 2. Menaing on 21.2.1 : year 2021 . Release 2(Apr to Jun) and

7. Browser : Chrome / IE / Mozilla / Microsoft Edge

8. Build No. : In agile daily builds happens so , BUild number is very important.

9. Steps to reproduce:

1. Open a browser and enter a URL
2. Enter Correct User Name (User ID : vxvcvx)
3. Enter Correct Password(password : vxvcvx)
4. Verify User login to the appliation.

Actual : User is not able to login to application , shows invalid crednetial.

Expected : User should able to login.

10. Attachments.

11. Assigned to : Devloer Name.

12. Priority.

13. Severiy.

14. Click on Create.

15. Then Bug Id automatically genrated.

How to Link a bug to a Story:

Go to Bug Detail apge.

Click on more

(https://www.google.com/search?q=how+to+link+a+bug+to+story+in+jira&rlz=1C1SQJL_enIN920IN920&hl=en-GB&sxsrf=ALeKk030S2_0k4veSAu3JLLQ__vjX7eVw:1616215193271&source=Inms&tbm=isch&sa=X&ved=2ahUKEwjVsObjhr7vAhWXzjgGHe-zDXsQ_AUoAnoECAkQBA&biw=1745&bih=852#imgrc=SdA0gP3VIMEVRM)

Click on Link on pop up will

open.(https://www.google.com/search?q=how+to+link+a+bug+to+story+in+jira&tbm=isch&hl=en-GB&chips=q:how+to+link+a+bug+to+story+in+jira,online_chips:linked+issues&rlz=1C1SQJL_enIN920IN920&sa=X&ved=2ahUKEwiX05PIhr7vAhUICCsKHe8hAZsQ4IYoB3oECAEQIA&biw=1730&bih=852#imgrc=J0vE-kCGlpU0oM)

This Issue has : "Blocks " Story Number

Click on Link.

Then this Bug is linked to the story.

Data Base :

Data Base

Combination of multiple Linked Table

Constraints :

Primary Key

Foregien Key

Unique Key

Not Null Values

Primary Key :

Uniue Value and contains all Not null values.

Foregien Key :

A foreign key is a key used to link two tables together. This is sometimes also called as a referencing key.

A Foreign Key is a column or a combination of columns whose values match a Primary Key in a different table.

Unique Key : All values should be Unique and One Null is Allowwed.

Difference Between Primary and Unique Key.

Table1 : it can have maximum one Primary Key

Table 2 : it can have n number of Unique key :

Primary Key : Unique Value and contains all Not null values.

Unique Key : All values should be Unique and One Null is Allowed.