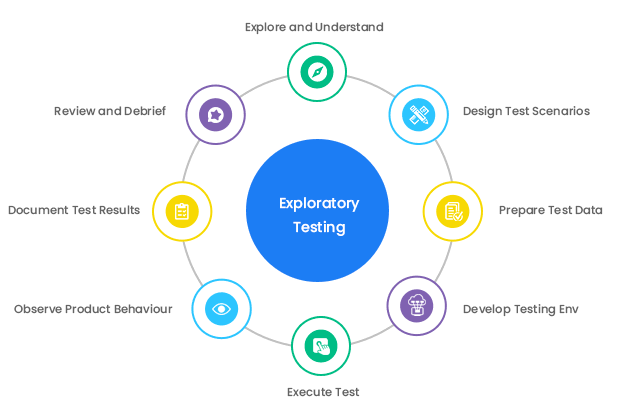
# 1). What is Exploratory Testing?

It’s a process that is concurrent where test design, execution, and logging happen simultaneously.

* By using experience, learning takes place to discover new solutions with new test patterns.
* Testing includes Scope and focus as a thinking activity.
* Discover expected problems.
* A brief description script prepared of how the test is performed.
* More structured than error guessing.
* Simulating real-world usage by experience.
* Mimic the end users' actions.

Mission

* Detailing about what to be tested, why and how to assess its quality.
* Keeping records is important of issues raised during tests.
* Pairing takes place for testing.

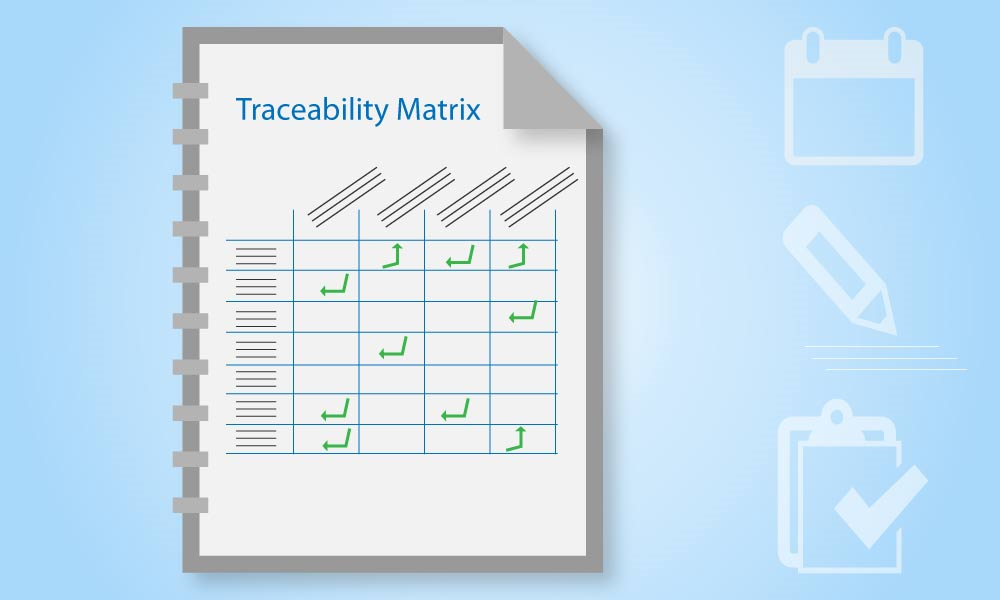
# 2). What is a traceability matrix?

A Traceability Matrix is a document that helps in recognizing how much requirement is finished and helps in tracing how much is remaining.

* The link between system requirements and test cases.
* It traces the requirements given by the client with the developed test cases.
* Traceability matrix prepared in rows and columns that help in identifying format in an easy way.

**Format and Structure.**

Traceability varies from company to company and organization to organization. But certain parameters help easily identify data be as **Functionality ID**, and **Test Case ID** in Excel sheet format.

* Requirement description that helps in clarifying what function is performed.
* The priority is Functionality is High level or Low-level requirement.
* Test Scenario ID which is unique to how the functionality works in pros and Cons.
* Test Case ID unique helps in identifying functionality tested in various ways.
* Development Status whether it is developed or not.
* The test Designer helps who made the test case.
* Test case Execution environments that show at what environment test cases got tested.
* Bug Status that is open/Closed/Retested.
* Bug ID that helps in tracking bugs in the tool tested with a link.
* Comments and review column.

***In a nutshell*** in short 100% test coverage helps to achieve quality bug-free, hassle-free for clients, and friendly for Users helps achieve the goal of the software.

# 3) What is Boundary Value Testing?

Software testing is a very resource-intensive activity for utilizing every aspect and resources test scenarios and test cases are designed. The testing technique is there called Boundary Value testing.

* Methodology for designing test cases that work on the limits of valid ranges.
* Refining with equivalence data. Highlighting error that is better than equivalence classes.
* Highlight error better than equivalence partitioning.
* Effective efforts at the extreme ends of the equivalence classes.
* For example: - We have to test between 1 to 50 input then we can put boundary value as input from starting 1 and 2 then at the end 49,50, so this is a more effective way to analyze with the help of boundary value process.

***In a nutshell***

Testing on the boundary values of valid and invalid partitioning.

# 4). What is Equivalence partitioning testing?

A technique that divides input data into groups that are expected to produce the same output as behavior. The aim is to do testing by equivalent parting.

Treat groups of inputs as equivalent and select one as a representative input to test that group.

* For example: - If we want to test data between 1 to 100/ (<=1) to (>=100) then the most effective way to do Equivalence partitioning is to divide groups and make it 5 groups.
* Representative groups are 1 to 20, 20 to 40, 40 to 60, 60 to 8,0, and the last one is 80 to 100.
* Then from this group select one as representative input data for testing. 15,34,45,66,98 then this process helps in resulting better coverage in less time.

***In a nutshell*** Test case designing technique that helps to reduce the number of test cases you need to execute. Helps while working with a large number of data this helps in coverage in an effective manner.

# 5). What is Integration testing?

Integration testing is associated with the architectural design phase. Test that is performed to check internal modules is integrated well while communication and their coexistence.

* Verify that modules are merging in one software properly and working.
* And they are responding as required the outcome is as expected.
* Testing helps to check that due to integration functionality is not affected.
* For enhancing quality there are different levels to approach for coverage and reliability in software.
* There are two types of test software that is

1. Static Testing: In static testing, software applications do not run only verification takes place with the help of review.
2. Dynamic Testing: In dynamic testing application runs and the code is checked and tested for the actual outcome.

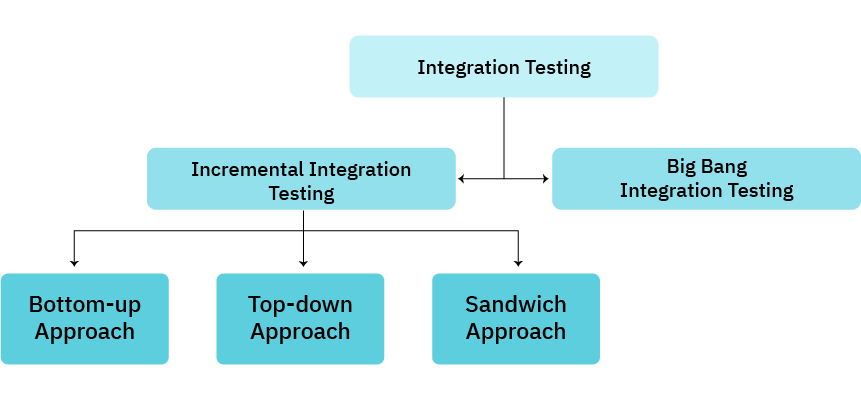
There are two levels of integration testing.

1). Component integration testing.

1. Functional testing: The black box technique is used against interface requirements for the component under test.
2. Non-functional testing: Testing done to check the reliability, and performance of the interface.

2). System integration testing.

**Integration testing takes place before system testing and after Unit testing. There are various types of integration testing.**

****

***In a nutshell***

**The important** level that is necessary to perform on the software to check efficiency, communication, and smoothness in between the modules that are integrated is called integration testing.

# 6). What determines the level of risk?

Risk is a factor that could result in negative consequences in the future and is marked as impact and likelihood.

* Testing finds defects that reduce the risk of project failure.
* Process improvement is helpful in determining risk factors.
* Quality is improved if risk is assured before it affects in project.

There are two types of risk: -

1. Project risk: - this is termed in likelihood and if occurs then mitigation requires resolving an issue with knowledge replacement.
2. Product risk: - This impact is high-level risk. This mitigation requires smoke or sanity testing.



***In a nutshell***

Risk assessment involves identification, analysis, and evaluation. Testing applied in allocating resources and reducing product risks.

# 7). What is Alpha testing?

# 8). What is beta testing?

# 9). What is component testing?

Component testing means a single module or single unit which it is called unit testing/ component testing/Module testing.

* The smallest testable part is taken for testing the first phase of test levels.
* Components can be tested by the developer in isolation also.
* With the help of stubs and drivers along with fake objects, mock and framework are also used to test components.
* Component unit testing is done by the development team to ensure that the code meets its requirements.
* The developer takes the help of a debugging tool to check the behavior of code in the software.
* Unit testing cannot be performed manually so with the help of an automation tool it is executed.
* The technique of white box testing is used by the developer.
* For testing skill required is internal programming code and architecture knowledge for testing.
* The developer puts inputs performs test scripts and asserts the actual outcome with the expected outcome.

***In a nutshell***Component testing is done by the developer to check the system programming code is working correctly as per the required actual outcome, this helps in minimizing the defect in an earlier stage and helps to efficient work during integration testing.

# 10). What is functional system testing?

Functional testing is testing the requirement of functions in the application as it is working as it meets specifications.

* Testing each function as per requirement.
* Ensuring the system meets specified functional requirements.
* Meet the business needs intended in the software.

The process usually involves steps: -

* Prepare test cases and test data with the help of the requirement of functionality.
* Find actual outcomes by adding input of test data.
* Compare the expected with the actual outcome.
* Test case execution.
* Comes to the conclusion that functions are performing well or not.

The technique used in Functional testing is Black box testing.

***In a nutshell***

***In a nutshell***

Functional testing is a comprehensive strategy testing that provides the foundation which is a crucial requirement that depends on a team's ability to deliver successful applications.

# 11). What is Non-Functional Testing?

Testing the attributes is nonfunctional testing that relates to reliability, efficiency of application, load test of the application, Stress stability in the application, maintainability also supports as nonfunctional testing and usability of the application supports testing, portability in the application, and performance testing.

* It’s all about the system and how it works.
* At all levels, this is performed and necessity for application.
* It describes the test required to measure characteristics and quantifies on scale its performance.
* Testing to check and fine-tune how systems responded!
* Criteria measured to judge the operation of a system.
* Measured on the scale of quality or standards.
* Testing helps in finding the ability to run software in any conditions such as high traffic, stress taking, breakdown of application, and system failures.

Can be divided into categories such as: -

* Reliability
* Usability.
* Performance.
* Security.

Non-functional Testing Types.

Maintainability Testing

Volume Testing

Security Testing

Scalability Testing.

Stress Testing

Load Testing

Configuration Testing

Usability testing

Failover Testing

***In a nutshell***

Testing is designed for testing the readiness of a system according to the parameters of non-functional requirements.

# 12). What is GUI Testing?

Graphical User Interface is the validation of elements that are functioning according to plan or not.

* For example, there are specifications in the application such as menus, icons, and buttons like login logout are the GUI functionality that needs to check their working status.
* GUI is a graphical User Interface command where you interact with the computer using images instead of text.
* GUI is what the user sees the interface is the interaction part of the user that is formed in graphical format.
* For users, they will not see the source code they see the design part, images, buttons, labels, and icons all are working properly or not that’s in GUI.

Checklist of GUI: Elements with proper size, position, length, width, and acceptance of characters and numbers. Input fields are there with input facility and working.

* Intended functionality working or not?
* Clear demarcation of different sections on screen working.
* Error messages are displayed or not.
* Fonts are clear, visible, and readable.
* Warning messages are aesthetically pleasing.
* Images with clarity, alignment, and links that are working fine.
* Different screen resolutions along with elements working or not.

GUI testing techniques are categorized in three parts.

1. Manual-Based Testing: This approach is based on manually looking at UI for all elements and verifying with specified requirements.
2. Record and play: GUI testing done on automation tools.
3. Model-Based Testing: Testing using the system’s behavior. Build a model, determine the input for testing, calculate the expected outcome, run the test cases, and calculate the actual outcome that decides whether the system passes or fails in GUI testing.

***In a nutshell***

GUI testing is required because users are not interested in the source code or how it is prepared for the user important is application is friendly, and simple to use, and the design of the application got attraction for the user.

1. The software becomes successful only on a User Interface that is user-friendly and covers various features that fulfill user needs or not. Hence GUI testing is very important in the software process.

# **13).** What is ad hoc testing?

Ad hoc testing is to break the process to find defects or bugs at an early stage.

* Aim at an unstructured and informal testing process.
* Ad hoc is not planned before that is done as random activity and unplanned.
* Ad hoc is done without any documentation and no test cases require to do this testing.

Aim of doing Ad hoc:

* The aim of Ad hoc is to find defects done on random parts of the application, here technique used is Error Guessing.
* The experienced person can able to do Ad hoc testing without documentation testing is to “Guess” the most likely errors and their source.
* This testing is unplanned, and unstructured, with no planning, no process to follow, no documentation, and no defects mapped on test cases.
* Sometimes difficult to reproduce defects due to not having test cases, test steps to follow, and no steps of test data mapped.

**Types of Ad hoc testing.**

* Buddy Testing: Two buddies can perform this testing suppose one is a developer and the second is the tester. Buddy in a pair can help each other for better test cases and better development processes in an early stage. This test is possible after Unit testing completion.
* Monkey Testing: Random tests performed to break the process of the system with the aim to find defects in the system.
* Pair Testing: Comparing buddy testing and pair testing in which buddy is the combination of unit and system testing together. (Experienced and non-experienced together work and share their views and ideas).

***In a nutshell***

* Performed for the completeness of testing and performed earlier to enhance quality before stable build occurs for testing, that saves a lot of time as performed earlier stage and doesn’t require any documentation, planning, and test design for referring can be done without this necessity and added to later stages for the betterment of the application.

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

Testing is performed to test the performance of a system, or software application performance on real-life based conditions while taking the load. Multiple users use at the same time the application load that needs to be tested in normal and extreme load-taken situations.

* Real-world load that the system is going to take and how it performs under stress.
* Testing requires to ensure expected usage levels and potential issues before the deployment of the application.
* There are different load conditions to stimulate with various scenarios like a high number of concurrent users, a large number of requests receiving, and stimulating heavy network traffic.

Some techniques of Load testing are

* Stress Testing: Checking high load on normal usage levels handling on the system.
* Spike Testing: Sudden spike shout-out needs to be checked during high load.
* Soak Testing: Sustainable load over a prolonged period of time.

Objectives of Load Testing: Maximize operating capacity.

* Determining the latest infrastructure is able to work smoothly in the system.
* Extreme load taken is sustainable on the system.
* Total count of users.
* Determining scalability.

***In a nutshell***

Load testing is essential to check the potential of the application when more users visit at the same time for exploration and tools are the best medium to check the load stress with less time and more productivity.

# 15). What is Stress Testing?

Stress testing is related to the reliability and stability of the application.

* Checking the software on robustness and measuring the error handling capabilities during the high load stress condition.
* Checking the application is reliable, stable able to take the high load and respond to error handling into the system, and importantly not crash in that deep stress condition.
* It is also known as Endurance Testing.
* How AUT (Application under Test) is performing under high load is getting tested.
* Determining at which point the system hardware or software breaks is stress testing.

Real-time example: when a Shopping app announces discount sales during the festive season it may witness to spike in traffic due to stress faced by the application.

***In a nutshell***

Stress testing is important as sudden spikes in the system during high load result in revenue loss for applications based on User usage. Stress testing helps to perform better in extreme conditions. The main purpose of stress testing is to make sure that the system recovers after a failure this situation is called recoverability.

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

White testing is internal testing of the software its structure, and design, the coding part is tested to verify input-output flow is as required.

* Improvise require in the usability, security, and design of the application or not.
* The code is visible in the software also known as Clear box testing, open box testing, Code-based testing, Transparent Testing, and Glass box testing.
* Two parts are there in the box testing approach: -

1. White box relating to inner workings that revolves around internal testing.
2. Other is Black box testing revolves around the external or end user perspective.

* The clear box or white box symbolizes the ability to see through outer shell into its inner workings.
* White box involves Internal security holes, expected output, loopholes in the coding process, the response of input into specific output, and checking on individual elements, objects, and functions into specified results.

**White box testing is performed by the developer or development team.** There are steps to perform in white box testing.

1. Understand the source code.
2. Create test cases and execute them.

**Types of white box testing**

1. **Unit testing: -** This type of test level that performed on each unit, module, each use case, each component, and block of code as it is developed. Done by the developer or development team or programmer.
2. **Testing for memory leaks: -** Memory leaks are the leading cause of slower-running applications. QA specialist checks memory leaks or reasons for slower running applications.
3. **White Box penetration testing: -** Developer tester has the full source code of the application, detailed network information, the IP server involved, and all other server information on the application runs. Aiming to attack code from several angles that expose several threats if any in the source code.
4. **White Box mutation testing: -** Is done to discover the best coding technique to use for expanding the solution of software.

***In a nutshell***

White box testing is complex. The complexity involved with the application during testing. A single simple operation in the application for programming coding testing takes lesser time while a complex application that takes weeks, or months to check is called White box testing.

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

Black box testing is the functional testing of software applications Which is tested without the knowledge of source code, implementation details, and internal paths of the application.

* Mainly focused on the input and output of the applications that are entirely based on the requirements and specifications of the client.
* Also known as Behavioral testing.
* Any software system is tested in Black Box testing.
* Example: - Operating systems like Windows, a database like Oracle, a website like Google, or any custom application can be tested based on their specified requirements.

**There is different testing technique.**

The techniques and strategies are: -

1. **Equivalence Class Testing: -** Minimize the number of test cases to an optimum level while maintaining the test coverage.
2. **Boundary Value Testing: -** Focused on the boundary values. Determining the certain values range is acceptable by the system or not. Helps in reducing test cases. Most suitable for systems where inputs are at certain ranges.
3. **Decision Table Testing: -** There is a matrix format where data is kept in causes and their effects which is identified by a unique code for each test case.

Generic steps of Black Box Testing.

* Requirements and specifications of systems are examined.
* Tester tests the SUT while putting positive scenarios into a valid input bar for testing along with that negative scenario also puts to detect the applications.
* Tester determines whether the output is expected or not, by executing test cases.
* Analysiation of actual and expected output.
* Defects are fixed if any and the re-test process goes on.

Black box testing tools are QTP and Selenium for functional testing.

Load Runner, JMeter for Non-functional testing.

***In a nutshell***

Black testing follows the Software Testing Life Cycle in this there are phases such as Requirement, Test planning & analysis, Design, and Test Execution, Completion criteria, finalizing closure.

# 18). Mention what are the categories of defects?

A defect in software testing is a deviation or variation in the software from end-user requirements or original specified requirements.

* Error in coding that causes incorrect or unexpected outcome from coding that doesn’t meet the actual requirements is defect.
* While executing the test cases tester come across such defects.
* The variation in test cases results with actual results whatever differs is the defect in the software.
* A bug report in software testing is a detailed document of bugs found in the application.
* Defect report: - Contains a detailed description of bugs, the date the bug was found, the status of the bug fixed or not, the name of the developer, the name of the tester, and the unique id for each bug case.

Categories of defect report helps prioritize the tasks high-priority first and then go down to low-priority defects.

1. Critical Defect: - These are defects due to which the application will not run properly that damage directly revenue and loss and the team has no backup for that defect documentation. That comes under critical. For example, the login function is not working that comes under a critical defect that needs to be fixed.
2. High Priority Defect: - That affects the direct product’s main features, such as the payment gateway not working then this comes under high priority defect that affects the company’s application income, which needs to be fixed on high priority.
3. Medium Priority Defect: - The minimal deviation in the software, sometimes mistakes in spelling errors, or the link which is rarely visited by the user, all come under medium Defect. Example: the GUI of the mobile device is not working on my phone which affects certain users not able to use the app properly.
4. Low Priority Defect: - The defect that affects very minor, such as spelling mistakes that can be fixed on a later basis comes under low priority.

***In a nutshell***

The Defect report helps the next time of avoiding the same bugs into repetition. Saves time with the help of mapping defect report.

# 19). Mention what big bang testing is?

The test levels are performed while validating a software, such as Unit testing, Integration testing, system testing and User acceptance testing.

* The Integration testing is performed before system testing and after unit component testing.
* In integration testing there is big bang technique of testing is performed.
* It’s a testing approach where all modules are integrated and set as a single unit of software that is tested.
* This type of testing approach like big bang is performed when time is limited and the deadline is close for delivering software and development teams are working parallel to deliver the software.
* Example: In one software there are 3 modules that is A, B, C then all integrated as a one module and tested as a whole one unit.

Features on Big Bang Integration: -

* **Simulation of Complete system: -** All components are integrated and tested at the same time.
* **Testing all components together: -** Beneficial of testing as a whole unit interaction in between all the components.
* **No component is left untested: -** Ensures all systems are tested with all aspects at a time.
* **Early detection of errors: -** Testing before system testing helps to correction of error before it gets deployed to system testing.
* **Allows complex interaction: -** While communicating whole as one unit with all elements at a time that detects the error early if not identified by other methods.
* **Behavior of low-level components: -** Low-level components are getting tested by using stubs and drivers which is very useful in testing before getting stable components.
* **Risky: -** This approach is not useful for large or complex projects.
* **End of development cycle: -** When all modules are prepared and integrated then it helps to test their internal system elements are communicating well.
* **Time Consuming and expensive: -** Time consuming because of testing all modules at a time and its very expensive to test together.

***In a nutshell***

The simplest form of integration that is easy to implement and after completion of integration bugs can be identified at an early stage, which helps to make the software bug free before going to system testing.

# 20) . What is the purpose of exit criteria?

In this testing is important for enhancing the software life cycle but at the same time, it’s important to check for exit criteria also. When to enter testing and when to exit is important to deliver software to the client.

Entry Criteria: - There are some prerequisites that need to be fulfilled before testing starts of software.

Exit Criteria: - Exit criteria are also a must after the testing process is completed.

There are entry and exit criteria for all levels in STLC.

In an ideal world, before entering to next levels the exit criteria of the early stage have to be fulfilled. There are activities and deliverables of every phase in STLC that needs to fulfill before getting to exit criteria.

* Requirement Analysis Phase: - In this phase, all requirements should be specified, Priorities of requirements should be set from high to low, preparing a requirement traceability matrix is a must to trace requirements, environment setup details would be gathered, and Automation feasibility analysis should be set.

**Deliverables:** - Requirement Traceability matrix.

* Automation feasible report.

**Exit Criteria: -** Signed off RTM.

* Automation feasible report signed off by the client.
* Test planning: - Test planning preparation, Tools selection, Efforts for test estimation, resources planning and determining roles and responsibilities, and Training required all includes in planning.

**Deliverable: -** Test plan/ Strategy document.

* Effort estimation document.

**Exit Criteria: -** Approved Test plan document.

* Effort estimation document signed off.
* Test case development: - Creating test cases, scripts preparing for automation if required, review and baseline of test cases preparation, create test data is all included in Test case development.

**Deliverable**: - Test scripts/ test cases.

* Test data.

**Exit Criteria: -** Reviewed test cases/test scripts.

* Reviewed and signed off Test data.
* Test Environment setup: - Understanding the required architecture, set-up, preparing hardware and the requirement list, test data, perform a smoke test on the build.

Deliverable: - Environment ready along with test data.

* Smoke test results.

**Exit Criteria: -** Environment setup as per planned.

* Test data setup is completed.
* Smoke test is successfully completed.
* Test execution: - Execution of test cases, documenting test results, logging defect reports for failed cases, mapping detects on RTM to test cases, retesting the defect fixes, track the closure of defects.

**Deliverable**: - Completed RTM with the status of bugs fixed, test case updates with their results, and defect reports.

**Exit Criteria: -** All tests are executed as per planned.

* Defects fixed or tracked their status and closure.
* Test cycle closure: - Test cycle completion based on Test coverage, Cost, Critical business objectives, and Quality, based on specific parameters prepare test metrics, prepare test closure, report of quality, and result analysis of defect from type and severity.

Deliverable: - Test closure report.

* Test metrics.

***Exit criteria***

* Closure report signed off by the client.

***In a nutshell***

The purpose is when to stop doing testing either at the end of all test levels when the product is going live or at the end of the system testing phase toward User Acceptance Testing. Exit criteria are necessary if all the test levels are completed, when time is running out, when the budget is going out of hand when it is going live the next day, when all defects are fixed, or when the boss says stop all considered under exit report.

# 21). When should "Regression Testing" be performed?

Regression testing is performed to test any defects in the modifications, new update functions, or changes caused by any new defects in the relevant unit or component. Typical regression testing is performed after retesting is done.

* Regression in the changes related to the system that includes new features, bug fixes, and improvements in the performance in any occurred because of new features updating.
* *Regression* testing is defined as a type of software testing that confirms a recent code or program has not affected adversely to the existing code or program.
* Testing based on full or partial changes that works on already tested and getting retest phase to check system is working smoothly as required.
* Testing is done to ensure new code changes do not affect to existing functionalities and features.
* Ensuring old codes, and programming is working well even when the latest code changes.

***Need for Regression Testing***

* Whenever there is new code introduced, code is changed then the need of regression testing is required to check that the modified code is not affecting the application. It is performed even after any modification and also performed when functional or defect issue is fixed.

**Scenarios of when to perform regression testing.**

* New functionality added.
* In case of changing requirements.
* After fixing defects.
* Software performance issues are fixed.
* Integrating a new external system.

***In a nutshell***

This testing ensures that the application working as expected even after new changes in the code or functionalities or any updates. Helps in saving time and money due to the efficiency of the application. As per the study, regression helps in saving time approximately more than 60% that is used for bug fixing. Saves 40% of the money due to regression testing. There are different types of regression testing to cover each phase appropriately.

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

There are 7 key principles that are performed by the Software tester and Quality Analyst:

Testing shows presence of defects

Pesticide paradox.

Exhaustive testing is not possible.

Early Testing.

Defect Clustering.

Testing is Context Dependent.

Absence of Error fallacy.

Exhaustive testing is not possible: - Based on risk assessment we have to follow the testing path because testing is the exhaustive path and on being a correct path, determine The important fact of business requirements met is testing but it is not possible to prove it is defect-free or error-free.ng risk is essential.

* It’s also known as brute force testing where the technique is used for all possible combinations of scenarios and inputs to test the application and get confirmation about system functions properly.
* For example: If we have to test the login module of username and password then we have prepared multiple scenarios to test it but in this, there is not any limitation to test it which is exhaustive and not possible also.

Testing shows the presence of defects: The important fact of business requirements met is testing but it is not possible to prove it is defect-free or error-free.

* It is accepted and proven that virtually it is impossible in the software industry.
* Testing is a continuous process for defect searching and for revealing hidden defects.

Pesticide paradox: - In these principles, it says the same set of tests if executed repeatedly then it is not possible to find different defects.

* It’s a theory that says when crops are exposed continually to pesticides then insects will get immunity that builds and to fight or combat it testers must ensure continually review and revise their test cases to find out new test cases without repeating the old test case again and again.

Early Testing.

Defect Clustering.

Absence of Error Fallacy.

Testing is Context-dependent.

# 23) Difference between QA v/s QC v/s Tester?

|  |  |  |  |
| --- | --- | --- | --- |
| **Titles.** | **Quality Assurance** | **Quality Controller** | **Tester** |
|  |  |  |  |
| **1). Quality.** | It’s a Quality Assurance. | It’s a Quality Control. | It’s a Testing. |
| **2). Procedure.** | Focuses on Procedures and Processes considering the quality rather than conducting actual testing on the system. | Focuses on the implementation of procedures and processes by executing actual testing on developed software. | Focuses on actual testing. |
| **3). Activities.** | It’s a Process-oriented activity. | It’s a product-oriented activity. | It’s a product-oriented activity. |
| **4). Technique.** | It’s a preventive activity. | It’s a corrective process. | It’s a preventive process. |
| **5). Verification & Validation.** | It’s a verification activity intended with the level of procedures, processes, and standards of a developed software. | It’s a verification ensuring documented requirements fulfilment of a developed software. | It’s a identification of bugs/error/defects by finding defects and preventing of a developed software. |
| **6). Subset.** | It’s a subset of software test lifecycle (STLC). | It’s a subset of Quality Assurance. | Testing is a subset of Quality Control. |
| **7). Standards & methodologies.** | To meet requirements, follows standards and methodologies. | Follows standards while working on the product. | Follows methods while working on the product. |
| **8). Life Cycle.** | It’s involving software development lifecycle. | It’s involving full software testing cycle. | It’s involving software testing lifecycle. |
| **9). Measure.** | It’s a Proactive measure. | It’s a reactive measure. | It’s an active measure. |
| **10). Technique.** | The technical technique applied is known as statistical Process Control (SPC). | The technical technique applied is known as Statistical Quality Control (SQC). | Testing is a mixture of both technical and professional aspects. |

# 24). Difference between Smoke and Sanity?

|  |  |  |
| --- | --- | --- |
| **Titles** | **Smoke Testing** | **Sanity Testing** |
| 1). Critical Functionalities | Smoke testing is performed to assure the critical functionalities are working fine. | Sanity testing is performed to check new functionalities updated well along with bug fixed. |
| 2). Objectives | Objective is to verify the stability of the system by then proceeding for more rigorous testing. | Objective is to verify the rationality of the system by then proceeding for more rigorous testing. |
| 3). Documentation or not. | Smoke testing is usually scripted or documented. | Sanity testing is usually not scripted or not documented. |
| 4). Done by! | This testing is performed by the testers or developers. | This testing is performed by the testers only. |
| 5). Subset | Smoke testing is a subset of acceptance testing. | Sanity is a subset of regression testing. |
| 6). Type of activity | Smoke testing is on the entire system exercise from end to end. | Sanity testing is on the particular component exercise of the entire system. |
| 7). Specialized | It’s on general health check-up. | It’s on specialized health  check -up. |

***In a nutshell***

Both testing helps us to avoid wasting time and helps in quickly determining is too flawed to merit. Smoke testing also known as build verification testing.

# 25). Difference between verification and Validation?

Verification: - It’s a process of checking whether the developed software has achieved its goal or not without carrying any bugs in it.

Validation: - The process of checking about software products is up to the mark and in other words, it has also known as high-level requirements.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Titles | ***Verification*** | ***Validation*** |
| 1). Activity | Activity includes verifying documents, codes, designing, and programs. | Activity includes validating testing and actual product developing in exact as per requirement. |
| 2). Testing comes under | Verification comes in static testing. | Validation comes in dynamic testing. |
| 3). Code | Doesn’t include the execution of code. | Does include execution of code. |
| 4). Methods | Verification is done by methods like walkthroughs, reviews, inspections, and desk checking. | Validation is done by methods like testing such as functional and non-functional, white box, and black box testing. |
| 5). Confirms | It confirms the specification as per requirements or not. | It confirms whether the expectation meets the requirements of a client or not. |
| 6). Goal | The goal is to design the architecture of software and specification. | The goal is validating an actual product. |
| 7). Bugs | Able to find bugs in earlier stages of the development. | Can find the bugs that are not found in the verification process. |
| 8). Team involves | The team of quality assurance does the verification. | The team of the tester and developing team do the validation by executing software code. |
| 9). Performed by | Consist of checking documents that are performed by humans. | Consist of executing the program and that is performed by the computer. |
| 10). Specification | After the complete specification is valid then verification starts. | Validation begins as soon as the project starts. |
| 11). Errors | Prevention of errors. | Detection of errors. |
| 12). Also known as | Also known as white box testing or static testing as a work product goes through reviews. | Also known as black box testing or dynamic testing as the work product is executed. |
| 13). % in defects | Helps to find out 50-60% of the defects. | Helps to find out 20-30% of the defects. |
| 14). Based on | Based on the opinion of the reviewer and may change from person to person. | Based on this fact and is often stable. |
| 15). Process. | It's about process, guidelines, and standards. | It's about the product itself. |

***In a nutshell***

* Verification is to verify that the specification, requirements all are going according to requirements without executing any codes through walkthrough, reviews.
* Validation is to validate that the requirement is giving actual results of what requirement is all about through executing the codes.

# 26). Explain types of Performance testing?

**Performance** testing is the process of testing response time, speed, stability, scalability, and resource usage of the application under a particular workload. The main goal of this testing is to identify and eliminate the bottlenecks in the application.

* It is also known as “Perf Testing” and the subset of Performance testing.
* The Focus is to check software programs that divide into speed, Scalability, and Stability.

**Types of Performance Testing.**

* **Load Testing: -** Checking the ability to perform under anticipated user loads. The objective is to identify bottlenecks while the application goes live.
* **Volume Testing: -** This relates to a large number of databases being populated and the overall system’s behavior is monitored in volume testing. The objective is to check the large volume of databases and their performance during large numbers of databases.
* **Stress Testing: -** Application testing under extreme workloads where checking takes place of high traffic handling and data processing. The objective is to identify the breaking point of the application during extreme loads and high traffic on the site.
* **Spike Testing: -** Is to check the software's large spikes during sudden load generated by the users. Example: Board result checking websites.
* **Endurance Testing: -** Making sure that software can handle expected load over long period of time.
* **Scalability Testing: -** The objective is to determine effectiveness in “Scaling up” that supports an increase in the load by users. Helps plan capacity that adds to software system.

***In a nutshell* Performance testing** helps consolidate, analyse and share test results. Helps to improvise the application from sudden workloads, high traffic and helps in performing during facing that situation.

# 27). What is Error, Defect, Bug, and failure?

 All these terms seem interchangeable but there is a difference between them. Let's start with: -

|  |  |  |  |
| --- | --- | --- | --- |
| **Error**: - A mistake made by a human is called an error | **Defect: -** The problem in the functioning of the system software is a defect. | **Fault: -** The incorrect step, process and data definition is a fault in the software product. | **Bug: -** Bug is a flaw in a system that causes the software system to behave in an intended manner. |
| That leads to a discrepancy between actual and expected results. | “A flaw in a component or system that fails to perform its required function Is called a defect as per ISTQB. | Due to lack of resources or sometimes not following proper steps results into fault. | Bug is not functioning as it is adhered to as per the requirement set. |

|  |  |  |  |
| --- | --- | --- | --- |
| Error | Defect | Fault | Bug |
| That is generally found out by the tester or testing team. | For example, Incorrect definitions and incorrect statements. | That means the incorporated logic causes error in the application. | Any type of logical error that causes code to break resulting in a bug in the software. |
| Error comes from the misunderstanding done by the development team | The defect is an issue in application coding that affects the whole program. | Unintended behaviour by the program in the application.  Causes warning in the program. | A bug once detected can be reproduced with the help of standard bug reporting templates. |
| To understand the requirements definition that results in buggy code. | Represent efficiency and inability in the application to meet criteria that prevent the application to function for what it is meant for. | It may lead to failure also while going towards release phase or deployed phase. | Most common type of bug is a crash of software due to various reasons. |
| Errors are generated due to wrong syntax, logic, and loop that can impact into end-user experience. | During the development phase the mistake done by the developer at the minor or major point resulting defect. | Minor fault results into high end errors and failure. | Bugs are retreated majorly by giving priority and on urgent basis due to risk of user dissatisfaction. |
|  |  | Preventing can be done by adopting programs like development methodologies, techniques, code analysis, peer review. | Typos in the software are very minor problems but sometimes it creates major disastrous results. |

# 28). Difference between Priority and Severity?

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority is the recognition on which the developer has to resolve the defect. | Severity is the impact at which degree the defect affects on the operation of the product. |
| Priority is categorized into three types: -   * Low: - Doesn’t harming the application and fixes can be done later on before release. * High: - Urgent defect fix. * Medium: - Moderate can be fixed long with other build. | Severity categorized into five types: -   * Critical: - Where application not running and there are no back files for same. * Major: - Hammering the application but need to fix urgent. * Minor: - That can fix along with another task. * Moderate: - That needs to fix as risk is moderate. * Cosmetic: - Affecting looks that need to enhance little more to look good. |
| Associated with scheduling. | Associated with functionality and standards. |
| Indicates bugs should be fixed till what time. | Indicates seriousness on the functionality of the product. |
| It’s the decision of the manager/client. | It’s the decision of the QA engineer. |
| Driven by business values. | Driven by the functionality. |
| Priority value is subjective that depends on project situation that can change over the time. | Severity is objective that less likely to change as per time. |
| Based on customer requirements. | Based on technical aspect of product. |

# 29). What is Bug Life Cycle?

# **Reopened.**

**The** bug life cycle is different from organization to organization and also from project to project. Due to differences in testing methods, testing tools are as per the platforms and projects. The defect life cycle goes into the workflow:

* New: - when the new defect is found by the tester its falls into a status that is “New”. The first stage and for this development team gets documents from the tester to fix bugs accordingly.
* Assigned: - The newly defined defects will get the approval from development team assigned, and the process given to resolve it. When a defect is given to the development team that is identified as a defect then the status will change from New to “Assigned”.
* Open: - IN this the development team opens the defect for fixing the bug. If a team defect is not appropriate then it goes from assigned to “Reject or Deferred State”.
* Fixed: - After fixing the bug or changes in code with the necessary steps team marks this as “Fixed” status.
* Pending Retest: - After fixing the defect development team passes new codes to retest by the tester which then goes from fixing status to “Pending Retest” status.
* Retest: - Tester starts working on new codes for testing to retest work then this process goes into “Retest” status.
* Reopen: - after retesting if that same bug occurs again then the status will come as Reopen then this bug starts again with a life cycle that is going again for Re fixing by the developer team.
* Verified: - After re-testing the bug it got fixed by the developer team then found no more bugs inside it then the tester says the bugs are fixed and the status arrived as “Verified”.
* Closed: - Final state of defect life cycle after fixing defects that have been resolved that will not persist again then this status comes under “Closed” status.

There are some more terms included in the defect life cycle. That is

* Duplicate: - The same concept of bug is repeated again in the concern then the status is stated as Duplicate.
* Rejected: - If DEV not accepting the bug as per Dev if it’s not bug then the status changed from defect to “Rejected”.
* Deferred: - If the bug presently is not required for fixing and if that will release in the next build then the status is assigned as Deferred.
* Not a bug: - If functionality is getting affected, that is considered as not a bug category.

# 30). Explain the difference between Functional testing and Non-Functional testing

|  |  |  |
| --- | --- | --- |
| Parameters/ Coverage. | Functional Testing. | Non-functional Testing. |
| 1. Execution. | Testing performed before non-functional testing. | Testing performed after functional testing. |
| 1. Requirement. | Easy to define functional requirements. | Difficult to define non-functional requirements. |
| 1. Focus. | Focus is on customer’s requirements. | Focus is on customer expectations. |
| 1. Objective. | Software validation takes place. | Performance of software validation takes place. |
| 1. Usage. | Helps in validating behavior of the application. | Helps in validating performance of the application. |
| 1. Manual testing. | Functional testing is easy to execute by manual testing. | Non- functional testing is very hard to execute by manual testing. |
| 1. Requirements. | Functional testing is carried out using specifications. | Non-functional testing is carried out using performance specifications. |
| 1. Functionality. | Describes what product does. | Describes how product works. |
| 1. Testing types. | * Unit testing. * Smoke testing. * User Acceptance testing. * Integration testing. * Regression testing. * Localization. * Globalization. * Interoperability. | * Performance testing. * Volume testing. * Scalability testing. * Load testing. * Stress testing. * Compliance testing. * Portability testing. * Disaster recover testing. * Usability testing. |
|  |  |  |

# 31). Difference between STLC (Software testing life cycle) V/s SDLC (Software development lifecycle)

|  |  |  |
| --- | --- | --- |
| Titles | STLC | SDLC |
| 1). Full form | Software Testing Lifecycle. | Software Development Lifecycle. |
| 2). Meaning | Developed software testing lifecycle. | Software developing lifecycle. |
| 3). Consider as | Consider as a child or successor. | Considered as a predecessor. |
| 4). Aims | Aims towards consistency for delivering quality product. | Aims to deliver product as per customer requirement. |
| 5). | Starts after SDLC completes. | Completed before STLC starts. |
| 6). Who does | Quality analyst plans process to follow as per requirements. | Business analyst and product analyst collects requirements. |
| 7). Intent | Intended with any pitfalls, weaknesses in the system | Intended to overcome hurdle to get software developed as a success. |
| 8). Involves | Planning of test by testing team | Designing phase planned and implemented. |
| 9). Written documents | Test cases and script written in this STLC | Written codes to build system by programming language. |
| 10). Setup | Environment setup prepared and smoke test take place. | Test environment setup takes place to validate it. |
| 11). Bugs | Finding bugs and handover takes place | Test the software and resolved bugs and fixing it. |
| 12). Involves. | Smoke testing, sanity testing, matrix, defect report includes. | Application in a production is tested in real end users. |
| 13). Output | STLC results in a tested software system. | End result is the creation of reusable software system. |
|  |  |  |

# 32). Difference between Test Scenarios, Test Cases, and Test Script?

|  |  |  |
| --- | --- | --- |
| Test Scenario | Test Cases | Test Scripts. |
| Is any Functionality that can be tested. | Set of actions to verify particular functionality working as required. | Set of instructions to test an application automatically. |
| Derived from Business requirement specification and Software requirement Specification. | Derived mostly from scenarios. | Derived mostly from test cases. |
| Focused on what to test. | Focused on what and how to test. | Focused on the results as per expectation. |
| Its one linear creates in less time. | Takes more time in detailing and resources require. | More resources require for script but less time for testing. |
| Focused end to end functionality to test. | Focused on test step, expected results from test data. | Focused different commands to develop a script. |
| Test Scenario | Test Cases | Test Script |
| Main task is to check full functionality of a application. | Main task is to verify compliance as per guidelines, standards and customer requirements. | Main task is to verify without skipping any functionality and validate results as per requirements. |
| Assessing quickly with the scope. | Detecting errors and defects. | Carrying automatic execution of test cases. |
|  |  |  |
|  |  |  |
|  |  |  |

Explain What Test plan in? What is the information that should be covered.

• What is priority?

• What is severity?

• Bug categories are…

• Advantage of Bugzilla.

• Difference between priority and severity

• 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?

• When to used Usability Testing?

• What is the procedure for GUI Testing?

To create HLR & Testcase of Web based (Instagram, WhatsApp web)

• To create HLR and Testcase on this Link. <https://artoftesting.com/>

• Write a scenario of only WhatsApp chat messages

# Scenario of chat in WhatsApp.

1. Verify chat message can search contact number for messaging.
2. Verify messages are typeable as a text.
3. Verify every symbol are editable in chat box.
4. Verify Send button is working.
5. Verify attachment file is uploading files as per requirement.
6. Verify Arrow of sending button is clickable.
7. Verify name is properly readable in personal chat window.
8. Verify name is clickable and showing more about persons sharing photos, videos, files.
9. Verify along with name, last seen time showing is correctly as per standard time zone.
10. Verify image is clickable in person window.
11. Verify video calling is clickable and taking towards calling a person.
12. Verify phone label is clickable and taking towards audio call.
13. Verify 3 button are clickable and showing in detail menu.
14. Verify Scrolling label is working fine.
15. Verify payment function is getting connected to every payment option.
16. Verify camera label is clicking photo as required.
17. Verify audio text messages are working fine as required.
18. Verify days segregation messages are showing correct date and time.
19. Verify GIF is getting loaded and getting selected as required.
20. Stickers are getting selected as per requirement.
21. Verify group creating members are getting selected as required select till 1024 member.
22. Verify notification received as per customised, if number block then no notification.
23. Verify privacy setting is working as required regarding disappearing messages.
24. Verify invite a friend function is working as required.
25. Verify Help centre is taking towards help option of call, messages.
26. Verify App language is getting changed as required.
27. Verify network selection is getting customised as required.
28. Verify call tunes is getting filtered as required.
29. Verify themes are getting changes as required.
30. Verify chat backup is working fine.
31. Verify specific font change option is working as per requirement.
32. Verify transfer chat history is getting connected to backup.
33. Verify status is getting changed as required.
34. Verify story is getting selected as required.
35. Verify wall papers are getting selected as required.
36. Verify status disappearing text, photos, videos is updated only after 24 hours.
37. Verify start your community function is working fine.
38. Verify calls label is taking towards recent call in WhatsApp.
39. Verify contact call label is working and adding contact as required.
40. Verify linked device function is working and getting connected.
41. Verify starred message function is working well.
42. Verify after adding payment option, payment function is sending amount as required.
43. Verify setting label is clickable.
44. Verify search label is clickable and working.
45. Verify multiple people can become admin as per requirement.
46. Verify admin can able to add / delete participant from group.
47. Verify if participant got deleted from group then he/she can’t see the messages from group.
48. Verify silent notification function is working as required.
49. Verify creating group function is working and adding people as required.
50. Verify delivery tick mark function of message working.
51. Verify double tick with different colour showing delivered and read by receiver.

Latest function on chat messages 2023.

1. Silence function is working for unknown calls and unsaved numbers in phone.
2. Verify chat lock privacy function is working for single chat lock or for a group lock.
3. Verify HD photos and Videos can be send /receive from android and IOS.
4. Verify status profile status ringing updating working as per updates status through ringing will get notified.
5. Verifying one account will work for 4 phones.
6. Verify anyone can exit from group without notifying group people.
7. Controlling from anyone can see yours online time will be decide by personal choice.

# Negative Scenario of Chat.

1). Try to send 30 images at a time.

2). Verify sending large files exceeding size limit.

3). Verify sending 100 messages at a time to any application and check the behaviour of chat of same person.

4). Verify forwarded messages can send to maximum how many people at a time.

5). Verify to add more than 256 members in a group.

• Write a Scenario of Pen

1. Verify the length and diameter of pen is as per specification.
2. Verify the material of pen is it plastic, metallic or any other is as per specification or not.
3. Verify the colour is as per specification or not.
4. Verify brand name on pen is visible and long lasting.
5. Verify information displayed on pen would be legible and clearly visible.
6. Verify it is ball pen, ink pen, or gel pen.
7. Verify pen is visible writing on every type of papers.
8. Verify pen weight is as per specification, very heavy pen would not be helpful while writing.
9. Verify pen is with the cap or rotator or twister.
10. Verify the colour of ink as per specification.
11. Verify on which surfaces pen is smooth accept paper, is it smooth on cardboard, rubber surfaces or wall paint.
12. Verify while writing with the pen the ink flow will be consistent without leaving blob.
13. Verify pen ink spillage in different condition downside up pen direction.
14. Verify text written with the pen are erasable or not.
15. Verify waterproof pen is not getting erased if water spill on the text.
16. Verify pen angel while writing with pen normal, straight and with tilting direction.
17. Verify grip pen is supportable towards friction while writing with the pen.
18. Verify pen with multiple ink supports is working fine as per requirement.
19. Verify ball pen and gel pen easily change the refill as required.
20. Verify

Negative aspect on Pen

1). Verify fountain Pen point is sharp then we cannot write on paper.

2). Verify pen nib, if not selected correctly, will broaden the size while writing or doing a signature.

3). Verify pen is not smooth while writing it puts pressure on the fingers.

4). Verify pen grip is not there then it will hurt in fingers for long content to write.

5). Verify pen company name is not visible or vanishes after use sometimes then it can't be referred to by name.

6). Verify quality of the Ink is not good then handwriting will go on the lighter side after a while.

7). Verify pen cap not fitting properly then after some time it is hard to carry the same pen.

8). Verify ball pen point gets warmer when carried in the pocket then it leaks and does spillage.

9). Verify with ball pen we cannot write on laminate, walls, and hard surfaces.

10). Verify we cannot write in glass with a pen.

11). Verify roller ball refill is dried out then need is to change refill.

12). Verify the quality of pen ink is not good so we need to shake the pen up and down and will have to write hard so that ink gets released freely.

13). Roller ball pen smudges on paper due to water based liquid ink.

14). If pen is smudging, then pen holding is important otherwise you can't write from every angle.

15). INK nib is large then while writing ink flow will be more than need to write carefully.

16). Ball pen writing with steep angle will give light handwriting.

17). Pen handwriting, we cannot erase it once written.

18). Fountain pen is getting leakage.

19). Fountain pens are a little messy while spillage.

20). Plastic pens are difficult to recycle and biodegrade.

21). Pen material is not good then it will break when drops from height.

22). The ink colour refill is red but due to the pen cap colour is black then people were misguided due to the mistake.

23). Multifunctional pen ink is not good then it is not used versatile and reliable.

24). Multifunctional pen button pressing is not good quality so we cannot use that colour ink pen for writing.

25). Brush pen nib size is in varied types but not useful for writing.

26). Ball pens aren't used for drawing and fine liners.

27). An ink pen without a refill is useless.

28). Dip pen is useless without an ink bottle carried along.

29). Technical pen size is not correct, and it's got leakage.

30). Felt pens are permanent markers that cannot be removed once used.

31). Pen weather conditions need to check, as per requirement if we require cold weather then a few pens are not working.

32). The look of a pen is not attractive so it's not demanding in the market.

33). Pen holder is not good quality then holding the pen is a difficult task.

34). Tic tac pen is not good quality then Tic tac got broken after a few usages.

35). Glitter pens are not used for professional purposes as ink got dried up very soon.

36). Verify pen at every room temperature from highest to lowest.

37). Verify functioning of a pen at every altitude.

38). Verify functioning of pen at zero degree gravity.

39). Verify the effect of liquids and oil to the text written by pen.

40). Verify if pen immerse in liquid for some time then is i

• Write a Scenario of Pen Stand

# The scenario of ATM Machine.

The scenarios of the ATM machine would cover test cases if the UI (user interface) of the application Verifying all the buttons, text boxes, labels, images, and links are present on the screen.



[This Photo](http://bongsiki.tistory.com/491) by Unknown Author is licensed under [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/3.0/)

1. Checking information available on the screen is visible and legible, readable.
2. Verifying the Colour font and size of font and UI of different objects are as per the specifications and eye-friendly while using the screen.
3. UI is responsive that will fit on different screens and different resolutions of ATM machines as specified.

Functional Test Scenarios for ATM Machine.

1. Buttons of the machine are working smoothly.
2. Touch screen of the ATM machine is smooth and operational.
3. Verify the User is getting the option to select language for the transactions.
4. Labels on the screen are directed to the next window.
5. Verify before showing account details of the user screen should ask for the pin and after validating it should get details.
6. Digit numbers are pressable in ATM machines.
7. Links are taken to the next windows for forget pin or reset pin or new pin setting links.
8. Verify a limited number of attempts of the user to put the pin number correctly.
9. Verify pin is masked formed while entering the pin number.
10. Verify if the pin code is not added successfully after a few attempts the account got blocked
11. Card while swiping shows the details of customer services like withdrawal, deposit, and check balance.
12. Verifying the screen whether it is a touch screen and whether the keypad buttons both are working fine.
13. Verifying a properly inserting a card that detects the bank card is from a different bank then it should show details of a user.
14. Card is inserted incorrectly then the error should show on the screen to insert the card properly and enter credentials in that to log in the account.
15. Verify the user can withdraw money as per their account available balance only.
16. Verifying the user gets transactional information in a printed copy if required.
17. User session time out is very important.
18. Transaction limit amount will be exceeded and shows a message warning informing the same.
19. Applicable fee gets deducted if exceeds the transactional limit for amount withdrawal.
20. User using the card with an expiry date will not get permission to do the transaction on that card.
21. Verify during the transaction happens at a time sudden power cut takes place then the transaction should show the message as null without affecting the balance of the users.
22. Checking other than the withdrawal deposit machine is working fine, taking money and calculating money properly.
23. Deposit machine typing information that is necessary for the third party while accepting money.
24. After adding personal details is taken to the next window to deposit then the session time is up to mark.
25. Reset the new pin in the ATM by inserting the card, then verify whether links getting to OTP with mobile messages or not.
26. For security purposes, only the correct pin code is getting access to services, wrong credentials are not allowed for doing transactions.

***Conclusion: -*** There are many more aspects of checking an ATM machine, these are few to detect errors or bugs in software application. The more scenario the more will get into Conclusion of bug free software.

# Scenario on Table

# Positive Scenario’s

1. Table height should be on a comfortable angle.
2. Table has to be stable for the writing desk.
3. Table has to be specified to take loads.
4. Material of the table has to be of good quality.
5. Wooden table has to be waterproof for rough usage.
6. The cocktail table looks stylish in decorating a room that is compact in size also.
7. Table that is able to hold weight décor should be quality material used.
8. Stone-based surface table is really very heavy but stays in one place.

# Negative Scenario’s

1. Table corners are sharp then it will hurt a person.
2. Table wood quality is not good then it will not be used much.
3. Laminate is not properly stuck then it will come out from the table.
4. Table glass is breakable.
5. Table weight is heavy so it is hard to drag from one place to another.
6. Table drawers are not opening properly because the sharp wood is not finishing well.
7. Table drawer knob is not of good quality.
8. Table weight taken by wheels is not of good quality then it is not easy to use that table.
9. Table supports are not stable then they will be dis-balancing and not good to use.
10. Table surface is not smooth then it is not helpful to use.
11. Table with glass support breaks if hot utensils are put on that surface.
12. Table polish colour is not good and sustainable then it will vanish and not look good.
13. Table with the stone platform is really very heavy and not easy to move on.
14. Metallic rods will get bent after a little damage then it would be disbalancing.
15. Metallic-shaped table are very heavy so sometimes baby playing nearby may get hurt.

# Scenarios of Door.

1. ***Verify*** door is single or bi-folded.
2. Verify door is opening inward or outward.
3. Verify door dimensions are as per specification.
4. Verify material used in door as per specification.
5. Verify PVC material used in doors if specified flame resistant then check for material.
6. Door back plate to lock the door are very heavy and some are very hard while opening and closing door.
7. Polish used for doors has to be of high quality for long duration doors.
8. Verify Folding doors are easily accessible than regular doors.
9. Eye hole doors magnifying glass will have to be highly clear glass for usage.
10. Verify doors has newspaper holder will be spacious to hold newspaper.
11. Timber doors are very long lasting but need maintenance within few months.
12. Verify door is sliding or rotating door.
13. Verify the type of locks in the door.
14. Verify the door locks in the interior side and exterior side.
15. Verify door has stopper or not.
16. Verify the door closes automatically or with spring mechanism.
17. Verify door is not making noise while opening or closing door.
18. Verify extensively use of water on doors is getting damage till which point.
19. Verify door condition in different climate conditions and temperature and humidity.
20. Verify how much force is required to open or door closes.
21. Verify door key is unique as per different locks.
22. Verify door lock will not open without key.
23. Verify door stopper holder in magnet will get proper size to stick and hold door.
24. Verify material it Is wood, glass or any other material.
25. Verify built door is made of glass, then is it transparent or non-transparent.
26. Verify pull and push doors is working fine as per mechanism.
27. Verify door stopper at the correct positioning as per specified.
28. Verify Swiping card lock system is working fine.
29. Verify swiping card is getting at the entry and exit are working as required.
30. Verify limited attempt is allowed to while trying to enter with invalid card, message is getting pop up with card lock.
31. Verify position of valid card while accessing for door.
32. Verify manipulating the machine without card is working or not.
33. Automatic doors: Verify when someone comes in or out, its automatically opens.
34. Verify it should close after person enters.
35. Verify inside/outside functionality works as required and on the button.
36. Verify maintenance on regular intervals as due to censor not works it will not harm person.

# Negative Scenario

1. Door corner is sharp then it will hurt a person.
2. Painting doors colour will vanish after few years of usage.
3. Slider door buffer will come out quickly after that sticking gum will vanish.
4. Car plastic door is covered with moisture vapor, if it vanishes then door is damage.
5. Fabric is used to cover door ,will not be usefull for long term as the gum to stick that fabric will vanish.
6. Wooden doors are very heavy, as sometimes efforts are more to open and close that doors.
7. Glass doors are breakable material if certain heavy material touches then it will break.
8. Patio doors are covered with rubber buffer from all corners and dirt stick around the rubber buffers and becomes very hard to use.
9. Verify swiping with invalid card is it unlock the door or not.
10. Verify Door heaviness with the magnet stopper, as sometimes magnet stopper is unable to hold the door.
11. Verify door knobs fights well as per the door size specifications.