**Module–2(Manual Testing)**

**• What is Exploratory Testing?**

**Exploratory testing is a software testing approach where testers actively explore the application, learning about it as they go, and designing and executing tests simultaneously, without relying on pre-defined scripts or test cases.**

**Exploratory Testing is a type of**[**software testing**](https://www.geeksforgeeks.org/software-testing-basics/)**in which the tester is free to select any possible methodology to test the software.**

**• What is traceability matrix?**

**A document that maps and tracks the relationships between different project artifacts, like requirements, design elements, test cases, and defects, ensuring that all requirements are addressed and verified throughout the project lifecycle.**

**• 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.**

**• What is Equivalence partitioning testing?**

**Equivalence partitioning (EP) is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived.**

**• What is Integration testing?**

**Integration 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.**

**• What determines the level of risk?**

**A factor that could result in future negative consequences; usually expressed as impact and likelihood’**

**Risks are of two types:**

**1:-Project Risks**

**2:-Product Risks**

**• What is Alpha testing?**

**It is always performed by the developers at the software development site.**

**Alpha testing combines white-box and black-box testing techniques to explore and evaluate the software.**

**It aims to identify and fix bugs in a controlled environment that simulates real-world conditions. This helps ensure the software’s functionality, reliability , and stability .**

**• What is beta testing?**

**Beta testing is a pre-release phase where a select group of users, representing the target audience, test a near-final version of a product (like software or a mobile app) to identify bugs, usability issues, and gather feedback for improvement before the official launch.**

**It is performed in Real Time Environment**.

**It is also the form of Acceptance Testing**

**It is only a kind of Black Box Testing.**

**It is always performed at the user’s premises in the absence of the development team.**

**Beta testing can be considered “pre-release” testing.**

**• What is component testing?**

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

**Component Testing – The testing of individual software components.**

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

**Sometimes known as Unit Testing, Module Testing or Program Testing.**

**Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.**

**• What is functional system testing?**

**Functional System Testing : A requirement that specifies a function that a system or system component must perform.**

**• What is Non-Functional Testing?**

**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.**

**• What is GUI Testing?**

**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.**

**• What is Adhoc testing?**

**Adhoc testing can be achieved with the testing technique called Error Guessing.**

**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.**

**In** **fact is does not create test cases altogether.**

**• What is load testing?**

**Load testing is a type of performance testing that simulates real-world user traffic to determine how a system or application behaves under normal and peak loads, identifying potential performance bottlenecks and ensuring stability and efficiency.**

**• What is stress Testing?**

**It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.**

**Stress Testing is done to make sure that the system would not crash under crunch situations.**

**Stress testing is also known as endurance testing.**

**• 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.**

**Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.**

**• What is black box testing? What are the different black box testing techniques?**

**Black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.**

**Specification-based testing technique is also known as ‘black-box’ or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.**

**The testers have no knowledge of how the system or component is structured inside the box.**

**There are black-box technique:**

**1:Equivalence partitioning**

**2:Boundary value analysis**

**3:Decision tables**

**4:State transition testing**

**5:Use-case Testing**

**6:Other Black Box Testing**

**• Mention what are the categories of defects?**

**Defects are generally categorized into three main types: minor, major, and critical. These categories are based on the severity and impact of the defect on the product's functionality, safety, and usability.**

**• 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.**

**• 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**

• **When should "Regression Testing" be performed?**

**Regression Testing: Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed**.

**• What is 7 key principles? Explain in detail?**

**"Seven key principles" can refer to different sets of fundamental ideas or guidelines depending on the context, such as in quality management, software testing, or internal audit, each with its own specific principles.**

**1. Testing shows presence of Defects**

**Testing can show that defects are present, but cannot prove that there are no defects.**

**Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.**

**2. Exhaustive Testing is Impossible!**

**Testing everything including all combinations of inputs and preconditions is not possible.**

**For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.**

**3.Early Testing**

**Testing activities should start as early as possible in the development life cycle.**

**Early testing means starting the testing process at the beginning of the SDLC, rather than waiting until the end.**

**4.Defect Clustering”**

**small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.**

**5. Pesticide Paradox**

**If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.**

**To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects**.

**6.Testing is Context Dependent**

**Testing is basically context dependent.**

**Testing is done differently in different contexts**

**Different kinds of sites are tested differently. For example**

**Safety – critical software is tested differently from an e-commerce site.**

**7. Absence of Errors Fallacy**

**If the system built is unusable and does not fulfill the user’s needs and expectations then finding and fixing defects does not help.**

**If we build a system and, in doing so, find and fix defects.**

**• Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Testing** |
| **Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.** | **Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.** | **Activities which ensure the identification of bugs/error/defects in the Software.** |
| **Focuses on processes and procedures rather than conducting actual testing on the system.** | **Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.** | **Focuses on actual testing.** |
| **Process oriented activities.** | **Product oriented activities.** | **Product oriented activities.** |
| **Preventive activities.** | **It is a corrective process.** | **It is a preventive process** |
| **It is a subset of Software Test Life Cycle (STLC).** | **QC can be considered as the subset of Quality Assurance** | **Testing is the subset of Quality Control.** |

**• Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| **Smoke** | **Sanity** |
| **Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine** | **Sanity Testing is done to check the new functionality / bugs have been fixed** |
| **The objective of this testing is to verify "stability" of the system in order to the with more rigorous testing** | **The objective of the testing is to verify the “rationality" of the system in order proceed to proceed with more rigorous testing** |
| **This testing is performed by the developers or testers** | **Sanity testing is usually performed by testers** |
| **Smoke testing is usually documented or scripted** | **Sanity testing is usually not documented and is unscripted** |
| **Smoke testing is a subset of Regression testing** | **Sanity testing is a subset of Acceptance testing** |
| **Smoke testing exercises the entire system from end to end** | **Sanity testing exercises only the particular component of the entire system** |
| **Smoke testing is like General Health Check Up** | **Sanity Testing is like specialized health check up** |

**• Difference between verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| **Evaluates the products to check whether it meets the specific requirements of the particular phase** | **Evaluates the products to check whether it meets the business and users’ needs** |
| **Verification does not involve code execution – part of static testing** | **Validation involves code execution – dynamic testing** |
| **The goal of verification is application and software architecture and specification** | **The goals of validations is an actual product** |
| **Verification is a part of QA process** | **Validation is a part of part of QC process** |
| **If tests the requirements, architecture ,design and code of the software product** | **If tests the usability, functionalities ,and reliability of the end product** |
| **Involves all the static testing techniques like reviews, walkthroughs, inspection** | **Includes all dynamic testing techniques :functional , non-functional , change related** |
| **If finds bugs early in the development cycle** | **It can find buges that the verification process can not catch** |
| **Performs before validation** | **Perform after validation** |

**• Explain types of Performance testing.**

**⚫Load testing**

**⚫ Stress testing**

**⚫ Endurance testing**

**⚫ Spike testing**

**⚫ Volume testing**

**⚫ Scalability testing**

**• What is Error, Defect, Bug and failure?**

**A mistake in coding is called error.**

**Error found by tester is called defect.**

**Defect accepted by development team then it is called bug,**

**Build does not meet the requirements then it is failure.**

**• Difference between Priority and Severity**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| **Severity determines – The defects effect on the application.** | **Priority determines – The defect urgency of repair.** |
| **How band the defect is QA testers give severity** | **How soon do we need to fix The Test lead of project manager gives priority** |
| **Levels**  **-critical :the software will not run.**  **-High:unexcepected fatal efforts**  **-Medium:a feature is malfunctioning**  **Low :a cosmetic issue** | **Levels:**  **-P1:fix before next build to test**  **-P2:Fix before the final release**  **-P3:We probably won’t get to these,but we want to track the, anyway to divide the priority severity.** |

**• What is Bug Life Cycle?**

**“A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program’s source code or its design.**

**• Explain the difference between Functional testing and Non Functional testing.**

|  |  |
| --- | --- |
| **Functional testing** | **Non Functional testing** |
| **Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.** | **Non-Functional testingchecksthe Performance, reliability, scalability and other non-functional aspects of the software system.** |
| **Functional testing is executed first** | **Non functional testing should be performed after functional testing** |
| **Manual testing or automation tools can be used for functional testing** | **Using tools will be effective for this testing** |
| **Business requirements are the inputs to functional testing** | **Performance parameters like speed , scalability are inputs to non-functional testing** |
| **Functional testing describes what the product does** | **Nonfunctional testing describes how good the product works** |
| **Easy to do manual testing** | **Tough to do manual testing** |

**• Write a scenario of only Whatsapp chat messages**

**Verify that on downloading the Whatsapp application, users can register using a new mobile number.**

**Verify that for a new mobile number user will get a verification code on his mobile and filling in the same verifies the new user account.**

**Check the maximum number of incorrect attempts allowed while filling out the verification code.**

**Verify that registering an existing mobile number for new user account registration is not allowed.**

**Verify that on successful registration all the contacts in the user’s contact directory get imported to the Whatsapp contact list.**

**Verify that the user can set DP and status on Whatsapp.**

**Verify that the user can update the existing DP and Whatsapp status.**

**Verify that the user can send messages to any individual selected from his contact list.**

**Verify that ‘Chats’ window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated.**

**Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person.**

**Verify that the user can check the message delivered and read the time for a message in the ‘Message Info’ section.**

**Verify that the user can share or receive contact with the other person.**

**Verify that the user can create a group by adding multiple people from his contact list.**

**Verify that the user can send and receive the message in group chats.**

**Verify that users can send and receive images, audio, video, and emoticons in the chat with individuals.**

**Verify that users can send and receive images, audio, video, and emoticons in group chats.**

**Verify that the user can send and receive chats in the secondary languages available.**

**Verify that users can delete text, images, audio, and video messages within a chat.**

**Verify that users can clear their complete chat history in an individual or group chat.**

**Verify that users can archive chats in an individual or group chat.**

**Verify that users can block a user to prevent any message from getting received from the blocked contact.**

**Verify that the user makes WhatsApp calls to the person in his contact list.**

**Verify that the user can receive WhatsApp calls from the person in his contact list.**

**Verify that users can mark chats as favorites and access all chats marked as favorites from the ‘Favorites’ section.**

**• Write a Scenario of Pen**

**Verify that the length and the diameter of the pen are as per the specifications.**

**Verify the outer body material of the pen. Check if it is metallic, plastic, or any other material specified in the requirement specifications.**

**Check the color of the outer body of the pen. It should be as per the specifications.**

**Verify that the brand name and/or logo of the company creating the pen should be clearly visible.**

**Verify that any information displayed on the pen should be legible and clearly visible.**

**• Write a Scenario of Pen Stand**

**Verify the color of pen stand.**

**Check Sixe in stand.**

**Check pettern in stand.**

**Check Type in stand.**

**Check single pen stand or not.**

**Check Design .**

**Check if it is a single pen stand or is it a key stand or something else**

**Is the check wooden or plastic or something else?**

**• 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.**

**• Write a Scenario of ATM**

**Verify the type of ATM machine, if it has a touch screen, both keypad buttons only, or both.**

**Verify that on properly inserting a valid card different banking options appear on the screen.**

**Check that no option to continue and enter credentials is displayed to the user when the card is inserted incorrectly.**

**Verify that the touch of the ATM screen is smooth and operational.**

**Verify that the user is presented with the option to choose a language for further operations.**

**Check that the user is asked to enter a pin number before displaying any card/bank account detail.**

**Verify that there is a limited number of attempts up to which the user is allowed to enter the pin code.**

**Verify that if the total number of incorrect pin attempts gets surpassed then the user is not allowed to continue further. And operations like temporary blocking of the card, etc get initiated.**

**Check that the pin is displayed in masked form when entered.**

**Verify that the user is presented with different account type options like- saving, current, etc.**

**Verify that the user is allowed to get account details like available balance.**

**Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal.**

**Verify that the user is only allowed to enter the amount in multiple denominations as per the specifications.**

**Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured.**

**Check that the user cannot withdraw more amount than the total available balance and a proper message should be displayed.**

**Verify that the user is provided the option to get the transaction details in printed form.  
Verify that the user’s session timeout is maintained.**

**Check that the user is not allowed to exceed one transaction limit amount.**

**Verify that the user is not allowed to exceed the one-day transaction limit amount.**

**Verify that the user is allowed to do only one transaction per pin request.**

**Check that in case the ATM machine runs out of money, a proper message is displayed to the user.**

**Verify that the applicable fee gets deducted along with the withdrawn amount in case the user exceeds the limit of the number of free transactions in a month.**

**Verify that the applicable fee gets deducted along with the withdrawn amount in case the user uses a card of a bank other than that of an ATM.**

**Check that the user is not allowed to proceed with the expired ATM card and that a proper error message gets displayed.**

**Verify that in case of sudden electricity loss before withdrawing cash, the transaction is marked as null and the amount is not withdrawn from the user’s account.**

**• Write a scenario of Microwave Owen**

**Verify that the dimensions of the oven are as per the specification provided.**

**Verify that the oven’s material is optimal for its use as an oven and as per the specification.**

**Verify that the oven heats the food at the desired temperature properly.**

**Verify that the oven heats food at the desired temperature within a specified time duration.**

**Verify the ovens functioning with the maximum attainable temperature.**

**Verify the ovens functioning with minimum attainable temperature.**

**Verify that the oven’s plate rotation speed is optimal and not too high to spill the food kept over it.**

**Verify that the oven’s door gets closed properly.**

**Verify that the oven’s door opens smoothly.**

**Verify that the text written over the oven’s body is clearly readable.**

**Verify that the digital display is clearly visible and functions correctly.**

**Verify that the temperature regulator is smooth to operate.**

**Verify that the temperature regulator works correctly.**

**Check the maximum capacity of the oven and test its functioning with that volume of food.**

**Check the oven’s functionality with different kinds of food – solid, and liquid.**

**Check the oven’s functionality with different food at different temperatures.**

**Verify the oven’s functionality with different kinds of container material.**

**Verify that the power cord of the oven is long enough.**

**Verify that the usage instruction or user manuals have clear instructions.**

**• Write a scenario of Coffee vending Machine**

**UI scenario – Verify that the dimension of the coffee machine is as per the specification.**

**Verify that outer body, as well as inner part’s material, is as per the specification.**

**Verify that the machine’s body color as well brand is correctly visible and as per specification.  
Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc.**

**Verify that the quantity of hot water, milk, coffee powder perserving is correct.**

**Verify the power/voltage requirements of the machine.**

**Verify the effect of suddenly switching off the machine or cutting the power. The machine should stop in that situation and in power resumption, the remaining coffee should not get come out of the nozzle.**

**Verify that coffee should not leak when not in operation.**

**Verify the amount of coffee served in single-serving is as per specification.**

**Verify that the digital display displays correct information.**

**Check if the machine can be switched on and off using the power buttons.**

**Check for the indicator lights when the machine is switched on-off.**

**Verify that the functioning of all the buttons work properly when pressed.**

**Verify that each button has an image/text with it, indicating the task it performs.**

**Verify that complete quantity of coffee should get poured in a single operation, no residual coffee should be present in the nozzle.**

**Verify the mechanism to clean the system work correctly- foamer.**

**Verify that the coffee served has the same and correct temperature each time it is served by the machine.**

**Verify that system should display an error when it runs out of ingredients.**

**Verify that pressing the coffee button multiple times leads to multiple serving of coffee.**

**Verify that there is the passage for residual/extra coffee in the machine.**

**Verify that machine should work correctly in different climatic, moistures and temperature conditions.**

**Verify that machine should not make too much sound when in operation.**

**• Write a scenario of chair**

**Verify that the chair is stable enough to take an average human load.**

**Check the material used in making the chair-wood, plastic etc.**

**Check if the chair’s leg are level to the floor.**

**Check the usability of the chair as an office chair, normal household chair.**

**Check if there is back support in the chair.**

**Check if there is support for hands in the chair.**

**Verify the paint’s type and color.**

**Verify if the chair’s material is brittle or not.**

**Check if cushion is provided with chair or not.**

**Check the condition when washed with water or effect of water on chair.**

**Verify that the dimension of chair is as per the specifications.**

**Verify that the weight of the chair is as per the specifications.**

**Check the height of the chair’s seat from floor.**

**• To Create Scenario (Positive & Negative) Gmail (Receiving Mail)**

**Positive Scenarios:**

* **Login to Gmail and then click the Logout button. This should disable access to account, unless another successful Login is done.**
* **Login to Gmail, click the Logout button. The Login page should be displayed.**
* **If there are shared accounts. Logout from any one of the account. This should log you out from all the accounts.**
* **Login to Gmail. Logout. Then Login again using the right credentials.**

**Negative Scenarios:**

* **Open the same Gmail Account in two Different Browsers. Logging out from one of the browser, should not log you out from the other browser / you should be able to continue the session from the other browser.**
* **Enter the right credentials in Login. Click Login. While the page loads, Close the tab. This should not Logout, instead you should be already logged in.**
* **Login to Gmail. Disconnect Wi-fi and then try to Logout. It should not log out and when again the internet is available. The session should be logged in.**

**Online shopping to buy product (flipkart)**

1. **Launch Flipkart website/app**
   * **Expected: Homepage should load successfully.**
2. **Login with valid credentials**
   * **Expected: User should be redirected to their dashboard/homepage.**
3. **Search for a product (e.g., "Adidas Running Shoes")**
   * **Expected: Search results related to the keyword should be displayed.**
4. **Apply filters (brand, price, rating, etc.)**
   * **Expected: Only filtered results should appear.**
5. **Click on a product from the results**
   * **Expected: Product detail page (PDP) should load with correct info: image, price, description, reviews, etc.**
6. **Check product availability (Pincode)**
   * **Expected: System should display if the product can be delivered to the entered location.**
7. **Add product to cart**
   * **Expected: Item should be added successfully, and cart icon should update.**
8. **Proceed to checkout**
   * **Expected: User is taken to the checkout page with product and price summary.**
9. **Add/select delivery address**
   * **Expected: Address is selected/added successfully.**
10. **Choose payment method (e.g., UPI, COD, Credit Card)**
    * **Expected: Payment options are displayed correctly.**
11. **Complete payment**
    * **Expected: Payment is processed and confirmation page is shown with order ID and delivery date.**
12. **Verify email/SMS confirmation**
    * **Expected: Order confirmation email/SMS is sent with product details.**

**Negative Scenarios to Consider:**

* **Product out of stock**
* **Invalid pincode**
* **Payment failure**
* **No internet / slow connection**
* **Session timeout during payment**

**• Write a Scenario of Wrist Watch**

**Verify the type of watch – analog or digital.**

**In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch.**

**In the case of a digital watch, check the digital display for hours, minutes, and seconds is correctly displayed.**

**Verify the material of the watch and its strap.**

**Check if the shape of the dial is as per specification.**

**Verify the dimension of the watch is as per the specification.**

**Verify the weight of the watch.**

**Check if the watch is waterproof or not.**

**Verify that the numbers in the dial are clearly visible or not.**

**Check if the watch is having a date and day display or not.**

**Verify the color of the text displayed in the watch – time, day, date, and other information.**

**Verify that clock’s time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.**

**Check if the second hand of the watch makes ticking sound or not.**

**Verify if the brand of the watch and check if its visible in the dial.**

**Check if the clock is having stopwatch, timers, and alarm functionality or not.**

**In the case of a digital watch, verify the format of the watch 12 hours or 24 hours.**

**Verify if the watch comes with any guarantee or warranty.**

**Verify if the dial has glass covering or plastic, check if the material is breakable or not.**

**Verify if the dial’s glass/plastic is resistant to minor scratches or not.**

**Check the battery requirement of the watch.**

**• Write a Scenario of Lift(Elevator)**

**Verify the dimensions of the lift.**

**Verify the type of door of the lift is as per the specification.**

**Verify the type of metal used in the lift interior and exterior.**

**Verify the capacity of the lift in terms of the total weight.**

**Verify the buttons in the lift to close and open the door and numbers as per the number of floors.**

**Verify that the lift moves to the particular floor as the button of the floor is clicked.**

**Verify that the lift stops when the up/down buttons on a particular floor are pressed.**

**Verify if there is an emergency button to contact officials in case of any mishap.**

**Verify the performance of the floor – the time taken to go to a floor.**

**Verify that in case of power failure, the lift doesn’t free-fall and gets halted on the particular floor.**

**Verify lifts working in case the button to open the door is pressed before reaching the destination floor.**

**Verify that in case the door is about to close and an object is placed between the doors if the doors sense the object and again open or not.**

**Verify the time duration for which the door remains open by default.**

**Verify if the lift interior is having proper air ventilation.**

**Verify lighting in the lift.**

**Verify that at no point the lift door should open while in motion.**

**Verify that in case of power loss, there should be a backup mechanism to safely get into a floor or a backup power supply.**

**Verify that in case the multiple floor number button is clicked, the lift should stop on each floor.**

**Verify that in case of capacity limit is reached users are prompted with a warning alert- audio/visual.**

**Verify that inside lift users are prompted with the current floor and direction information the lift is moving towards- audio/visual prompt.**

**• Write a Scenario of whatsapp Group (generate group)**

1. **Open WhatsApp**
   * **Expected Result: WhatsApp opens successfully to the chat list screen.**
2. **Click on "New Chat" icon**
   * **Expected Result: Options like “New Group”, “New Contact”, etc. appear.**
3. **Select “New Group”**
   * **Expected Result: User is navigated to the contact selection screen.**
4. **Select contacts to add to the group**
   * **Expected Result: Contacts are selected and displayed at the top of the screen.**
5. **Tap on the arrow/next button**
   * **Expected Result: User is navigated to the “Group Info” screen.**
6. **Enter group name (e.g., “Project Team”)**
   * **Expected Result: Group name input is accepted (max 25 characters).**
7. **Optionally: Set group icon from camera, gallery, or emoji**
   * **Expected Result: Image/icon is successfully set or changed.**
8. **Tap on “Create” button**
   * **Expected Result: Group is created, visible in chat list with selected members added.**

**Negative Test Cases / Edge Scenarios:**

* **Try to create group without adding members → should show an error.**
* **Enter group name with only spaces or invalid characters.**
* **Add more than the allowed number of participants (WhatsApp limit).**
* **Network failure during group creation.**
* **App crashes or freezes during group creation.**

**Write a Scenario of Whatsapp payment**

1. **Launch WhatsApp**
   * **Expected Result: WhatsApp home screen loads successfully.**
2. **Navigate to the chat of the recipient (who also has WhatsApp Payments enabled)**
   * **Expected Result: Chat window opens.**
3. **Click on the attach/₹ icon → Choose "Payment"**
   * **Expected Result: UPI interface is displayed.**
4. **Enter payment amount (e.g., ₹500)**
   * **Expected Result: Amount field accepts value, and "Next" button is enabled.**
5. **Enter note/message (optional)**
   * **Expected Result: Note is optional, not required to proceed.**
6. **Click “Next”**
   * **Expected Result: UPI PIN screen is shown with bank details and amount confirmation.**
7. **Enter valid UPI PIN**
   * **Expected Result: Payment is processed successfully.**
8. **Verify confirmation message**
   * **Expected Result: Success message appears in chat: “You paid ₹500 to [Contact Name]”**
9. **Check transaction history**
   * **Expected Result: Transaction appears under “Payments” in the user profile or settings.**

**🔹 Negative/Edge Scenarios:**

* **Enter incorrect UPI PIN → show appropriate error message.**
* **Insufficient balance → transaction fails with message.**
* **No internet connection → payment fails gracefully with retry option.**
* **Payment to user not registered with WhatsApp Pay → disallow or prompt to invite.**
* **Send amount exceeding UPI limit (e.g., ₹1,00,000) → display limit error.**
* **Try sending payment from unverified number → block or redirect to verification.**

**When to used Usablity Testing?**

**Usability testing should be conducted early and often throughout the product development lifecycle, starting with prototyping and continuing post-launch to ensure ongoing improvements based on user feedback and behavior**.

**• What is the procedure for GUI Testing?**

**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.**

**1. Requirement Analysis**

* **Understand the application’s functionality and UI requirements.**
* **Identify key GUI components like buttons, text fields, menus, icons, etc.**

**2. Test Planning**

* **Define the scope of GUI testing.**
* **Determine the types of tests (functional, usability, compatibility, etc.).**
* **Prepare a test strategy and schedule.**

**3. Test Case Design**

* **Create detailed test cases covering:**
  + **Visual elements: Colors, fonts, icons, images.**
  + **Layout: Alignment, spacing, positioning.**
  + **Functional elements: Buttons, links, drop-downs, forms.**
  + **User interactions: Hover effects, clicks, drag-and-drop.**

**4. Test Environment Setup**

* **Prepare the testing environment with the required hardware, software, and network configurations.**
* **Install the application on different devices/browsers (for cross-platform testing).**

**5. Test Execution**

* **Execute the test cases manually or using automated tools like Selenium, QTP, etc.**
* **Perform functional testing to verify UI elements work correctly.**
* **Conduct usability testing to ensure the UI is user-friendly.**

**6. Defect Reporting and Tracking**

* **Log defects with detailed descriptions, screenshots, and steps to reproduce.**
* **Prioritize and track defects until they are resolved.**

**7. Regression Testing**

* **Re-test the application after fixes to ensure that the issues are resolved and no new issues have been introduced.**

**8. Final Reporting**

* **Compile test results, defect reports, and overall test coverage.**
* **Provide a final evaluation of the GUI’s quality.**

**9. Test Closure**

* **Review and document lessons learned.**
* **Archive test cases, scripts, and reports for future reference.**

**• What isthe difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| **Process followed by the development team within the software organization to develop a software product.** | **Process of carrying out various activities to ensure the quality of the software.** |
| **Stand for Software Development Life Cycle.** | **Stand for Software Testing Life Cycle.** |
| **Covers the entire life cycle of the software,** | **Limited only to the testing phase.** |
| **Working software product** | **Test reports, bug reports, and quality assurance** |
| **Developers, designers, testers, project managers** | **Primarily testers, QA engineers** |
| **Development team start developing the software in the development phase.** | **Testing team writes test scenarios to validate the quality of the software.** |
| **Technical team provides support to update and maintain the software** | **Test cases and automation scripts are maintained for updates.** |
| **Deliver functional, high-quality software** | **Ensure the software is defect-free** |

**• What is the difference between test scenarios, test cases, and test script?**

|  |  |  |
| --- | --- | --- |
| **Test scenarios** | **Test cases** | **Test script** |
| **A high-level description of what needs to be tested** | **A detailed step-by-step guide on how to test a specific scenario** | **A set of automated test steps written in a programming language** |
| **High-level** | **Detailed** | **Highly detailed (includes coding)** |
| **Covers broad functionality to be tested** | **Provides exact steps to execute and expected outcomes** | **Automates execution of test cases** |
| **Written in simple text format** | **Typically in tabular format with steps, inputs, and expected results** | **Written as code in a scripting language (e.g., Selenium, Python, Java, etc.)** |
| **"Test login functionality with valid and invalid credentials"** | Steps: **1. Open login page 2. Enter username/password 3. Click login 4. Verify success/error message** | **Selenium script that automates login testing in a web browser** |

**• 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.**

**Integrating and coordinating the testing activities into the software life cycle activities:**

**acquisition, supply, development, operation and maintenance**

**Assigning resources for the different activities defined**

**Defining the amount, level of detail, structure and templates for the test documentation.**

**Test Objectives:**

**What are you trying to achieve through testing? What specific functionalities or features need to be verified?**

**Test Scope:**

**What will be tested and what will not be tested? What are the boundaries of the testing effort?**

**Test Environment:**

**What hardware, software, and network configurations will be used for testing?**

**Test Resources:**

**What resources (e.g., testers, tools, data) are required for testing?**

**Test Schedule:**

**When will testing activities take place? What are the deadlines for different testing phases?**

**Test Cases:**

**What specific test cases will be executed to verify the software or hardware?**

**Test Procedures:**

**How will the test cases be executed? What are the steps involved in each test case?**

**Entry and Exit Criteria:**

**What conditions must be met before testing can begin and what conditions must be met before testing can be considered complete?**

**Risk Analysis:**

**What are the potential risks associated with the testing process? What are the contingency plans in case of unforeseen issues?**

**Defect Management:**

**How will defects be identified, reported, and tracked? What are the procedures for resolving defects?**

**Test Reporting:**

**How will the results of the testing activities be documented and reported? What metrics will be used to measure the effectiveness of the testing process?**

**Test Data:**

**What data will be used for testing? How will the data be prepared and managed?**

**Test Tools:**

**What tools will be used for testing? How will the tools be configured and used?**

**• What is priority?**

**priority refers to the urgency or importance of fixing a bug or defect, determining the order in which issues should be addressed and resolved.**

**• What is severity?**

**In software testing, severity refers to the degree of impact a bug or defect has on the functionality or user experience of the application. It's a measure of how critical the bug is and how much it affects the system's ability to function as expected.**

**• Bug categories are…**

**Bug Category: Security, Database, Functionality (Critical/General), UI**

**• Advantage of Bugzila .**

**Effective Bug Tracking:**

**Bugzilla is designed specifically for tracking bugs, issues, and feature requests, allowing teams to manage and prioritize them efficiently.**

**Robust Reporting:**

**The tool provides comprehensive reporting capabilities, enabling users to generate various reports, charts, and graphs related to bug statistics and project progress.**

**Streamlined Communication:**

**Bugzilla facilitates clear and organized communication between developers, testers, and other stakeholders by providing a central location for reporting and discussing issues.**

**Lightweight and Fast:**

**Bugzilla is known for its lightweight implementation and speed, minimizing database calls and generating efficient HTML.**

**Open Source:**

**As an open-source tool, Bugzilla is free to use and distribute, making it a cost-effective solution for many organizations.**

**Customizable:**

**Bugzilla offers a high degree of customization, allowing users to tailor the system to their specific needs and workflows.**

**Advanced Search:**

**Bugzilla provides advanced search options, allowing users to find specific bug details and track their progress effectively.**

**User Roles and Permissions:**

**Bugzilla allows for the definition of user roles and permissions, enabling fine-grained control over who can edit or see which bugs.**

**Automation:**

**Bugzilla includes automation features that streamline bug tracking processes, such as automated email notifications and workflows.**

**Integration:**

**Bugzilla can be integrated with other tools and systems, such as version control systems and project management software.**

**Used by Large Open Source Projects:**

**Bugzilla is used by many large open-source projects, demonstrating its reliability and scalability.**

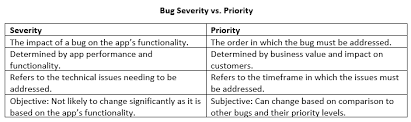
**Improved Product Quality:**

**By effectively tracking and resolving bugs, Bugzilla helps teams improve the overall quality of their products.**

**Increased Productivity:**

**By streamlining bug tracking and communication, Bugzilla helps teams increase their productivity and efficiency.**

**• Difference between priority and severity**



**HLR and Testcases:**

**https://docs.google.com/spreadsheets/d/1LfJdiqcRMYGkaBCNaPLDAZF7s\_5XIZCDFbkzyVDzkhE/edit?gid=2030213084#gid=2030213084**

**• What are the different Methodologies in Agile Development Model?**

**• Explain the difference between Authorization and Authentication in Web testing.**

**What are the common problems faced in Web testing?**