Software Testing Assignment - Module­­\_2

# **1.What is Exploratory Testing?**

Exploratory testing is an unscripted, hands-on approach to testing software, where a tester :-

* Learns about the software's features and functionality
* Designs and executes tests simultaneously
* Explores the software to identify defects and issues

# Uses their skills, experience, and creativity to test the software

# **2.What is traceability matrix?**

Traceability matrix is a document that demonstrates the relationship between requirements and other Artifacts.

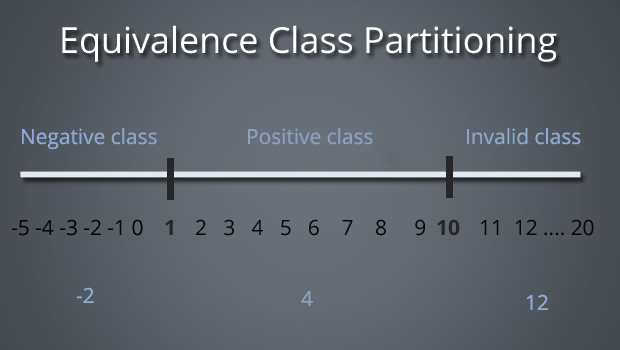
It's used to prove that requirements have been fulfilled. And it typically documents requirements, tests, test results, and issues.

# **3.What is Boundary value testing?**



* Boundary-value analysis is a software testing technique in which tests are designed to include representatives of boundary values in a range. This is the part of Black Box Testing Methods.

# **4.What is Equivalence partitioning testing?**



* This is the method of black box testing that is use to drive the long-range inputs into equivalence partition and pick one representative from the partition for checking the hole range.

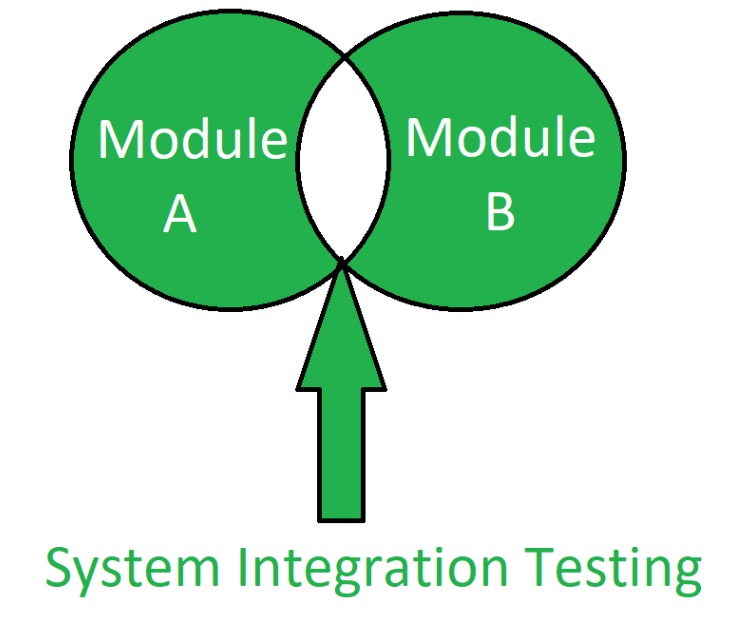
Ex. -Input field for:

- Partition 1: 0-5

- Partition 2: 6-10

- Partition 3: 11-14

**5.What is Integration testing?**



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

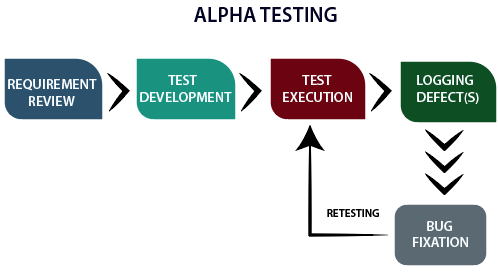
**6.What determines the level of risk?**

🡪 there are two types of risk :-

1.project risk :- product is is development phase

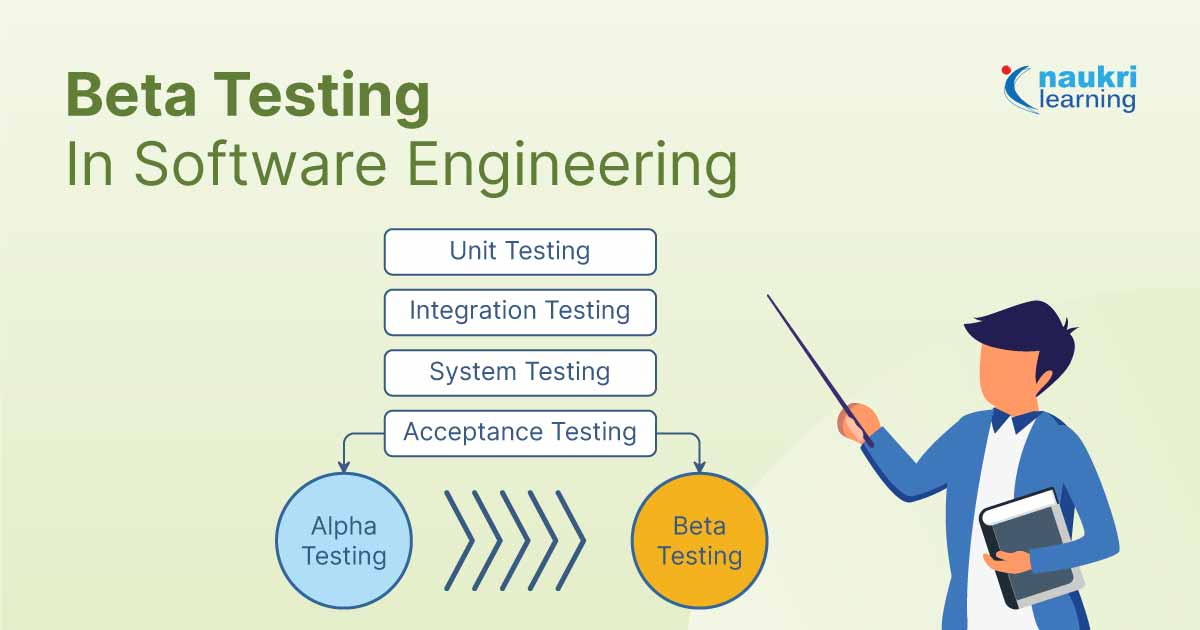
2.product risk :- .risk can be faced while the product is in production phase.

# **7.What is Alpha testing?**



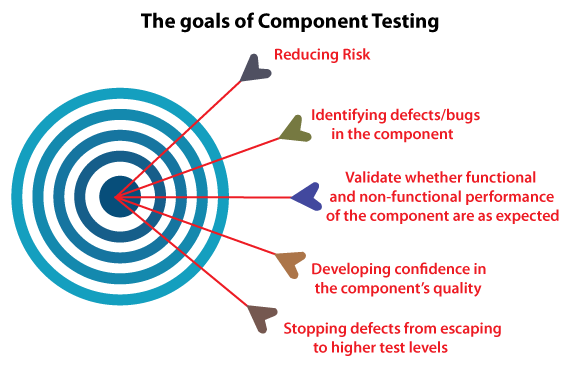
* Alpha testing is a critical phase of software testing conducted by the development or QA team before beta testing. The purpose of alpha testing is to identify and resolve critical bugs and issues in the software before it is released to the public.

# **8.What is beta testing?**



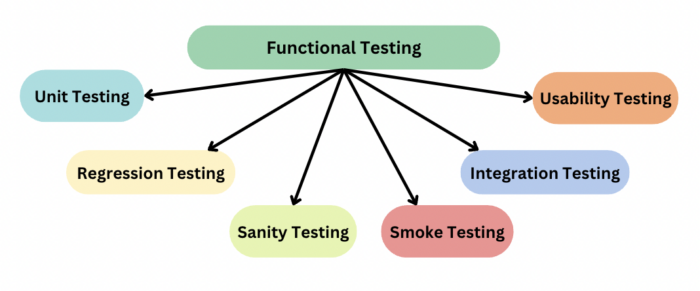
* Beta testing is the last phase of the testing process before a product is released. The purpose of this phase is to evaluate the level of customer satisfaction with the product.

# **9.What is component testing?**



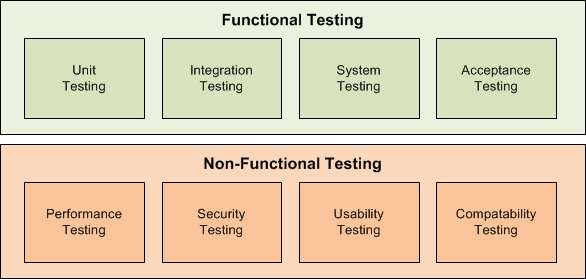
* Component testing, also known as unit testing or module testing, is a level of software testing that focuses on verifying the individual components or units of a system.

# **10.What is functional system testing?**



* Functional system testing meets the functional requirements and works as expected. It involves testing the system's functionality, user interactions, and business processes from start to finish.

# **11.What is Non-Functional Testing?**



* Non-functional testing is a type of software testing that evaluates the software's characteristics, properties, and behaviors that are not related to its functional requirements.

# **12.What is GUI Testing?**



* GUI testing is a software testing technique that checks the Graphical User Interface of the software application to ensure the application's functionality and features meet the business requirements. Some of the common things that are checked in GUI testing include like Size, position and width of the images, Error messages, Sections on screen, Font and color of the text, Alignment of the text and images etc.

# **13.What is Ad-hoc testing?**

Ad-hoc testing is informal testing technique type. Main aim of this testing is find the defect with random checking and brake the system.

* **Type of ad-hoc testing:-**

1. Buddy testing
2. Pair tasting
3. Monkey testing

# **14.What is load testing?**

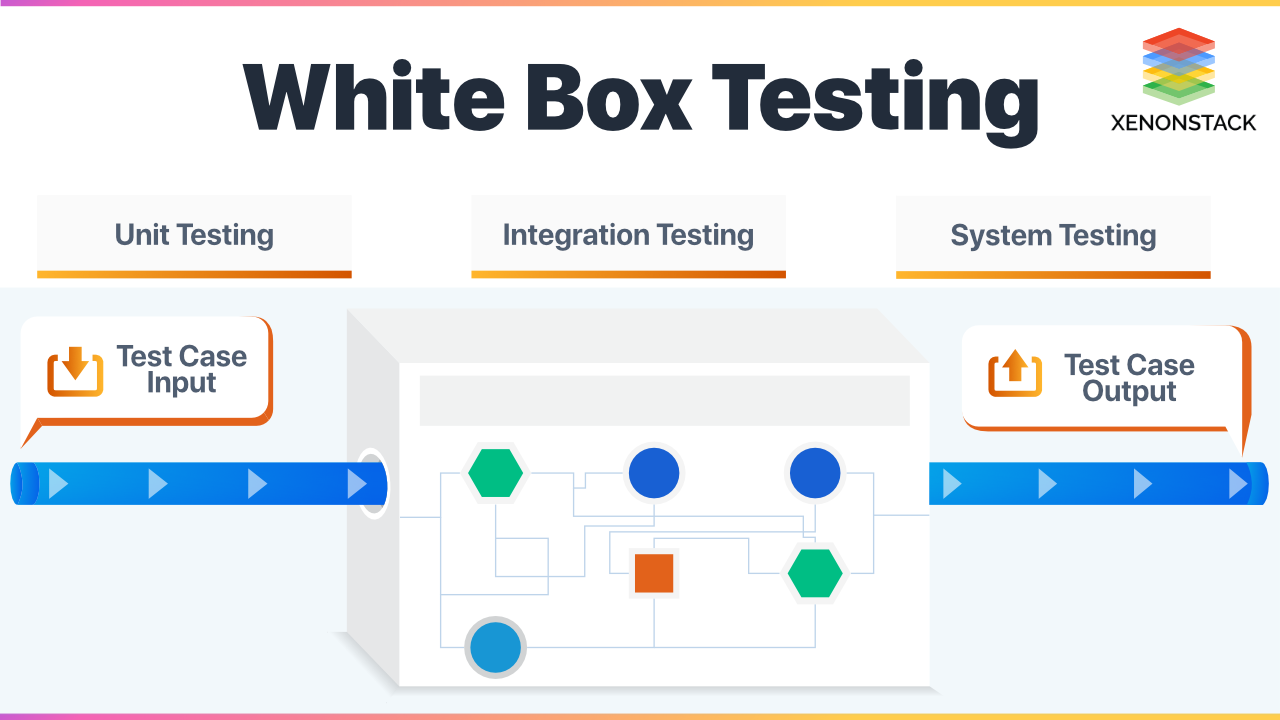
This a type of performance testing. that determines how well a system, software product or application performs under real-life load conditions.

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

# **15.What is stress Testing?**

Stress testing is a type of software testing that helps to ensure the stability and reliability of a system. tress testing is performed until the system fails or crashes, which helps to identify the weak points and vulnerabilities of the system.

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



White box testing is approach in which testing is based on an analysis of the internal structure. White box testing analyzes the internal code, internal design and working functionality. White box testing is also known as transparent testing an open ox testing.

There are three methods of black box testing:

1. **Statement coverage:** Statement coverage also known as line coverage and segment coverage. This method covers only true condition
2. **Decision/Branch coverage:** This method covers the both conditions true and false.
3. **Condition coverage:** Test all conditions (true/false) in each decision point.

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

Black box testing, to check the application without having any knowledge of the internal part / source code. In black-box testing the tester is concentrating on what the software does, not how it does it.

There are four methods of black box testing.

1. **Equivalence partitioning (EP):** It is a method to drive the long-range inputs into equivalence partition and pick one representative from the partition for checking the hole range.
2. **Boundary value analysis:** To check the boundary values with valid range or invalid range.
3. **Decision tables:** Boolean expressions: true or false

Decision table shows the relationship between inputs and possible outputs are mapped together.

1. **State transaction/transaction testing:** To test the system by all given transaction stored in finite state machine.

**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.

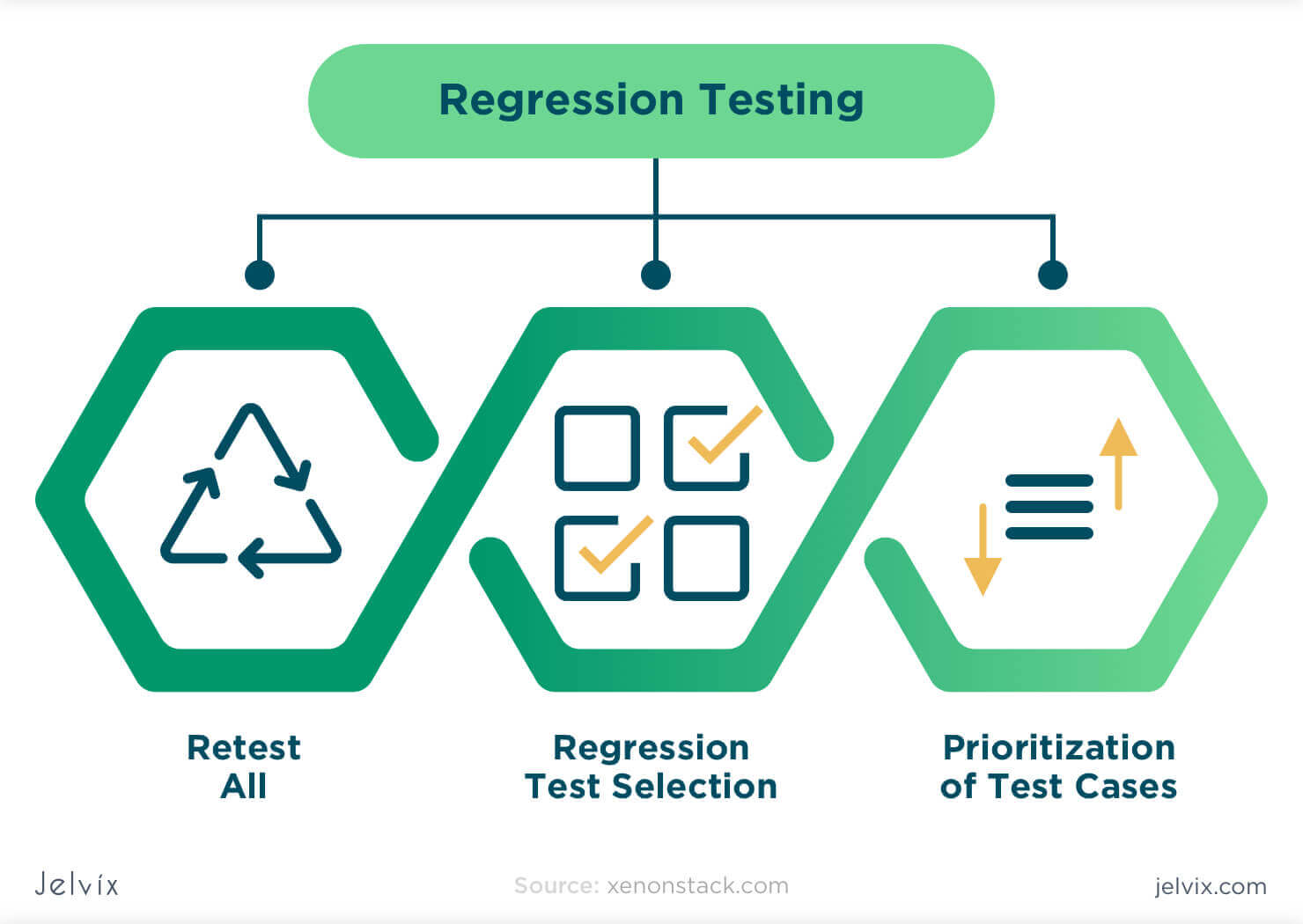
**19. Mention what big-bang testing is?**

🡪 That is a part of Integration Testing. Big Bang testing is a testing approach where all individual components or modules of a software application are tested together in a single.

**20. What is the purpose of exit criteria?**

🡪 The main purpose of exit criteria is all condition must be full fill before leaving the project. Ensure that the project has met its objectives and requirements.

**21. When should "Regression Testing" be performed?**



🡪 Regression testing should be performed in the following scenarios:

* When new functionality is added to the system
* When some defect is identified in the software and the code is debugged to fix it
* When the code is modified to optimize its working
* Whenever the production code is modified
* When a website's new feature is added
* When a bug is fixed by developers
* When there is an update in the database from one software to another

# **22.What is 7 key principles? Explain in detail.**

The 7 Key Principles of Software Testing are fundamental guidelines that ensure effective testing and quality assurance. These principles are applied in the software testing industry. Here's a detailed explanation of each principles.

## **Testing Shows the Presence of Defects:** Testing can only expose the existence of defects. This principle said that testing is not a proof of correctness, but rather a way to identify errors.

1. Exhaustive Testing is Impossible**:** It's impossible to test every possible scenario, input, or condition. It’s time consuming.
2. **Early Testing:** The earlier you test, the cheaper it is to fix defects. This principle highlights the importance of testing early in the development cycle to reduce costs and improve quality.
3. **Defects Clustering:** A few specific areas of the software have a high concentration of bugs, defects and error.
4. **The Pesticide paradox:** Repeatedly using the same test cases can lead to diminishing returns.
5. **Testing is Context-Dependent**: Testing strategies and techniques vary depending on the project, technology, and requirements
6. **Absence of errors is a fallacy**: Even if no defects are found, it doesn't mean the software is error-free. Defect free software is not as important as meets user requirement important.

# **23.Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Tester** |
| Focused on preventing defect and ensuring quality throughout the software development process. | Focuses on detecting and correcting defects in the software product. | Focuses on actual testing. |
| Quality Assurance (QA) is a process-oriented approach to ensuring that a product or service meets a set of quality standards | Quality Control (QC) is a product-oriented approach that focuses on identifying and fixing defects in the final product. | It includes activities that ensure the identification of bugs/error/defects in a software. |
| Involves testing, inspection, and validation to ensure the software meets the required standards. | Involves planning, designing, and implementing processes and procedures to ensure quality. | Product-oriented activities. |
| QA is process oriented | QC is product oriented. | Product-oriented activities. |
| QA is a managerial tool. | QC is a corrective tool. |  |

# **24.Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| **Smoke testing** | **Sanity testing** |
| To check the stability of the system. | To check the rationality of the system. |
| Smoke testing is scripted, documented, well-planned according to STLC | Sanity testing is unscripted, not documented, not well-planned. |
| This is the subset of acceptance testing | This is the subset of the regression testing. |
| That is performed by the developer & tester. | That is performed by tester only. |
| Ex. -To check the login functionality works, and the application doesn't crash or produce errors. | Ex. -To check the login functionality to work as expected with a focus on the logical flow and expected behavior. |

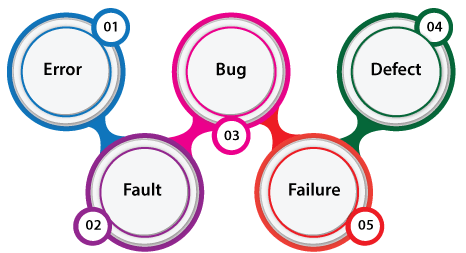
# **25.Difference between verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Verification does not involve code execution | Validation involves code execution. |
| The verifying process includes checking documents, design, code, and program | Validation is a dynamic mechanism of testing and validating the actual product |
| It finds bugs early in the development cycle | It can find bug that the verification process can’t find the bug |

**26. Explain types of Performance testing?**

* Performance testing is a type of software testing that evaluates the performance of a system or application under various conditions, such as:
  + **Load:** Testing with a large number of users or transactions
  + **Stress:** Testing beyond normal limits to identify breaking points
  + **Endurance:** Testing for prolonged periods to evaluate stability
  + **Scalability:** Testing with increased load or users to evaluate performance

**27. What is Error, Defect, Bug and failure?**



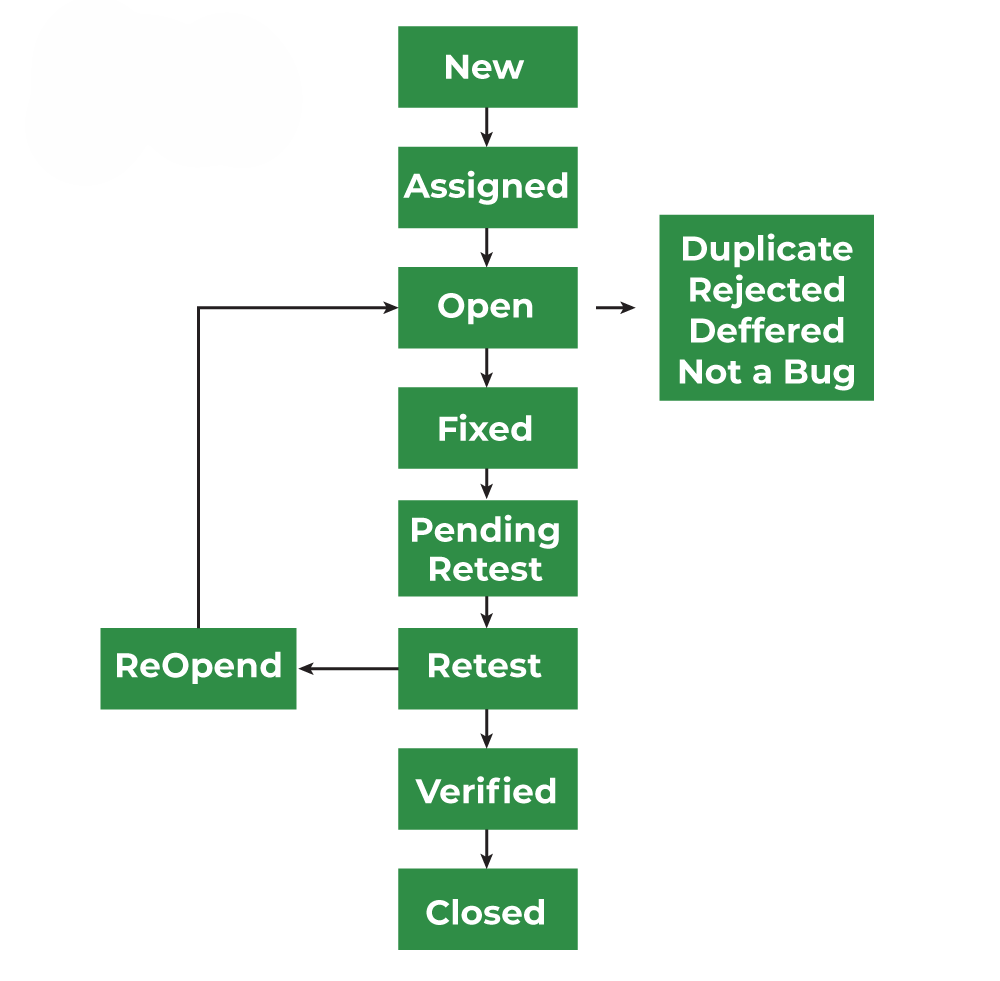
* Error: A mistake in coding is error.
* Defect: An error found by tester is defect.
* Bug: Defect accepted by development team is bug.
* Failure: Build does not meet the requirements is failure.

**28. Difference between Priority and Severity?**

|  |  |  |
| --- | --- | --- |
|  | Severity | Priority |
| 1 | Refers to the **impact** of the defect on the system’s functionality. | Refers to the **urgency** or **importance** of fixing the defect. |
| 2 | Technical impact and functionality of the defect. | Business impact and urgency of fixing the defect. |
| 3 | How serious the defect is in terms of system behavior. | How quickly the defect needs to be addressed. |
| 4 | - System crash- Missing functionality- Incorrect calculations | - Feature affecting customer experience- Critical bug before release- Client's urgent request |
| 5 | Affects the system’s core functionality (e.g., crash, data loss). | Can be urgent based on business needs (e.g., customer complaints, upcoming release). |
| 6 | - High- Medium- Low | - High- Medium- Low |
| 7 | Severity is independent of business needs. | Priority is determined by business needs, irrespective of severity. |

**29.** **What is Bug Life Cycle?**

🡪 The duration or timespan between the first time defect found and the time when defect closed, rejected, differed or postponed is called bug life cycle.



**30. Explain the difference between Functional testing and Non-Functional testing?**

|  |  |
| --- | --- |
| **Functional testing** | **Non-Functional testing** |
| Ensure the software performs its intended functions correctly. | Evaluate the software's performance, security, usability, and other non-functional aspects. |
| Easy to do manual & automation with Functional Testing. | Tough to do manual testing with Non-functional testing. |
| First to execute Functional. | Non-functional should be executed after Functional testing. |
| Functional testing describes what the product does. | Nonfunctional testing describes how good the  product works. |

**31. Explain the difference between STLC & SDLC ?**

|  |  |
| --- | --- |
| **STLC** | **SDLC** |
| Software Testing  Life Cycle | Software Development Life Cycle |
| **Phases of STLC :-**  **1.Requirnment Analysis**  **2.Test Planning**  **3.Test Case Development**  **4.Test Environment Set-up**  **5.Test Execution**  **6.Test Cycle Closer** | **Phases of SDLC :-**   1. **Planning** 2. **Analysis** 3. **Designing** 4. **Coding** 5. **Testing** 6. **maintenance** |
| **Focused on Software Testing** | **Focused on Software Development** |
| **Helps to make software defect free** | **Helps to develop good quality software** |
| **STLC phases are performed after SDLC phases** | **SDLC phases are completed before STLC phases** |
| **QA team defines the test plan** | **Coder create a well organized development plan** |
| Software Testing  Life Cycle | Software Development Life Cycle |

32.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 action executed to verify particular feature or functionality. | Is a set of instructions to test an app automatically. |
| Helps to test end to end functionality in an Agile  Way. | Helps in executing testing an app. | Helps to test specific things repeatedly. |
| Includes an end-to-end functionality to be tested. | Includes test step, data, expected result for testing. | Includes different commands to develop a script. |
| Allow quickly assessing the testing scope. | Allow detecting error and defects. | Allowing carrying out an automatic execution of test case. |

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

A Test Plan is a detailed document outlining the approach, scope, and timeline for testing a software application or system. It serves as a guide for the testing process, ensuring that the testing is thorough, efficient, and effective.

A comprehensive Test Plan should cover the following information:

1. **Test Objectives:** Clearly state the goals and objectives of testing.
2. **Scope:** Define what is included and excluded from testing.
3. **Test Environment:** Describe the hardware, software, and network settings for testing.
4. **Test Approach:** Outline the testing methodology, techniques, and tools.
5. **Test Cases:** List the specific test scenarios and expected results.
6. **Test Data:** Identify the data required for testing and how it will be obtained.
7. **Test Schedule:** Provide a timeline for testing, including milestones and deadlines.
8. **Resources:** Identify the personnel, equipment, and budget required for testing.
9. **Risks and Assumptions:** Document potential risks and assumptions made during testing.
10. **Test Deliverables:** Define the outputs and artifacts expected from testing, such as test reports and defect logs.
11. **Test Criteria:** Establish the criteria for determining when testing is complete and successful.

34.What is priority?

Priority is defined as a parameter that decides the order in which a defect should be fixed. Defects having a higher priority should be fixed first.

* Defects/ bugs that leave the software unstable and unusable are given higher priority over the defects that cause a small functionality of the software to fail.
* It refers to how quickly the defect should be rectified.
* **Types of Priorities:**
* Priority in software testing can be divided into 3 categories:
* **Low:** The defect is irritant but a repair can be done once the more serious defects can be fixed.
* **Medium:** The defect should be resolved during the normal course of the development but it can wait until a new version is created.
* **High:** The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed.

35.What is severity?

Severity is defined as the extent to which a particular defect can create an impact on the software. Severity is a parameter to denote the implication and the impact of the defect on the functionality of the software.

* A higher effect of the bug on system functionality will lead to a higher severity level.
* A QA engineer determines the severity level of a bug.
* **Types of Severity:**
* Severity in software testing can be classified into 4 categories:
* **Critical:** This severity level implies that the process has been completely shut off and no further action can be taken.
* **Major:** This is a significant flaw that causes the system to fail. However, certain parts of the system remain functional.
* **Medium:** This flaw results in unfavorable behavior but the system remains functioning.
* **Low:** This type of flaw won’t cause any major breakdown in the system.

36.Bug categories are…

1. **Functional Bugs:**
2. **Performance Bugs:**
3. **Security Bugs:**
4. **Unit-level bugs:**
5. **Usability Bugs:**
6. **Syntax Errors:**
7. **Compatibility Errors:**
8. **Logic Bugs**

**37.Advantage of Bugzilla**

**Advantages of Bugzilla :-**

* **Deadlines: To fix the bugs, deadlines can be established.**
* **Types: It reports in a variety of formats and types.**
* **Request System: You can use the ‘request system’ provided by Bugzilla to ask other users to**
  + **evaluate codes, provide information, and other things.**
* **Flexible: Bugzilla is quite flexible, so you can modify it to fit your unique process and requirements.**
* **Bug tracking tool: Bugzilla is extremely good at monitoring and handling bugs and issues**

38.Difference between Priority and Severity

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Refers to the impact or potential impact of a defect on the system, user, or business. | Refers to the order in which defects should be addressed or fixed |
| Measures the potential damage or risk caused by the defect. | Determines the urgency and importance of resolving the defect. |
| Typically categorized as:  - Critical (High)  - Major (Medium)  - Minor (Low) | Typically categorized as:  - High (Must-fix, critical)  - Medium (Should-fix, important)  - Low (Nice-to-fix, minor) |

39.When to used Usability Testing?

* Your work isn't done despite testing throughout the design and development cycle. You’ll want to continue usability testing after a product or feature goes live to optimize consistently. This is how organizations create unique experiences that stand the test of time as user behavior evolves.

40.What is the procedure for GUI Testing?

The procedure for GUI testing includes the following steps:

* **Identify components**: Identify the GUI components and what needs to be tested
* **Check visuals**: Verify the visual aspects of the application or website
* **Write test cases**: Write test cases to verify the working and style of the GUI components
* **Automate and test**: Automate repetitive test cases and manually test the ones that can't be automated
* **Report and retest**: Report the defects and retest again

|  |  |  |
| --- | --- | --- |
| 41 | Write a scenario of only Whatsapp chat message | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 42 | Write a scenario of Pen | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 43 | Write a scenario of Pen Stand | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 44 | Write a scenario of Door | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 45 | Write a scenario of ATM | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 46 | Write a scenario of Microwave Owen | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 47 | Write a scenario Coffee Vending Machine | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 48 | Write a scenario of Chair | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 49 | Write a scenario of Whatsapp Payment | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 50 | Write a scenario of G-mail receiving mail | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 51 | Write a scenario of Whatsapp Group Chat | [Click here](https://docs.google.com/spreadsheets/d/1S60zemPUFpa-BVOiuTB7Gd3JCEoVQVJ-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 52 | Write a HLR , TEST CASES of Whatsapp Web | [Click here](https://docs.google.com/spreadsheets/d/15eDnTnZjXJl-0lxb-KFYKeWb_dhKCagJ/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 53 | Write a HLR , TEST CASES of Instagram | [Click here](https://docs.google.com/spreadsheets/d/1EKvUwoicP85krfohpNGtg8vn5XTM3Ps4/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 54 | Write a HLR , TEST CASES of Facebook | [Click here](https://docs.google.com/spreadsheets/d/1BmqBCYsGM56T5xCi-yz_42WlJfOvlCq-/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |
| 55 | Write a HLR , TEST CASES of Art Of Testing | [Click here](https://docs.google.com/spreadsheets/d/1HbBF-1a-a7p_QQknqasjJgKg5-MAOvIF/edit?usp=sharing&ouid=107657968599494976437&rtpof=true&sd=true) |