

NAME: KSHATRIYA RONAK OMPRAKASH

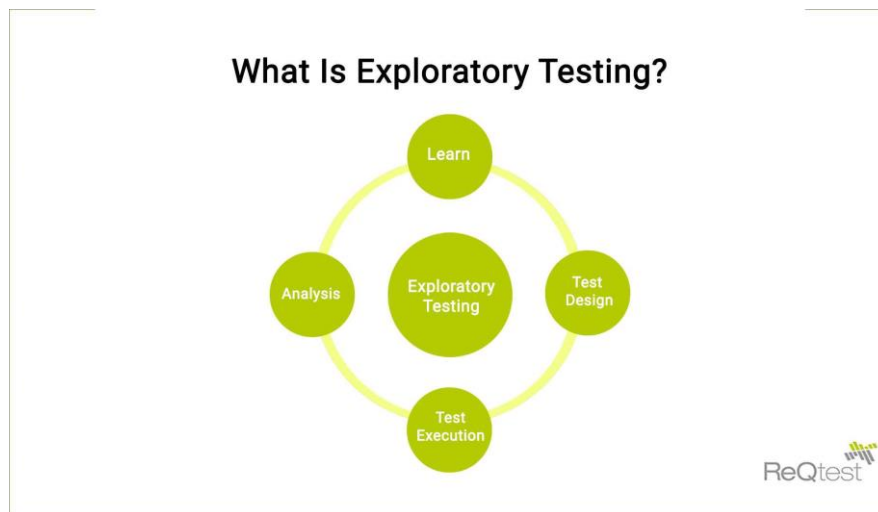
**COURSE: SOFTWARE TESTING (MANUAL AND
AUTOMATION)**

ASSIGNMENT: MANUAL TESTING

1. What is Exploratory Testing?

Exploratory testing is a concurrent process where Test design, execution and logging happen simultaneously.

- Testing is based on a test charter that may include
Scope of the testing (in and out)
- The focus of exploratory testing is more on testing as a “thinking” activity.



2. What is traceability matrix?

A **software process** should help you keeping the virtual table up-to-date.

To find out the cause of defect.

To ensure the complete coverage of testing.

[illegible]

3. What is Boundary value testing?

Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges. Boundary value analysis is a method which refines equivalence partitioning.

E.g. 1 to 50

LB : 1

UB : 50

0 1 2 49 50 51

a-1 a a+1 b-1 b b+1

4. What is Equivalence partitioning testing?

Equivalence partitioning is the process of defining the optimum number of tests by: Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function,

E.g. range : 1 to 100

emp_no :

1 to 50 :

1 to 10 : 6 : pass

11 to 20 : 15 : pass

21 to 30 : 22

31 to 40 : 33

41 to 50 : 49 :

91 to 100 : 93 : pass

101 to 110 : 105 : fail

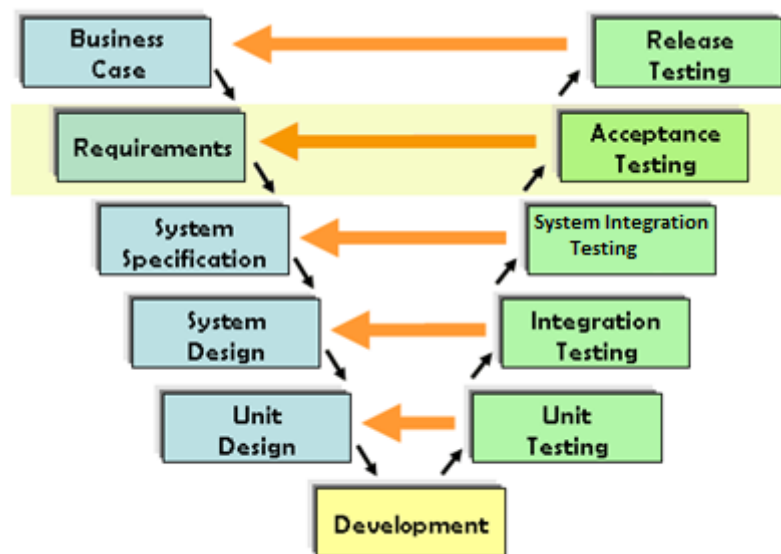
5. What is Integration testing?

Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

Integration Testing is a level of the software testing process where individual units are combined and tested as a group.

e.g username & password always checked by the login button.

Combinable need to check both.



6. What determines the level of risk?

A risk - A factor that could result in future negative consequences, usually express as impact and likelihood (possibility).

To identify the risk is not enough, you need to identify the mitigation (reduce the harmness).

Types_of_Risk

Project Risk

Product Risk

7. What is Alpha testing?

- It is always performed by the developers at the software development site.
- Sometimes it is also performed by Independent Testing Team.
- Alpha Testing is not open to the market and public
- It is conducted for the software application and project.
- It is always performed in Virtual Environment.

8. What is Beta testing?

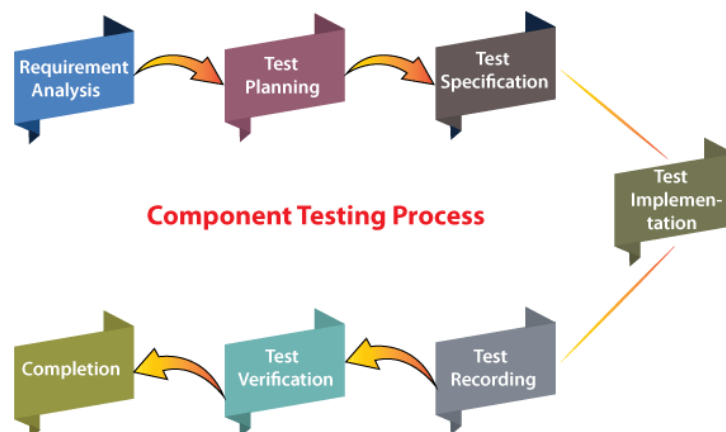
- Beta Testing is always performed at the time when software product and project are marketed.
- It is always performed at the user's premises in the absence of the development team.
- It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area.
- Beta testing can be considered "pre-release" testing.

9. What is component testing?

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested.

– A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”

- Run by software developers.
- Unit testing is performed by using the White Box Testing method.
- S/w Developer can test the source code as well as executable code.



10. What is functional system testing?

Testing based on an analysis of the specification of the functionality of a component or system.

- ‘Specification’ – E.g. Requirements specification, Use Cases, Functional specification or maybe undocumented.
- ‘Function’ – what the system do
 - **Web Based Testing:** 1. Are you able to login to a system after entering correct credentials?
 - 2. Does your payment gateway prompt an error message when you enter incorrect card number?

- **Desktop Based Testing:** 1. Verifies Installation testing, Check for broken lines,
2. Testing with different client accounts, Theme change and Print,
- **Mobile Based Testing:** 1. To validate whether the application works as per as requirement whenever the application starts/stops
2. To validate whether the application goes into minimized mode whenever there is an incoming phone call. In order to validate the same we need to use a second phone, to call the device

11. What is Non-Functional Testing?

Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability

- May be performed at all Test levels
- It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.
- **Web Based Testing:** 1. Identify the software processes that directly influence the overall performance of the system.
2. In website number of user/customer will increase, how the website will handled to every customer/user.

Mobile Based Testing: 1. In mobile, automatically will switch off without any reason.

2. To stop the application which is not in our hand.

12. What is GUI Testing?

To check the look and feel of the app.

Graphical User Interface (GUI) testing is the process of testing the system's , GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

Approach of GUI Testing

1) Manual based GUI

2) Record & Replay GUI

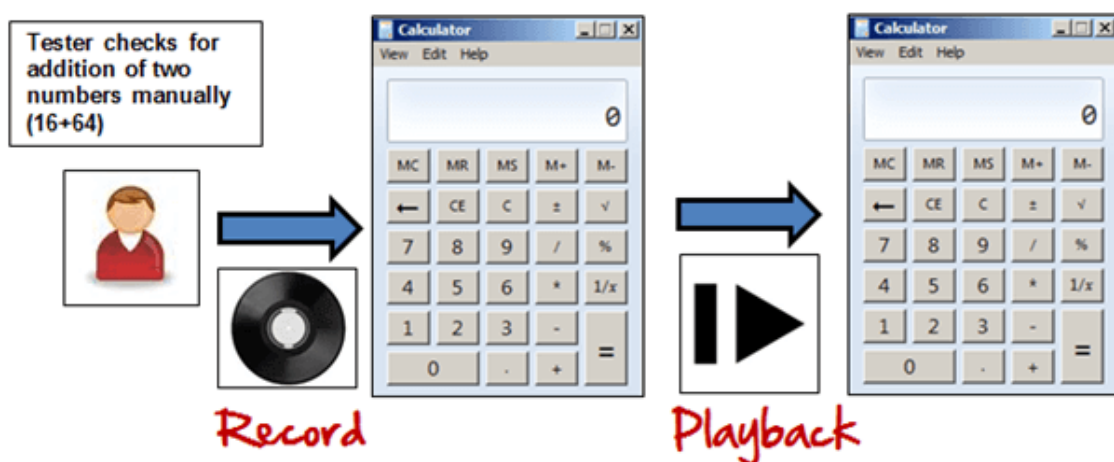
3) Model based

For e. g

If mobile is in every orientation mode so display image, video properly.

Font size, style, and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS.

The description text box should be multi-lined.



13. What is Adhoc testing?

- Adhoc testing is an informal testing type with an aim to break the system.
- It does not follow any test design techniques to create test cases.
- This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Main aim of this testing is to find defects by random checking.
- Adhoc testing can be achieved with the testing technique called Error Guessing.

Types of Adhoc Testing :

- 1) Buddy Testing (one developer, One tester)
- 2) Pair Testing (Two persons paired as testers)
- 3) Monkey Testing (randomly jump to the app. area to break)

14. What is load testing?

- Stability + response time + applying load (app will withstand with designed no. of users)
- Loading testing identifies the following problems before moving the application to market or Production Response time for each transaction
- Performance of System components under various loads.

e.g. app will handle 1000 users at every 5 sec.

You have to check 1000 or ≤ 1000 users with your app.

15. What is stress Testing?

Stability + response time + applying load (app will withstand with designed no. of users).

Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.

Stress testing is also known as endurance testing.

e.g app will handle 1000 users at every 5 sec.

You have to check 1000 or ≥ 1000 users with your app.

16. What is white box testing and list the types of white box testing?

White Box Testing: Testing based on an analysis of the internal Structure of the component or system.

The testers require knowledge of how the software is implemented, how it works. also called "Glass box testing " or "Open box Testing".

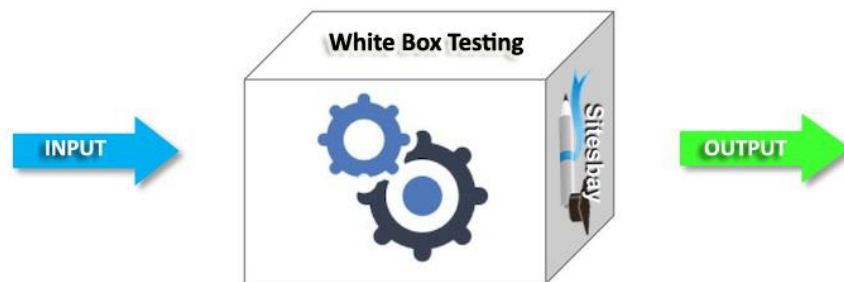


Fig: White Box Testing

White box Testing Techniques:

1. Statement coverage (segment)

Covers only the true conditions

2. Decision coverage (branch coverage)

It covers both the true and false conditions

3. Condition coverage

Full condition coverage does not guarantee full decision Coverage.

17. What is black box testing? What are the different black box testing techniques?

- Testing, either functional or non-functional, without reference to the internal structure of the component or system.
- -You are just executing the executable code as the source code is not accessible as an internal part of the system.
- -have no knowledge of how the system or component is structured inside the box.
- -In black-box testing the tester is concentrating on what the software does, not how it does it.

Black Box Testing Techniques:

1. Equivalence partitioning

2. Boundary value analysis

3. Decision tables

4. State transition testing

1) Equivalence partitioning (EP)

This is the method to test the range by dividing the range into equivalent partitions & select one representative from each partition.

Test the representative to check the whole partition.

e. g range : 1 to 100

emp_no :

1 to 50 :

1 to 10 : 6 : pass

11 to 20: 15 : pass

21 to 30 : 22

31 to 40 : 33

41 to 50 : 49 :

51 to 60

.

91 to 100 : 93 : pass

101 to 110 : 105 : fail

2) Boundry Value Analysis (BVA)

e.g 1 to 50

LB : 1

UB : 50

0 1 2 49 50 51

a-1 a a+1 b-1 b b+1

3) Decision Table

Decision tables is more focused on business logic or business rules. Also called 'cause-effect' table "Cause Effect Graphics".

True / false (boolean value result)

e.g Myntra :

Insiders : ≥ 2 years then price 15%

Mr X is a member from last 1.5 years.

Mr Z is a member from last 4 yeras.

Mr. X

- 1) Membership is there (True)
- 2) Membership ≥ 2 y (False)
- 3) get the price discount 15% (False)

Mr. Z

- 1) Membership is there (True)
- 2) Membership ≥ 2 y (True)
- 3) get the price discount 15% (True)

4) State Transaction Testing

Finite State Machine :

In which each transactions are stored & we need to test each state of transaction.

e.g. To test ATM machine (as it is an best example of Black box)

18. Mention what are the categories of defects?

- Defects can be categorized into different types basing on the core issues they address.
- Some defects address security or database issues while others may refer to functionality or UI issues.

Functionality Defects:

Defects directly related to functionalities. Not working features properly. e.g Calculator has no '=' button for the calculation.

Performance Defects:

Software doesn't meet the expected performance requirements. e.g Website's loading time to open.

User Interface Defects:

Difficult to operate for the users. Not user friendly. e.g Login page has no cancel button, Alignment problem.

Compatibility Defects:

Software does not work correctly on different hardware and software configuration. e.g Application not running on Android or Windows platform. Application interface shows differently in different browsers.

Security Defects:

Software doesn't protect the user's data from malicious attack. e.g Password entered in visible form.

Authentication: Accepting an invalid username/password

Authorization: Accessibility to pages though permission not given

Documentation Defects:

Document is incorrect or inaccurate to use the features of the app. e.g TC had a wrong entry.

19. Mention what bigbang testing is?

- In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
- Big Bang testing has the advantage that everything is finished before integration testing starts.

Advantages: Convenient for small systems.

Disadvantages: 1. Fault Localization is difficult.

2. Given the sheer number of interfaces that need to be tested in this approach, some interfaces links to be tested could be missed easily.

20. What is the purpose of exit criteria?

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

21. When should "Regression Testing" be performed?

- Regression testing should be carried out:
- when the system is stable and the system or the environment changes
- when testing bug-fix releases as part of the maintenance phase It should be applied at all Test Levels
- It should be considered complete when agreed completion criteria for regression testing have been met
- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation.

22. Difference between QA v/s QC v/s Tester.

	QA	QC	Testing
Approach →	Proactive	Reactive	Reactive
Focus →	Process-oriented	Product-oriented	Focused on finding mistakes and bugs
Techniques →	Developing strategies, establishing standards, creating test management plans	Verifying the quality of software using testing, code reviews, continuous	Conducting all types of testing through manual or automated tools
Timeframe →	Throughout the development process	Before the product release	Along the development process
Executor →	The entire team, including stakeholders	A small, dedicated team	Test engineers, developers

23. Difference between Smoke and Sanity?

Smoke Testing	Sanity Testing
Smoke Testing is performed to check that the application's critical functions are working properly.	Sanity testing is performed to check that the newly added functionalities/bug fixes are working fine.
Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.
Smoke Testing is performed on initial build.	Sanity Testing is performed on stable build .
Smoke testing is a subset of acceptance testing	Sanity testing is a subset of Regression Testing .
The majority of smoke testing is documented or scripted.	The majority of sanity testing is not documented or scripted .

24. Explain types of Performance testing

Types of Performance Testing

- **Load testing**
- **Stress testing**
- **Endurance testing**
- **Spike testing**
- **Volume testing**
- **Scalability testing**

-> **Load Testing**

Stability + response time + applying load (app will withstand with designed no. of users)

e.g app will handle 1000 users at every 5 sec.

You have to check 1000 or ≤ 1000 users with your app.

-> **Stress Testing**

Stability + response time + applying load (app will withstand with designed no. of users)

e.g app will handle 1000 users at every 5 sec.

You have to check 1000 or ≥ 1000 users with your app.

-> **Scalability Testing**

Stability + response time + applying load (app will withstand with designed no. of users)

e.g You are checking the performance of the app.

continue with load until your system will be crashed.

app will handle 1000 users at every 5 sec.

1500 users : 10 sec

2000 users : 20 sec

.....

1,00,000 users crashed..

->Volume Testing (Flood Testing)

Stability + response time + applying load (app will withstand with designed no. of users)

To check the capacity or volume of database.

->Endurance Testing (Soak Testing)

Stability + response time + applying load (app will withstand with designed no. of users)

e.g To check how the system will run continuously.

-> Spike Testing

Stability + response time + applying load (app will withstand with designed no. of users)

e.g to check extreme increment or decrement of load according to the response time.

25. What is Error, Defect, Bug and failure?

Error: A mistake in coding is error.

Defect: An error found by tester is defect.

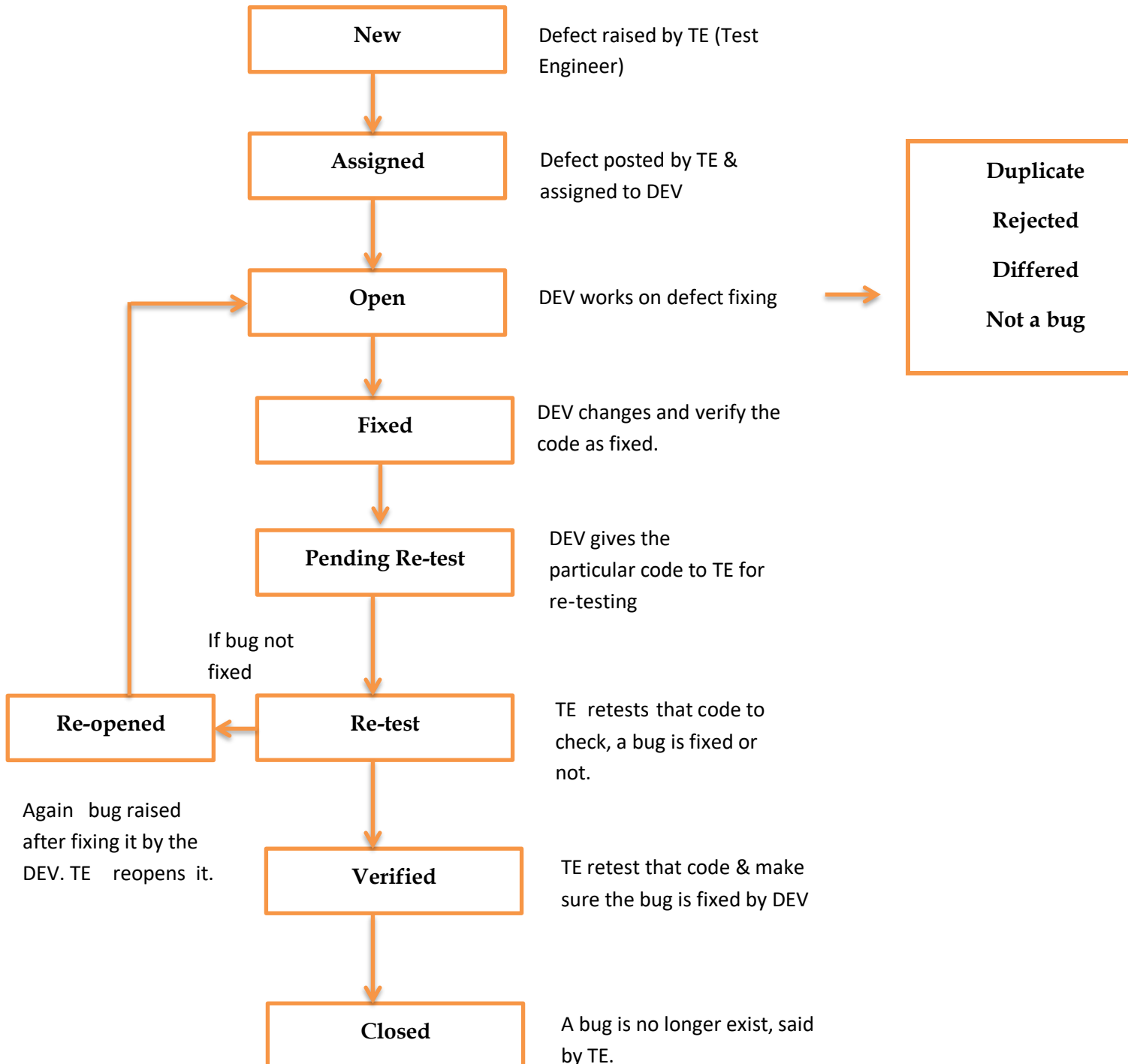
Bug: Defect accepted by development team is bug.

Failure: Build does not meet the requirements is failure.

26. Difference between Priority and Severity.

H2K Infosys Dream , Strive & Achieve Victory		SEVERITY	
PRIORITY	HIGH	Key features failed and no workaround E.g. Login button is not working	Basic feature failed but it has a huge impact on customer's business E.g. Misspelled Company logo
	LOW	Key features failed but there is no impact on customer's business E.g. Calculation fault in yearly report which end user won't use regularly	Cosmetic Issues E.g. Font family mismatch in a report






27. What is Bug Life Cycle?



28. Explain the difference between Functional testing and Non Functional testing.

Functional Testing	VS	Non-Functional Testing
<ul style="list-style-type: none"> Test the functionality of the software. It has to be done before Non-Functional Testing. It is also called as Behavioral Testing and focuses on the underlying application features. It can be done manually, though test cases 		<ul style="list-style-type: none"> Test the non-functional aspects or readiness of the the software including performance, usability, reliability. It will be done after Functional Testing completes. Focuses on the performance of the application. It's hard to do it manually. It usually need

29. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

	SDLC	STLC
 Core Objective	Oversees the complete development of a software product	Focuses specifically on ensuring the software is tested thoroughly.
 Phases Involved	Encompasses planning, designing, building, testing, and maintenance.	Consists of test planning, test case development, environment setup, and test
 Team Composition	Includes project managers, business analysts, developers, and designers.	Involves quality assurance personnel and testers who specialize in identifying defects.
 Outcome	Includes project managers, business analysts, developers, and designers.	Involves quality assurance personnel and testers who specialize in identifying defects.
 Primary Focus	Prioritizes developing functional and feature-complete software	Emphasizes ensuring software is defect-free and meets quality standards

30. What is the difference between test scenarios, test cases, and test script?

Test Scenario	Test Case	Test Script
Is any functionality that can be tested.	Is a set of actions executed to verify particular features or functionality.	Is a set of instructions to test an app automatically.
Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirement Specification (SRS).	Is mostly derived from test scenarios.	Is mostly derived from test cases.
Helps test the end-to-end functionality in an Agile way.	Helps in exhaustive testing of an app.	Helps to test specific things repeatedly.
Is more focused on what to test.	Is focused on what to test and how to test.	Is focused on the expected result.
Takes less time and fewer resources to create.	Requires more resources and time.	Requires less time for testing but more resources for scripts creating and updating.
Includes an end-to-end functionality to be tested.	Includes test steps, data, expected results for testing.	Includes different commands to develop a script.
The main task is to check the full functionality of a software application.	The main task is to verify compliance with the applicable standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.
Allows quickly assessing the testing scope.	Allows detecting errors and defects.	Allows carrying out an automatic execution of test cases.

31. Explain what Test Plan is? What is the information that should be covered?

A document describing the scope, approach, resources and schedule of intended test activities.

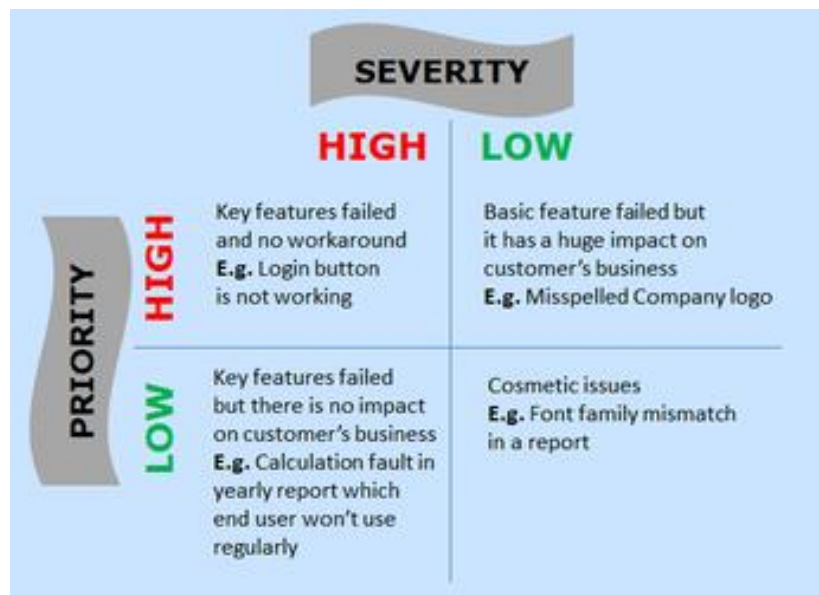
Determining the scope and risks, and identifying the objectives of testing.

A document describing the scope, approach, resources and schedule of intended test activities.

- Determining the scope and risks and identifying the objectives of the testing.
- Defining the overall approach of testing including test entry and exit criteria.
- Integrating and coordinate the testing activities into software life cycle.
- Scheduling test analysis, design, implementation, execution and evaluation activities.

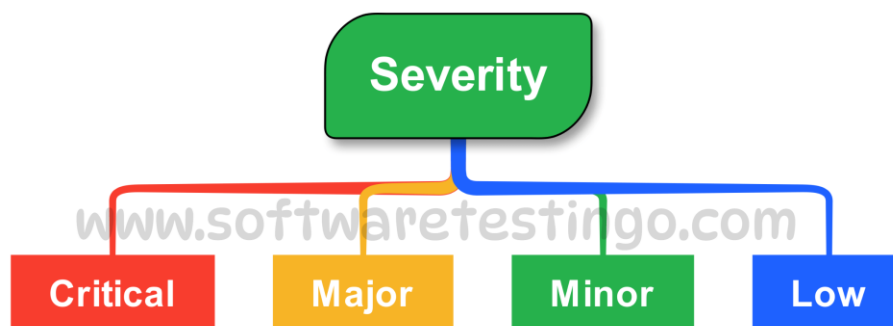
32. What is priority?

- If you are raising any bug for any application, how soon you want the developer to fix that bug is called priority.
- Priority is considered as customer's point of view. But priority can be set by the QA tester. Later on it can be changed by project manager.
- The importance given to the bug to fix it that is priority.
- The high priority indicates that the bug to fix it first.



33. What is severity?

- The impact of Defect /bug on the customer business workflow is known as Severity.
- If that impact is more then, there is high severity.
- If that impact is less then, there is low severity.



34. Bug categories are...

Here are some common types of bugs in software testing:

- **Functional bugs**

These bugs occur when a program doesn't respond as expected, such as when a button doesn't submit a form or an app crashes.

- **Security bugs**

These bugs allow unauthorized access to a computer system, such as by compromising user authentication or data integrity.

- **Usability bugs**

These bugs impact the user experience, such as when the user interface is confusing or the workflows are non-intuitive.

- **Performance bugs**

These bugs affect a software's performance, such as its speed, response time, or resource consumption.

- **Syntax errors**

These errors occur when the code is not written correctly and doesn't follow the syntax rules of the programming language.

- **Unit-level bugs**

These bugs are related to the functionality of a single software unit, such as a class, method, or procedure.

- **Integration level bugs**

These bugs are expensive because they are usually caught late and force changes to several components or data structures.

- **Logical bugs**

These bugs are unique and can't be identified by scanners or typical security tools

35. Advantage of Bugzilla.

Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.

Key features of Bugzilla includes

- **Advanced search capabilities**
- **E-mail Notifications**
- **Modify/file Bugs by e-mail**
- **Time tracking**
- **Strong security**
- **Customization**
- **Localization**

36. Difference between priority and severity.

<u>PRIORITY</u>		<u>SEVERITY</u>
Fix now Drop what you're doing and fix <u>immediately</u>	1	Bleeding money Money is lost every second (or rep.)
Fix asap Fix it as soon as there is time, today	2	Broken Some important functionality cannot be used
Fix this sprint Sometime during the next weeks	3	Barely usable You can work around the issue, kind of
Fix this release Plan it for the next sprint or before release	4	Needs improvement To be good it really needs to be fixed
Fix in due time Plan it for future release	5	Nice to have Cosmetic fix to make the service better

37. What are the different Methodologies in Agile Development Model?

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage.

● Individuals and interactions, Over processes and tools

Suppose the team finds any issue in software then they search for another process or tool to resolve the issue. But, in Agile, it is preferable to interact with client, manager or team regarding issue and make sure that the issue gets resolved.

● Working software, Over comprehensive documentation

Documentation is needed, but working software is much needed. Agile is not saying that documentation is not needed, but working software is much needed. For example, you have 20-page documents, but you do not have a single prototype of the software. In such a case, the client will not be happy because, in the end, the client needs a document.

● Customer collaboration, Over contract negotiation

Contract negotiation is important as they make the budget of software, but customer collaboration is more important than over contract negotiation. For example, if you stuck with the requirements or process, then do not go for a contract which we have negotiated. You need to interact with the customer, gather their requirements.

● Responding to change, over following a plan

In the waterfall model, everything is planned, i.e., at what time, each phase will be completed. Sometimes you need to implement the new requirements in the middle of the software, so you need to be versatile to make changes in the software.

38. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Authentication	Authorization
Validate users are who they claim to be	Decides what users can or cannot access
Commonly performed before authorization	Commonly performed after authentication
Generally transmits through ID token	Generally transmits through Access token
Governed by OpenID Connect (OIDC) protocol	Governed by OAuth framework
Example : User is required to authenticate before accessing Unix server	Example : After User authenticates Unix server , the system decides what objects or information user are allowed to access

39. What is the procedure for GUI Testing?

- Build the model
- Determine Inputs for the model
- Calculate expected output for the model
- Run the Tests
- Compare the actual output with the expected output
- Decision on further action on the model
- Some of the modeling techniques from which test cases can be derived:
 - Charts – Depicts the state of a system and checks the state after some input.
 - Decision Tables – Tables used to determine results for each input applied

40. write HLR and test cases.

Instagram	<u>CLICK HERE</u>
Facebook	<u>CLICK HERE</u>
ART OF TESTING	<u>CLICK HERE</u>
WHATAPP	<u>CLICK HERE</u>

41. WRITE TEST SCENARIOS.

PEN
FAN
ELEVATOR
DOOR
PEN STAND
ATM
MICROWAVE OWEN
EMAIL RECIEVING
FLIPKART
WHATAPP PAYMENT
CHAIR
COFEE VENDING MACHINE
WRIST WATCH
WHATAPP CHAT MESSAGE
WHATAPP PAYMENT

[Click HERE](#)