1. **What is integration testing?**

* Integration testing checks if different parts of your application work well together when combined. It ensures they communicate and function correctly as a whole.

1. **What is functional testing?**

* Functional testing checks if the software works properly by testing its features. It ensures the system does what it is supposed to do.

1. **What is Non-Functional Testing?**

* Non-functional testing checks how well the software works rather than what it does. It tests performance, speed, reliability, and usability of the system.

1. **What is load testing?**

* Load testing checks how the software behaves when many users use it at the same time. It ensures the system can handle high traffic without issues.

1. **What is stress testing?**

* Stress testing checks how the software performs under extreme conditions, like very high traffic or limited resources. It helps find the breaking point of the system.

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

* White box testing techniques analyze the internal structures the used data structures, internal design, code structure, and the working of the software.
* White box testing is primarily performed by software developers who have a deep understanding of the code and its internal structure.
* **Types of White Box Testing:**

1. **Unit Testing** – Checks if each part or function of the application works correctly. Ensures the application meets design requirements during development.
2. **Integration Testing** – Examines how different parts of the application work together. Done after unit testing to make sure components work well both alone and together.
3. **Regression Testing** – Verifies that changes or updates don’t break existing functionality. Ensures the application still passes all existing tests after updates.
4. **What is black box testing? What are the different black box testing techniques?**

* Black box testing is a software testing technique that evaluates a system's functionality without knowing its internal workings.
* The term "black box" refers to the system's unknown internal mechanics.
* **black box testing techniques include:-**

1. **Equivalence Partitioning -** Dividing input data into valid and invalid partitions to test each group.
2. **Boundary Value Analysis -** Testing at the edges or boundaries of input values to find errors.
3. **Decision Table Testing** - Using decision tables to test different input combinations and their expected outcomes.
4. **State Transition Testing** - Testing the system’s behavior based on different states or conditions.
5. **Error Guessing** - Making educated guesses to test areas where errors are likely to occur.
6. **Difference between QA v/s QC v/s Tester**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Aspect** | **QA (Quality Assurance)** | **QC (Quality Control)** | |  | | --- | |  |  |  | | --- | | **Tester** | |
| **Focus** | Process improvement and defect prevention | Identifying and fixing defects in the product | Performing tests to find bugs or issues in the software |
| **Approach** | Proactive (prevents defects) | Reactive (finds defects) | Testing the system based on requirements and expected behavior |
| **Activities** | Creating standards, guidelines, and processes | Reviewing and inspecting the final product | Running test cases, reporting bugs |
| **Goal** | Ensures the right processes are followed | Ensures the product meets quality standards | Identifies defects and ensures the product works correctly |
| **Role** | Involves the entire development process | Involves product evaluation after development | Focuses specifically on testing the software |

1. **Difference between Smoke and Sanity?**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Smoke Testing** | **Sanity Testing** |
| **Purpose** | To check if the basic functionalities of the system are working after a build or update. | To verify if a specific issue has been fixed or if a particular functionality works as expected. |
| **Scope** | Broad and shallow testing of major functions to ensure the system is stable enough for further testing. | Narrow and focused testing of specific areas that are impacted by changes. |
| **When** | Done at the initial stage of testing, usually after a new build is received. | Done after receiving a new build, specifically to check if the bug or issue has been fixed. |
| **Depth** | Shallow and quick, without going into much detail. | Deeper than smoke testing but focused on a specific issue or functionality. |
| **Test Coverage** | Covers major areas of the system but not in detail. | Tests only the areas affected by recent changes or bug fixes. |
| **Goal** | To determine if the software is stable enough for further testing. | To confirm that specific changes or bug fixes work as expected. |

1. **Difference between verification and Validation**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Verification** | **Validation** |
| **Purpose** | Ensures the product is being built correctly (process-oriented). | Ensures the product meets user requirements (product-oriented). |
| **Focus** | Focuses on processes, documents, and plans. | Focuses on the actual product and its functionality. |
| **When** | Done during the development phase. | Done after development, during or after testing. |
| **Type** | Static testing (reviews, inspections). | Dynamic testing (execution of test cases). |
| **Goal** | To check if the product is designed according to specifications. | To check if the product fulfills user needs and works as expected. |
| **Example** | Reviewing documents, designs, or code. | Running test cases to ensure the software works correctly. |

1. **What is error, defect, bug and failure?**
2. **Error:** A mistake made by the developer while coding or designing.
3. **Defect**: A flaw in the software that does not meet the specified requirements.
4. **Bug**: A defect identified during testing that causes unexpected behaviour.
5. **Failure**: When the software does not work correctly during actual use.
6. **Difference between Priority and Severity**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Priority** | **Severity** |
| **Definition** | Defines how quickly an issue needs to be fixed. | Defines how serious the issue is in terms of impact on the system. |
| **Focus** | Focuses on the urgency of fixing the issue. | Focuses on the impact of the issue on functionality. |
| **Decided By** | Decided by the business team or project manager. | Decided by the testing team based on functionality impact. |
| **Example** | A typo on the homepage is a **high priority** but **low severity** issue. | A system crash is a **high severity** but might be **low priority** if it occurs rarely. |

1. **Write a Scenario of Pen**

 Verify the material of the outer body of the pen (e.g., metallic, plastic) as per the specifications.

 Check if the outer body color matches the requirement specifications.

 Verify that the brand name and logo are clearly visible and legible on the pen.

 Verify that any information printed on the pen is readable and does not fade easily.

 Verify the type of pen (ballpoint, ink, or gel pen) as mentioned in the specifications.

 Verify that the pen writes clearly on different types of paper (plain, rough, glossy).

 Check whether the pen has a cap or is a retractable design.

 Verify the ink color of the pen as per the given specifications.

 Test if the pen writes smoothly on other surfaces like cardboard or rubber.

 Verify that the pen’s ink does not leak when tilted upside down.

* Verify that the pen’s ink does not leak at higher altitudes.

 Verify whether the text written by the pen can be erased, if applicable.

 Test the functionality of the pen under normal writing pressure.

 Verify that the text written by the pen does not fade before the time specified.

 Verify that the pen writes normally when tilted at an angle rather than held straight.

 Check if the grip of the pen provides enough friction for comfortable handling.

 In the case of a ballpoint pen, verify the tip size as per the specifications.

 For ball and gel pens, verify that the refill can be easily replaced by the user.

1. **Write a Scenario of Pen Stand**

 Check what **material** is used to make the pen stand.

 Verify how much **pressure** the pen stand can handle when a pen is placed on it.

 Check the **size and shape** of the pen stand to ensure it matches the requirements.

 Verify how much **weight** the pen stand can hold without breaking or bending.

 Test if the stand can hold the **maximum number of pens** specified by the manufacturer.

 Expose the pen stand to **extreme temperatures** (very hot or very cold) to see if it gets damaged.

 Test the pen stand in **high humidity** to check if the material gets affected.

 Verify if the pen stand can stand properly on **different surfaces** like smooth, uneven, or inclined ones.

 Apply slight **force** to the pen stand to see if it moves or stays in place.

1. **Write a Scenario of Door**

* Verify if the door is single door or bi-folded door.
* Check if the door opens inwards or outwards.
* Verify that the dimension of the doors are as per the specifications.
* Verify that the material used in the door body and its parts is as per the specifications.
* Verify that color of the door is as specified.
* Verify if the door is sliding door or rotating door.

* Check the position, quality and strength of hinges.
* Check the type of locks in the door.
* Check the number of locks in the door interior side or exterior side.
* Verify if the door is having peek-hole or not.
* Verify if the door is having stopper or not.
* Verify if the door closes automatically or not – spring mechanism.
* Verify if the door makes noise when opened or closed.
* Check the door condition when used extensively with water.
* Check the door condition in different climatic conditions- temperature, humidity etc.
* Check the amount of force- pull or push required to open or close the door.

1. **Write a Scenario of ATM**

 Verify the ATM Card Insertion Slot is as per the specification.

 Verify that the ATM machine accepts a valid card and prompts for PIN details.

 Verify error message when an incorrect card is inserted (e.g., expired card).

 Verify error message when an incorrect PIN is entered.

 Verify that the user is asked to enter the PIN after inserting a valid ATM card.

 Verify that the PIN is encrypted for security purposes.

 Verify that the card gets blocked after exceeding the maximum number of incorrect PIN attempts.

 Verify that the user can only perform one cash withdrawal per PIN request.

 Verify the ATM machine logs out immediately after a successful withdrawal.

 Verify the message displayed when there is insufficient balance in the ATM.

 Verify the cash withdrawal functionality by entering a valid amount.

 Verify the machine correctly dispenses cash after the transaction.

 Verify the receipt printer functionality and ensure the printed data is correct.

 Verify the functionality of the screen and buttons, ensuring they are clearly visible and responsive.

 Verify the machine logs out properly after a session, ensuring no data leakage.

1. **When to used Usablity Testing?**

* During this testing validates application provided context sensitive help or not to the user.
* Checks how easily the end users are able to understand and operate the application is called usability testing.

1. **What is the procedure for GUI Testing?**

* GUI testing involves systematically checking the visual elements of a software application, like buttons, menus, text fields, and images, to ensure they function correctly, are visually appealing, and provide a user-friendly experience by verifying their size, position, alignment, color, text content, and behavior under different interactions across various devices and browsers.
* **Identify test areas:** Define which elements of the GUI need to be tested, including buttons, text boxes, dropdowns, icons, images, and navigation menus.
* **Functional testing:** Verify that each GUI element performs its intended function correctly, like clicking buttons triggers the appropriate action, text fields accept valid input, and error messages display accurately.
* **Navigation testing:** Ensure users can easily navigate through different screens and sections of the application using menus, links, and buttons.
* **Accessibility testing:** Evaluate if the GUI adheres to accessibility guidelines, allowing users with disabilities to interact with the application using assistive technologies.
* **Performance testing:** Assess the responsiveness of the GUI under different load conditions to check for delays or lags in interaction.

1. **Write a scenario of chair**

 Verify that the chair is stable enough to support the average human load.

 Check the material used in making the chair (e.g., wood, plastic) to ensure it meets the specifications.

 Check if the chair’s legs are level to the floor to ensure proper stability.

 Check the usability of the chair for different purposes, such as an office chair or normal household chair.

 Check if there is back support in the chair to ensure comfort for the user.

 Check if there is support for hands in the chair (e.g., armrests) for better comfort.

 Verify the paint’s type and color to make sure it is as per the specifications.

 Verify if the chair’s material is brittle or not, ensuring it’s durable and strong.

 Check if the chair comes with a cushion for added comfort.

 Check the condition of the chair when washed with water or see the effect of water on the chair’s material.

 Verify that the dimension of the chair is as per the specifications, ensuring it fits well in the intended space.

 Verify that the weight of the chair is as per the specifications to ensure it's manageable and meets requirements.