**MODULE-2 (MANUAL TESTING)**

1. What is exploratory testing?

* Exploratory testing is a software testing approach where testers explore the software, learn its functionalities, and simultaneously design and execute test cases. It is highly effective approach to testing software that gives testers the freedom to adapt experiment on the fly based on their observations of the system and user behaviors.

1. What is traceability matrix?

* A traceability matric is a document that maps and traces user requirements with test cases.

1. What is boundary value testing?

* The process of testing between extreme ends or boundaries between partitions of the input values.
* It is used to test boundary values because the input values near the boundary have higher chances of error. So, the tester focuses on, while entering boundary value whether the software is producing correct output or not.

1. What is equivalence partitioning testing?

* Equivalence partitioning is a software testing technique that divides the input data of a software unit partitions of equivalent data from which test cases can be derived. Test cases are design to cover each partition at least once.
* The advantage of this approach is reduction in time required for testing software due to lesser number of test cases.

1. What is integration testing?

* Integration testing is the second level of software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group.
* The focus of integration testing level is to expose defects at the time of interaction between integrated components or units.

1. What determines the level of risk?

* The risk level is determined by two dimensions which are probability and impact.
* Risks should be prioritized according to their level, which is obtained by assessing the likelihood of the event occurring and the impact of that event.

1. What is alpha testing?

* Alpha testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. It is one of the user acceptance testing.
* Alpha testing is always performed in virtual environment and within the organization.
* It comes under the category of both black box and white box testing.
* It is definitely performed and carried out at the developing organization location with the involvement of developers.

1. What is beta testing?

* Beta testing is the process of testing a software product or service in a real-world environment before its official release. It is always performed by the customers at their own site.
* It is an essential step in the software development life cycle as it helps identify bugs and errors that may have been missed during the development process.
* Beta testing is one of the types of user acceptance testing.
* Beta testing commonly uses black box testing.
* Beta testing does not require a lab or testing environment.

1. What is component testing?

* A software application is a combination of multiple components or modules. Testing the individual component of any application is called component testing.
* Component testing is the way to identify the issues at the module level, so defects are captured at the early stage for each module and hence reduces the efforts of both developers and testers while testing the entire application.
* This testing eliminates the errors at the initial stage and builds confidence at the component level.
* Component testing is performed before integration testing and after unit testing.
* It is a black box testing method so internal structure of code is unknown for the tester.

1. What is functional system testing?

* Functional testing is a type of software testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application.
* Functional testing verifies the operations and actions of an application. It is based on the requirements of the customer.
* It is carried out using the functional specification.
* Manual or automation tools can be used for it. It is easy to do manual testing.
* Example of functional testing are unit testing, smoke testing, sanity testing, integration testing, white box testing, black box testing, regression testing etc.

1. What is non-functional testing?

* Nonfunctional testing is a type of software testing that is performed to verify the non-functional requirements of the application. It checks the performance, reliability, scalability and other non-functional aspects of the software system.
* It verifies the behavior of the system is as per requirements or not. It tests all the aspects that are not tested in functional testing.
* It is tough to do manual testing.
* Example of non-functional testing are load testing, stress testing, security testing, compatibility testing, portability testing, scalability testing, reliability testing etc.

1. What is GUI testing?

* Graphical user interface testing is the process for ensuring proper functionality of the graphical user interface for a specific application. GUI testing evaluates a design of elements such as layout, colors and fonts, font sizes, labels, text boxes, text formatting, images, buttons, links, icons and content.
* GUI testing processes may be either manual or automatic and are often performed by third party companies, rather than developers or end users.
* It allows to test the functionality from a user’s perspective. It is a technique that tests the part of application visible to the user.

1. What is ad hoc testing?

* Ad hoc testing is a type of software testing that is performed informally and randomly with an aim to break the system. It is not performed in a structured way so it is not based on any methodological approach.
* Ad hoc testing does not follow any documentation, test cases and test design.
* It is done after formal testing. It takes less time than other testing techniques.
* This testing can be performed at any time during software development life cycle process.

1. What is load testing?

* Load testing determines the behavior of the application when multiple users use it at the same time. It is the response of the system measured under varying load conditions.
* The goal of load testing is to identify bottlenecks and determine the maximum number of users or transactions the system can handle.
* It helps to ensure that system can handle the expected usage levels and identify any issues before the system is deployed to production.

1. What is stress testing?

* Stress testing is a technique that determines the robustness of software by testing beyond the limits of normal operation.
* Stress testing is to test the system behavior under extreme conditions and is carried out till the system failure.

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

* White box testing is a software testing technique that involves testing the internal structure and workings of a software application. The tester has a knowledge about source code so that tester can verify the correctness of the software at the code level.
* It is also known as glass box testing.
* **There are 3 types of white box testing**: -
* Statement coverage
* Branch coverage
* Condition coverage

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

* This method of test can be applied virtually to every level of software testing.
* Black box testing is a method of software testing that examines the functionality of an application without having knowledge about internal structure or workings.
* **There are 4 black box testing techniques**: -
* **Equivalence partitioning** – Divides the input data of a software unit into partitions of equivalent data from which test cases can be derived.
* **Boundary value analysis** – The process of testing between extreme ends or boundaries between partitions of the input values.
* **Decision tables** – A decision table is good way to deal with combinations of inputs.
* **State transition technique** – A black box test design technique in which test cases are designed to execute valid and invalid state transitions.

1. Mention what are the categories of defects?

* Functional defects
* Interface defects
* Performance defects
* Compatibility defects
* Security defects
* Usability defects
* Data defects
* Documentation defects
* Regression defects
* Critical defects
* Minor defects
* Integration defects
* Logical defects

1. Mention what big bang testing is?

* Big bang integration testing is a testing approach where all components or modules are integrated and tested as a single unit. This is done after all modules have been completed and before any system level testing is performed.

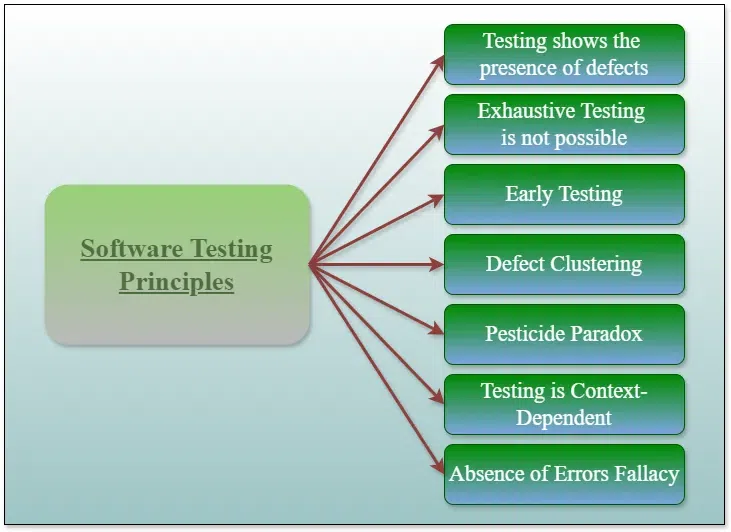
1. What is the purpose of exit criteria?

* The purpose of exit criteria is to determine if a test plan or project can exit to the next stage or be considered complete.

1. When should "Regression Testing" be performed?

* Regression testing is performed after a software change or update has been made. Its purpose is to ensure that the recent modifications have not affected the existing functionality of the software.
* So, in regression testing we need to make sure that product continues to perform properly in production environment before we release it.

1. What is 7 key principles? Explain in detail?



* **Testing shows the presence of defects: -** The goal of software testing is to make the software fail. It reduces the presence of defects. Software testing can ensure that defects are present but it cannot prove that software is defect free. Multiple tests can never ensure that software is 100% bug free. Testing can reduce the number of defects but not remove all defects.
* **Exhaustive testing is impossible: -** It is the process of testing the functionality of the software in all possible inputs and pre conditions is known as exhaustive testing. This testing is impossible because software can test only some test cases and assume that the software is correct and it will produce the correct output in every test case. It will take more cost and effort to check every test case so software can never test at every test case.
* **Early testing: -** The defect detected in the early phases of SDLC will be less expensive. So, early testing will start at the initial phase that is requirement analysis phase for better performance of software.
* **Defect clustering:** - A small number of modules can contain most of the defects. So recheck only that area which have defects.
* **Pesticide paradox: -** Repeating the same test cases again and again will not find new bugs. So, it is necessary to review test cases and check with other test case to find new bugs.
* **Testing is context dependent: -** The testing approach depends on the context of the software developed. Different types of software need to perform in different types of testing.
* **Absence of errors fallacy: -** It is not necessary that software is 100% bug free but it is mandatory to fulfill all the customer requirements.

1. 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 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 | QC can be considered as the subset of quality assurance | Testing is the subset of quality control |

1. Difference between Smoke and Sanity?

|  |  |
| --- | --- |
| **Smoke testing** | **Sanity testing** |
| It is performed to ascertain that the critical functionalities of the program are working fine | It is done to check the new functionality /bugs have been fixed |
| The objective of this testing is to verify the stability of the system in order to proceed with more rigorous testing | The objective of the testing is to verify the rationality of the system in order 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 |
| It is a subset of acceptance testing | It is a subset of regression testing |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |

1. Difference between verification and Validation

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| The process of evaluating work products of a development phase to determine whether they meet the specified requirements for the phase | The process of evaluating software during or the end of the development process to determine whether is satisfies specified business requirements |
| It includes checking documents, design, codes and programs | It includes testing and validating the actual product |
| Verification is the static testing | Validation is the dynamic testing |
| It does not include the execution of the code | It includes the execution of the code |
| Methods used in verification are reviews, walkthroughs, inspections and desk checking | Methods used in validation are black box, white box testing and non-functional testing |
| It can find the bugs in the early stage of the development | It can only find the bugs that could not be found by the verification process |
| Quality assurance team does verification | Validation is executed on software code with the help of testing team |
| It consists of checking of documents/files and is performed by human | It consists of execution of program and is performed by computer |
| The goal of verification is application and software architecture and specification | The goal of validation is an actual product |
| Verification is for prevention of errors | Validation is for detection of errors |

1. Explain types of Performance testing.

* Load testing – It checks the product’s ability to perform under anticipated user loads. The objective is to identify performance congestion before software product is launched in the market.
* Stress testing – It involves testing a product under extreme workloads to whether it handles high traffic or not. The objective is to identify the breaking point of the software product.
* Endurance testing – It is performed to ensure the software can handle the expected load over a long period.
* Scalability testing – The software application’s effectiveness is determined by scaling up to support an increase in user load. It helps in planning capacity additions to software system.

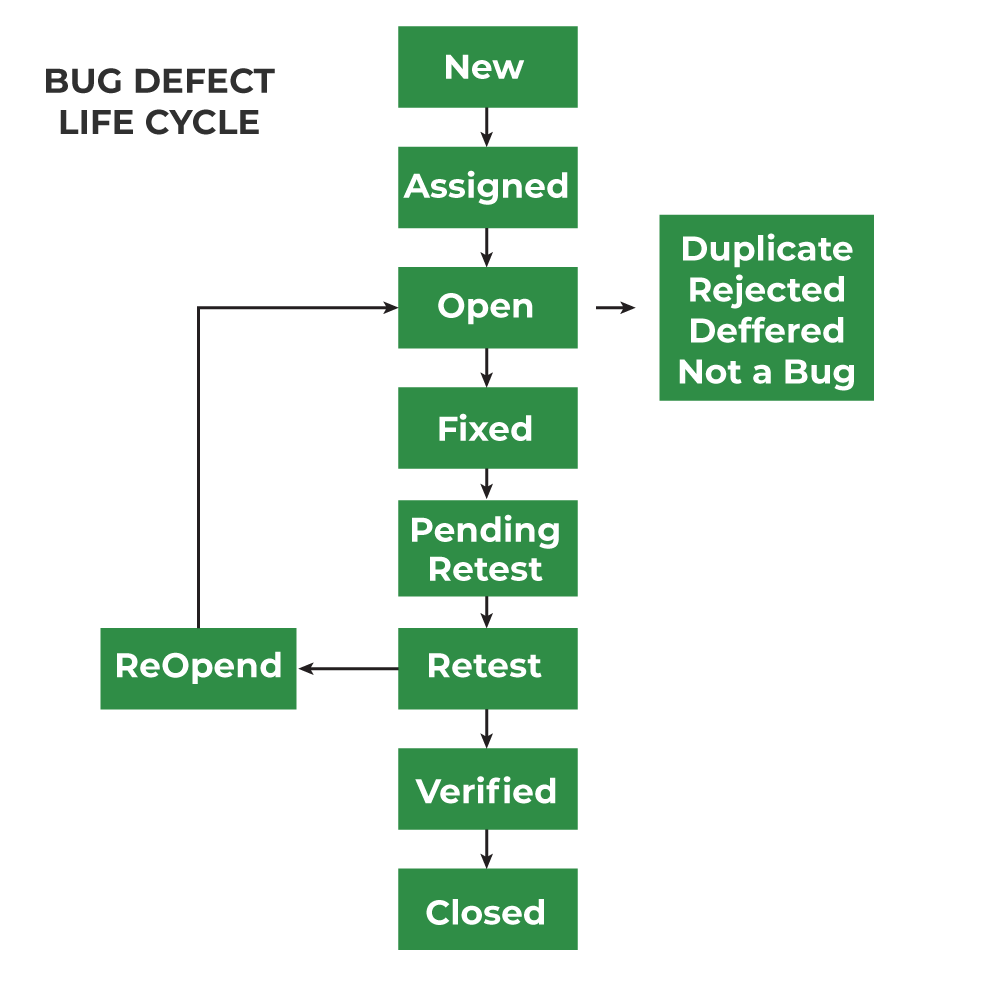
1. What is Error, Defect, Bug and failure?

* Error – A mistake in coding is called error.
* Defect – Error found by tester is called defect. The variation between the actual results and expected results is known as defect.
* Bug – Defect accepted by development team then it is called bug.
* Failure – When a defect reaches the end customer is called failure.

1. Difference between Priority and Severity

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority is a parameter to decide the order in which defects should be fixed | Severity is a parameter to donate the impact of a particular defect on the software |
| Priority means how fast the defects has to be fixed | Severity means how severe the defect is affecting the functionality |
| Priority is related to scheduling to resolve the problems | Severity is related to the quality standard |
| The product manager decides the priorities of defects | The testing engineer decides the severity level of the defect |
| It is based on customer’s requirements | It is based on the technical aspect of the product |

1. What is Bug Life Cycle?



* **New** – When a new defect is logged and posted for the first time it is assigned a status as “new”.
* **Assigned** – Once the bug is posted by the tester, the lead of tester approves the bug and assigns the bug to the developer team.
* **Open** – The developer starts analyzing and works on the defect fix.
* **Fixed** – When a developer makes a necessary code change and verifies the change, he or she can make bug status as “fixed”.
* **Pending retest** – Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the tester end that status assigned as “pending retest”.
* **Retest** – Tester does the retesting of the code at this stage to check whether the defect is fixed by developer or not and changes the status to “retest”.
* **Verified** – The tester retests a bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned as “verified”.
* **Reopen** – If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again, the bug goes through the life cycle.
* **Closed** – If the bug is no longer exists then the tester assigns the status as “closed”.
* Duplicate – If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate”.
* **Rejected** – If the developer feels the defect is not a genuine defect, then it changes the defect to “rejected”.
* **Deferred** – If the present bug is not a prime priority and if it is expected to get fixed in the next release, then status is assigned as “deferred” to such bugs.
* **Not a bug** – If it does not affect the functionality of the application then the status assigned to bug is “not a bug”.

1. Explain the difference between Functional testing and Non-Functional testing.

|  |  |
| --- | --- |
| **Functional testing** | **Non-functional testing** |
| It focuses on testing functionality of the software system | It focuses on testing the non-functional aspects of the software system |
| It is based on requirements of customer | It is based on expectations of customer |
| Functional testing is easy to execute manually | It is hard to execute non-functional testing manually |
| It tests what the product does | It describes how the product does |
| Functional testing is based on the business requirement | Non-functional testing is based on the performance requirement |
| Examples of functional testing are unit testing, smoke testing, integration testing, regression testing, white box testing, black box testing etc. | Examples of non-functional testing are performance testing, load testing, stress testing, security testing, scalability testing etc. |

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

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| SDLC stands for software development life cycle | STLC stands for software testing life cycle |
| SDLC is a process that focuses on developing software that can meet the client’s expectations and work efficiently in the technological infrastructure | STLC specifies what type of test activities must be carried out and when should the testing team accomplish those test activities |
| It focuses on both development and testing process | It focuses only on testing process |
| Phases of SDLC are requirement gathering, analysis, design, coding, testing, deployment and maintenance | Phases of STLC are requirement analysis, test planning, test design, environment setup, test execution, test closure |
| SDLC is taken as the predecessor relationship with other life cycle | STLC is taken as the successor relationship with other life cycle |
| Project managers, business analysts, designers and developers are involved in SDLC | Quality assurance and tester teams are involved in STLC |
| The goal is to deliver a reliable and completely functional software product | The goal is to check and confirm the software meets the particular requirements and functions appropriately |
| The outcome of SDLC is delivering a higher quality product to customer | The outcome of STLC is to deliver bug free software |

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

|  |  |  |
| --- | --- | --- |
| **Test scenarios** | **Test cases** | **Test script** |
| It is any functionality that can be tested | It is a set of actions executed to verify particular features or functionality | It is a set of instructions to test an app automatically |
| It is derived from test artifacts like business requirement specification and software requirement specification | It is mostly derived from test scenarios | It is mostly derived from test cases |
| It helps test the end-to-end functionality in an agile way | It helps in exhaustive testing of an app | It helps to test specific things repeatedly |
| It is more focused on what to test | It is focused on what to test and how to test | It is focused on the expected result |
| It takes less time and few resources to create | It requires more resources and time | It requires less time for testing but more resources for scripts creating and updating |
| It includes an end-to-end functionality to be tested | It includes test steps, data, expected results for testing | It includes different commands to develop a script |
| It allows quickly assessing the testing scope | It allows detecting errors and defects | It allows carrying out an automatic execution of test cases |
| The main task is to check the full functionality of the 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 |

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

* A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources, test estimation and test deliverables
* Test plan is a base of every software’s testing. It is the most crucial activity which ensures availability of all lists of planned activities in an appropriate sequence
* **There are three types of test plan:**
* Master test plan
* Phase test plan
* Specific test plans
* **Information that should be covered in a test plan:**
* Introduction – This section provides an overview of the purpose, scope, and objectives of the test plan
* Scope – It defines the boundaries of the testing effort, including what will and will not be tested
* Objectives – This section outlines the goals and desired outcomes of the testing process, such as identifying defects, validating functionality and ensuring compliance with requirements
* Approach – It describes the testing methodologies, techniques and strategies that will be employed, such as manual testing, automated testing, or combination of both
* Roles and responsibilities – It specifies the responsibilities of individuals or team involved in the testing process, including testers, developers, project managers, and stakeholders
* Test environment – This section details the hardware, software, tools and other resources required to conduct testing effectively
* Test deliverables – It lists the documents, reports, and artifacts that will be produced as part of the testing process, such as test cases, test scripts and defect reports
* Test schedule – This outlines the timeline and milestones for the testing effort, including start and end dates, testing phases and checkpoints
* Risks and assumptions – It identifies potential risks, challenges and assumptions that may impact the testing process
* Dependencies – This section highlights any dependencies on external factors, such as third-party systems or resources, that may affect the testing schedule or outcomes
* Exit criteria – It specifies the conditions that must be met for testing to be considered complete and outlines the criteria for accepting or rejecting the software
* Approvals – This section documents the stakeholders or individuals who must review and approve the test plan before testing can commence

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

* **Scrum** – It is one of the most popular agile frameworks. It divides the project into small, time boxed iterations called sprints. Each sprint involves a cross functional team working collaboratively to deliver a potentially shippable product increment.
* **Kanban** – Kanban is a visual agile methodology focused on workflow management. It utilizes a kanban board, which consists of columns representing different stages of work and cards representing individual tasks or user stories.
* **Extreme programming (XP)** – XP is an agile methodology that emphasizes engineering practices to ensure high quality software delivery. It promotes practices such as test-driven development, pair programming, continuous integration, collective code ownership and frequent releases. XP prioritizes customer feedback and encourages close collaboration between developers and customers throughout the development process.
* **Lean agile** – It combines principles from lean manufacturing with agile methodologies to eliminate waste, optimize efficiency and maximum value delivery. It focuses on delivering value to customers quickly, minimizing lead time and continuously improving processes.
* **Crystel** – Crystel methodologies prioritize communication, simplicity and reflective improvement. They support for frequent delivery, close collaboration and adapting practices to fit the unique needs of each project.
* **Dynamic systems development method** – DSDM is an agile framework that provides a structured approach to project management and delivery. It emphasizes user involvement, iterative development and incremental delivery within fixed timeframes and budgets. DSDM focuses on delivering high value features early and adjusting requirements based on feedback throughout the development lifecycle.
* **Feature driven development** – FDD is an agile methodology that emphasizes feature driven planning and design. It involves breaking down large projects into smaller, manageable features and prioritizing their implementation based on business value. FDD utilizes iterative and incremental development cycles, with a focus on delivering working features at regular intervals.

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

|  |  |
| --- | --- |
| **Authentication** | **Authorization** |
| It is the process of identifying a user to provide access to a system | It is the process of giving permission to access the resources |
| The user or client and server are verified | It is verified that if the user is allowed through the defined policies and rules |
| It is usually performed before the authorization | It is usually done once the user is successfully authenticated |
| It requires the login details of the user, such as username and password etc. | It requires the user’s privilege or security level |
| Data is provided through the token Ids | Data is provided through the access tokens |
| For example, entering login details is necessary for the employees to authenticate themselves to access the organizational emails or software | For example, after employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles |
| Authentication credentials can be partially changed by the user as per the requirement | Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner or manager of the system and he can only change it |

* **Common problems faced in web testing:**
* Cross browser compatibility
* Responsive design
* Performance
* Security
* System integration
* Usability
* App functionality
* Communication
* User experience

1. Write a scenario of only WhatsApp chat messages

* Verify that users can send text messages to individual contacts or groups
* Check that the sent message appears in the chat window with correct formatting and timestamp
* Verify the user can send multimedia messages like images, videos and audio files
* Verify the multimedia files are successfully sent and received by users
* Verify that emojis can be sent and received in messages
* Check that emojis are displayed correctly and are compatible across different devices and platforms
* Verify that the user can edit the sent messages
* Verify that the users can forward messages to other contacts or groups
* Check that forwarded messages retain their original formatting
* Verify that sender can delete the sent messages from both sender’s and receiver’s chat history
* Check that deleted messages leave an indication that a message was deleted
* Verify that users can search for specific message within a chat and user can search their contact for the chat
* Verify that users receive notifications for new messages
* Check the end-to-end encryption to ensure that messages are secure and cannot be intercepted by unauthorized parties
* Check that backup and restore functionality to ensure users can safely store and retrieve their chat history
* Verify that user can create new group chats, adding/removing participants and managing group settings
* Verify that user can reply messages to particular person in group chat

1. Write a Scenario of Pen

* Verify the type of pen, whether it is ballpoint pen, ink pen or gel pen
* Verify that user is able to write clearly over different types of papers
* Verify that the pen is with a cap or without a cap
* Verify that color of ink on the pen
* Verify that the text written by the pen should have consistent ink flow without leaving any blob
* Verify that pen’s ink should not leak in case it is tilted upside down
* Verify that the text written by the pen is erasable or not
* Check the functioning of the pen by applying normal pressure during writing
* Verify that the material used in outer body of the pen
* Verify that the pen is easily breakable or not
* Check if the text written by the pen is waterproof or not
* Verify that pen can support multiple refills or not
* Check the grip of the pen that user comfortably grip the pen while writing
* Verify that mechanism to refill the pen is easy to operate for ink pens
* Verify the size of the tip of ballpoint pen
* Verify that effect of oil and other liquids on the text written with a pen
* Verify the functioning of the pen when user tries to write on surfaces like glass, plastic, wood etc.
* Check the weight of the pen should be as per specifications

1. Write a Scenario of Pen Stand

* Check the surfaces of the pen stand like it is smooth or uneven
* Verify that it stands stable on the table or not
* Check that applying the slight pressure from different angles it breaks or not
* Verify the material used to make the pen stand is as per specifications or not
* Verify the color and design of the pen stand is as per specifications or not
* Check the size capacity of pen stand with filling variety of pens, pencils, markers and other stationary items
* Verify that the pen stand can hold all items securely without tipping over
* Verify that the height of the stand is not too big
* Check that it is easy to clean the pen stand when its dusty or stain
* Check the cleaning methods with the use of different clothes or water
* Check the pen stand quality on different environmental conditions such as humidity or extreme temperature
* Verify that packaging of pen stand is protect the product during transportation
* Verify that it has any additional features like compartments for paper clips or a slot for a phone or not
* Check that the pen stand for any visual defects such as scratches, dents or discoloration

1. Write a Scenario of Door

* Verify that the door is single or bi-folded door
* Verify that door opens in inward direction or outward direction
* Verify that the material used in door body is wooden or glass or anything
* Verify that the color of the door is as specified
* Verify that the dimension of the door is as per specification
* Verify that the quality and strength of the door
* Check the number of locks in the door is interior side or exterior side
* Verify that door is having peek-hole or not
* Verify that door is having stopper or not
* Verify that door closes automatically or not
* Verify that condition of door in different climate conditions like temperature, humidity etc.
* Verify the door makes the noise when its opened or closed
* Verify that pull and push functions or not
* Verify that amount of force required when open or close the door
* Verify that the door is sliding door or rotating door

1. Write a Scenario of ATM

* Verify that the type of ATM machine, if it has touch screen or keypad buttons or both
* Verify that inserting a valid card different banking options appear on the screen
* Verify that the touch of the ATM screen is smooth and operational
* Verify that the user can choose an option for language or not
* Verify that ATM machine have computer speaker for to do the next step
* Verify that there is limited number of attempts user is allowed to enter the pin code
* Check that user is asked to enter a pin number before displaying card or bank account details
* Verify 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 user can see options for cash withdraw or cash deposit after choosing the account type
* Verify that the user is allowed to do only one transaction per pin request
* Verify that there is limit for cash withdraw in one transaction
* Verify that the user’s session timeout is maintained
* Verify that if ATM machine runs out of money, a proper message is displayed to the user
* Verify that user is provided the option to get the transaction details in printed form
* 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 user is not allowed with the expired ATM card and error message gets displayed
* Verify that any kind of issue occurs in ATM machine, the transaction is marked as null and the amount is not withdrawn from user’s account

1. When to used Usability Testing?

* Usability testing can be conducted on the current iteration of a product before beginning any new design work, after you have begun the strategy work around a new application.
* There are four ways to use usability testing:
* Before you start designing
* Once you have a prototype
* Before launching the product
* At regular intervals after launch

1. What is the procedure for GUI Testing?

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

1. Write a scenario of Microwave Owen

* Verify that the dimension of the oven is as per the specification
* Verify that the oven’s material is optimal for its use as an oven and as per specification
* Verify that the oven heats the food at the desired temperature properly
* Verify that the oven heats the food within specified time duration
* Verify that ovens functioning with the maximum and 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 open and closed smoothly
* Verify that battery requirement of the microwave oven and check the functions at that power
* Verify that the text written over the oven’s body is clearly visible
* Verify that the digital display is clearly visible and functions correctly
* Verify that the temperature regulator smooth to operate and works correctly
* Check the oven’s functionality with different kinds of food like solid and liquid and at different temperature
* Check the maximum capacity of the oven and its functioning with that volume of food
* Verify that oven’s functionality with different kinds of container material

1. Write a scenario of Coffee vending Machine

* Verify that the dimension of the coffee machine is as per specification
* Verify that outer body as well as inner part’s material is as per specification
* Verify that the machine’s body color and brand is visible and as per specification
* Verify that input mechanism for coffee ingredients like milk, water, coffee beans etc.
* Verify that the quantity of hot water, milk, coffee powder per serving is correct
* Verify the voltage requirements of the machine
* 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
* Verify the indicator lights when the machine is switched on-off
* Verify that the functioning of all the buttons work properly when pressed
* Check that machine can be switched on and off using the power buttons
* Verify that each button has an image/text with it, indicating the task it performs
* Verify that mechanism to clean the system work correctly
* 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 machine should work correctly in different climatic, moistures and temperature conditions
* Verify that machine should not make noise when in operation
* Check the amount of time the machine takes to serve a single serving of coffee
* Check the functioning of the coffee machine when multiple buttons are pressed simultaneously
* Verify the complete quantity of coffee should get poured in a single operation, no residual coffee should be present in the nozzle

1. Write a scenario of chair

* Verify that the chair is strong enough to take an average human load
* Check that the material used in making the chair is wood, plastic etc.
* Check that chair’s legs are level to the floor
* Verify that the usability of the chair as an office chair or normal household chair
* Verify that there is back support in the chair
* Verify that there is a support for hands in the chair
* Verify the chair’s color and paint’s type
* Verify that the chair’s material is brittle or not
* Check if cushion is provided with chair or not
* Verify that the dimension of chair is as per specifications
* Check that condition when washed with water or effect of water on chair
* Check that the height of the chair’s seat from floor
* Verify that the weight of the chair is as per specifications
* Verify that the chair is adjustable like height adjustment, forward tilt adjustment etc.
* Verify that the edge of chair is needed to be in round shape or sharp

1. Write a Scenario of Wrist Watch

* Verify the type of watch is analog or digital
* If there is an analog watch, check the correct time displayed by the second, minute and hour hand of the watch
* If there is a digital watch, check the digital display for hours, minutes and seconds is correctly displayed
* Verify that the material of the watch and its strap
* Check the shape of the dial is as per specification
* Verify the dimension of the watch is as per specification
* Verify the weight of the watch
* Check the watch is waterproof or not
* Verify that the numbers in the dial are clearly visible or not
* Check the watch is having a date and day display or not
* Verify the color of the text displayed in the watch
* Verify the brand of the watch and check its visible in the dial
* Check the clock is having stopwatch, timers and alarm functionality or not
* If it’s a digital watch, check the format of the watch 12 hours or 24 hours
* Verify that the watch comes with any guarantee or warranty
* Verify the dial’s glass/plastic is resistant to minor scratches or not
* Verify the dial material like plastic or glass is breakable or not
* Check the battery requirement of the watch
* Verify that clock’s time can be corrected using the key in case of analog clock and buttons in case of digital clock
* Check the second hand of the watch makes ticking sound or not

1. Write a Scenario of Lift (Elevator)

* Verify the dimensions of the lift
* Verify the type of door of the lift is as per specification
* Verify the material used in the lift interior and exterior
* Verify the capacity of the lift in terms of the 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 there is an emergency button to contact officials in case of any mishap
* Verify the time taken to go to the floor
* Verify that in case of power failure, the lift doesn’t free fall and gets halted on the particular floor
* Verify if the lift interior is having a proper lights and air ventilation
* Verify that the multiple floor number button is clicked, the lift should stop on each floor
* Verify that in case of power loss, there should be a backup mechanism to safely get into a floor
* Verify that it gives a warning alert when users limit is reached above the capacity
* Verify that lift stops when the up/down buttons on a particular floor are pressed

1. Write a Scenario of WhatsApp Group (generate group)

* Verify that the “create group” option is easily accessible from the WhatsApp interface
* Verify that initiating the group creation process is easy and straightforward
* Check if the user initiating group creation has necessary permissions to do
* Verify that only authorized users can create new groups to maintain security and privacy
* Check if users can set a unique and appropriate name for the group
* Verify that users can select or upload an icon that represents the group’s theme or purpose
* Verify the functionality to add participants to the group from the user’s contacts
* Check the users can easily search and select multiple contacts to add to the group
* Check the various privacy setting available for the group, such as allowing only admins to edit group or restricting who can send messages
* Verify that settings can be configured during the group creation process
* Verify the invitation mechanism to invite users to join the group via WhatsApp invitation links or direct invites
* Check the maximum capacity of participants allowed in a group to ensure it meets WhatsApp’s specified limits
* Verify that users are notified if the group is at capacity and cannot add more members
* Check the various error scenarios, such as network issues, server errors or invalid inputs during the group creation process
* Verify that appropriate error messages are displayed to the user, to guide them to resolve the issue
* Check the performance of the group creation process under different network conditions
* Verify that the process is fast and responsive, even with the large number of contacts or groups already existing
* Check the consistency and compatibility when creating group across different devices and operating system

1. Write a Scenario of WhatsApp payment

* Verify the setup process to link a bank account or payment method with WhatsApp
* Verify that users can easily navigate through the setup screens and enter their banking details securely
* Verify that the verification process for linking the bank account is smooth and efficient
* Check the process of sending payments to contact within WhatsApp
* Verify that users can initiate payments from within chat window or directly from payments section
* Check that users can specify the amount, add optional notes, and select the payment method before sending
* Check the process of receiving payments from contacts
* Verify that users receive timely notifications when someone sends them a payment
* Check the transaction history feature to view past payments
* Verify that users can easily access their transaction history and filter transactions by date, contact or amount
* Verify that transaction details are displayed correctly
* Verify that transactions are encrypted and secure from unauthorized access
* Check authentication mechanisms such as PIN, fingerprint, or face ID for authorizing payments
* Check various error scenarios such as insufficient funds, invalid recipient or network issues during transactions
* Verify that users receive clear and informative error messages to resolve issue
* Verify that users can easily request refunds and the process is handled transparently
* Check that it has ability to make international payments and supports different currencies
* Verify that exchange rates are accurate and transparently displayed to users before confirming the transaction
* Check the performance of WhatsApp payments under various network conditions and user loads
* Verify that payments are processed quickly and reliably, even during peak usage times