1. What is Exploratory Testing?

Ans:- We have to explore the application to understand the requirement completely, then only we will start testing.

* In Simple Words, to understand the requirement completely,first we have to explore the application, hence it is called as exporatory.
* Test engineer will do testing, when there is no requirement at all or the requirement are missing.

1. What is traceability matrix?

Ans:- To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

* Once the Development Team-lead analyzes the requirements, he will prepare the System Requirement Specification, Requirement Traceability Matrix.
* Types of Traceability Matrix

1. Forward Traceability – Mapping of Requirements to Test cases
2. Backward Traceability – Mapping of Test Cases to Requirements
3. Bi-Directional Traceability - A Good Traceability matrix is the References from test cases to basis documentation and vice versa.
4. What is Boundary value testing?

Ans:- Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges

* Boundary value analysis is a method which refines equivalence partitioning.
* Boundary value analysis generates test cases that highlight errors better than equivalence partitioning.
* The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.
* At those points when input values change from valid to invalid errors are most likely to occur.
* Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design.

1. What is Equivalence partitioning testing?

Ans:- Aim is to treat groups of inputs as equivalent and to select one representative input to test them all

* EP can be used for all Levels of Testing
* Equivalence partitioning is the process of defining the optimum number of tests by:
* Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function,
* Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.
* If we want to test the following IF statement: “If value is between 1 and 100 (inclusive) (e.g value >=1 and value <=100) Then..

1. What is Integration testing?

Ans:- Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

* Integration Testing is a level of the software testing process where individual units are combined and tested as a group.
* There are 2 levels of Integration Testing

1. Component Integration Testing
2. System Integration Testing
3. Component Integration Testing

* Testing performed to expose defects in the interfaces and interaction between integrated components

1. System Integration Testing

* It tests the interactions between different systems and may be done after system testing
* It verifies the proper execution of software components and proper interfacing between components within the solution.

1. What determines the level of risk?

Ans:- A factor that could result in future negative consequences; usually expressed as impact and likelihood

* Risks are of two types

1. Project Risks:- Example of Project risk is Senior Team Member leaving the project abruptly.
2. Product Risk:- Example of product risks would be Flight Reservation system not installing in test environment

1. What is Alpha testing?

Ans:- It is always performed by the developers at the software development site.

* Sometimes it is also performed by Independent Testing Team.
* Alpha Testing is not open to the market and public
* It is conducted for the software application and project.
* It is always performed in Virtual Environment.
* It is the form of Acceptance Testing.
* Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.

1. What is beta testing?

Ans:- It is always performed by the customers at their own site.

* It is not performed by Independent Testing Team.
* Beta Testing is always open to the market and public.
* It is performed in Real Time Environment.
* It is also the form of Acceptance Testing.
* Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.
* It is only a kind of Black Box Testing.
* Betatesting is also known as Field Testing, Pre-release testing and pilot testing.

1. What is component testing?

Ans:- The testing of individual software components.

* Unit testing in Extreme Programming involves the extensive use of testing frameworks
* A unit test framework is used in order to create automated unit tests.
* Unit testing frameworks are not unique to extreme programming, but they are essential to it.
* Unit testing is performed by using the White Box Testing method.
* Component(Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.
* Component Testing – The testing of individual software components.
* Sometimes known as Unit Testing, Module Testing or Program Testing.

1. What is functional system testing?

Ans:- A requirement that specifies a function that a system or system component must perform

* A Requirement may exist as a text document and/or a model
* There is two types of Test Approach

1. Requirement Based Functional Testing
2. Process Based Testing

* Functional System Testing Functionality As below:

Accuracy:- Provision of right or agreed results or effects

Interoperability:- Ability to interact with specified systems

Compliance:- Adhere to applicable standards, conventions, regulations or laws

Auditability:- Ability to provide adequate and accurate audit data

Suitability:- Presence and appropriateness of functions for specified task.

1. What is Non-Functional Testing?

Ans:- Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

* Non-Functional testing checks the Performance reliability, scalability and other non-functional aspects of the software system.
* Non functional testing should be performed after functional testing
* Using tools will be effective for this testing
* Performance parameters like speed , scalability are inputs to non-functional testing
* Nonfunctional testing describes how good the product works
* Tough to do manual testing
* Types of Nonfunctional testing are

1. Performance Testing
2. Load Testing
3. Volume Testing
4. Stress Testing
5. Security Testing
6. Installation Testing
7. Penetration Testing
8. Compatibility Testing
9. Migration Testing
10. What is GUI Testing?

Ans:- Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test.

* GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.
* WHAT DO YOU CHECK IN GUI TESTING?
* Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
* Check you can execute the intended functionality of the application using the GUI Check Error Messages are displayed correctly
* Check for Clear demarcation of different sections on screen
* Check Font used in application is readable
* Approach of GUI Testing

1. MANUAL BASED TESTING
2. RECORD AND REPLAY
3. MODEL BASED TESTING
4. What is Adhoc testing?

Ans:- The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

* Adhoc testing can be achieved with the testing technique called Error Guessing.
* Adhoc testing is an informal testing type with an aim to break the system.
* In fact is does not create test cases altogether!
* Main aim of this testing is to find defects by random checking
* Adhoc testing can be achieved with the testing technique called Error Guessing.
* Types of Adhoc Testing

1. Buddy Testing
2. Pair Testing
3. Monkey Testing
4. What is load testing?

Ans:- Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails

* Load Testing is done in order to check when the application fails by increasing the number of users and keeping the system resources as constant.
* Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneousl
* This testing usually identifies –
* The maximum operating capacity of an application
* Determine whether current infrastructure is sufficient to run the application
* Sustainability of application with respect to peak user load
* Number of concurrent users that an application can support, and scalability to allow more users to access it.
* It is a type of non-functional testing. Load testing is commonly used for the Client/Server, Web based applications – both Intranet and Internet.
* Some extremely popular sites have suffered serious downtimes when they get massive traffic volumes. E-commerce websites invest heavily in advertising campaigns, but not in Load Testing to ensure optimal system performance, when that marketing brings in traffic.
* Why Load Testing?
* Load testing gives confidence in the system & its reliability and performance.
* Load Testing helps identify the bottlenecks in the system under heavy user stress scenarios before they happen in a production environment.
* Load testing gives excellent protection against poor performance and accommodates complementary strategies for performance management and monitoring of a production environment.
* Load testing will determine whether system needs to be fine-tuned or modification of hardware and software is required to improve performance.
* Strategies of Load Testing
* Manual Load Testing
* In house(Organization) developed load testing tools
* Open source load testing tools
* Enterprise(Record and Play) load testing tools
* Load Testing is to test the system under normal workload conditions, and it is just testing or simulating with the actual workload.
* Load testing does not break the system

1. What is stress Testing?

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

* It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.
* Stress testing is also known as endurance testing.
* Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.
* Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.
* The main purpose of stress testing is to make sure that the system recovers after failure which is called as recoverability.
* The goal of stress testing is to analyze the behavior of the system after failure. For stress testing to be successful, system should display appropriate error message while it is under extreme conditions
* Types of Stress Testing
* Application Stress Testing
* Transactional Stress Testing
* Systemic Stress Testing
* Exploratory Stress Testing
* Stress testing Tools
* Stress Tester
* Neo Load
* App Perfect
* Metrics for Stress Testing
* Measuring Scalability & Performance

1. Pages per Second
2. Throughput
3. Rounds

* Application Response
* Hit time
* Time to the first byte
* Page Time
* Failures
* Failed Connections
* Failed Rounds
* Failed Hits

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

Ans:- : Testing based on an analysis of the internal structure of the component or system.

* Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.
* White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.
* the types of white box testing
* Test/Code Coverage
* Statement coverage
* Decision coverage
* Condition coverage
* Other White Box Techniques
* Branch Condition testing
* Branch Condition Combination testing
* Modified Condition Decision testing
* Dataflow testing
* Linear Code Sequence And Jump (LCSAJ) testing

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

Ans:- Testing, either functional or non-functional, without reference to the internal structure of the component or system.

* Specification-based testing technique is also known as ‘black-box’ or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
* The testers have no knowledge of how the system or component is structured inside the box. In black-box testing the tester is concentrating on what the software does, not how it does it.
* The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
* The tester is oblivious to the system architecture and does not have access to the source code.
* a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. Input Output If Output = Expected result then pas
* Techniques of Black Box Testing
* Equivalence partitioning
* Boundary value analysis
* Decision tables
* State transition testing
* Use-case Testing
* Other Black Box Testing
* Syntax or Pattern Testing

1. Equivalence partitioning

* Aim is to treat groups of inputs as equivalent and to select one representative input to test them all

1. Boundary value analysis

* Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges

1. Decision tables

* Decision Table the relationships between the inputs and the possible outputs are mapped together
* Inputs are usually defined in terms of actions which are Boolean (true or false)

1. State transition testing

* State transition testing is used where some aspect of the system can be described in what is called a ‘finite state machine’. This simply means that the system can be in a (finite) number of different states, and the transitions from one state to another are determined by the rules of the ‘machine’. This is the model on which the system and the tests are based.
* State Transition Testing uses the following terms:
* State Diagram: A diagram that depicts the states that a component or system can assume, and shows the events or circumstances that cause and/or result from a change from one state to another.
* State Diagram: A diagram that depicts the states that a component or system can assume, and shows the events or circumstances that cause and/or result from a change from one state to another.
* State Transition: A transition between two states of a component or system.
* State Transition Testing: A black box test design technique in which test cases are designed to execute valid and invalid state transitions. Also known as N-switch testing.

1. Mention what are the categories of defects?

Ans:- Types of Defect

* Data Quality/Database Defects:
* Critical Functionality Defects:
* Functionality Defects:
* Security Defects:
* User Interface Defects:

1. Mention what bigbang testing is?

Ans:- In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

* Big Bang testing has the advantage that everything is finished before integration testing starts.
* The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.
* Here all component are integrated together at once, and then tested.

1. What is the purpose of exit criteria?

Ans:- : Exit Criteria defines the items that must be completed before testing can be concluded.

* In an Ideal world, you will not enter the next stage until the exit criteria for the previous stage is met. But practically this is not always possible.
* Exit criteria is used to determine when testing at any stage is complete The set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed.

1. When should "Regression Testing" be performed?

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

* when the system is stable and the system or the environment changes
* when testing bug-fix releases as part of the maintenance phase
* Change in requirements and code is modified according to the requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix

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

Ans:-

1. Testing shows presence of Defects
2. Exhaustive Testing is Impossible!
3. Early Testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent
7. Absence of Errors Fallacy
8. Testing shows presence of Defects

Ans:- Testing can show that defects are present, but cannot prove that there are no defects.

* Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.
* As we find more defects, the probability of undiscovered defects remaining in a system reduces.
* However Testing cannot prove that there are no defects present

1. Exhaustive Testing is Impossible!

Ans:- Testing everything including all combinations of inputs and preconditions is not possible.

* So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
* For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.
* This is very unlikely that the project timescales would allow for this number of tests.
* So, accessing and managing risk is one of the most important activities and reason for testing in any project.
* We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions).

1. Early Testing

Ans:- Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.

* Testing activities should start as early as possible in the development life cycle
* Remember from our Definition of Testing, that Testing doesn’t start once the code has been written!

1. Defect Clustering

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

* Defects are not evenly spread in a system
* They are ‘clustered’
* Similarly, most operational failures of a system are usually confined to a small number of modules

1. The Pesticide Paradox

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

* To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
* Testing identifies bugs, and programmers respond to fix them
* As bugs are eliminated by the programmers, the software improves
* As software improves the effectiveness of previous tests erodes

1. Testing is Context Dependent

Ans:- Testing is basically context dependent.

* Testing is done differently in different contexts
* Different kinds of sites are tested differently.
* For example

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

1. Absence of Errors Fallacy

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

* If we build a system and, in doing so, find and fix defects ....
* It doesn’t make it a good system
* Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectations

1. Difference between QA v/s QC v/s Tester

Ans:-

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Quality Assurance | Quality Control | Testing |
| 1 | Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. | Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements. | Activities which ensure the identification of bugs/error/defects in the Software. |
| 2 | 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 |
| 3 | Process oriented activities. | Product oriented activities. | Product oriented activities. |
| 4 | Preventive activities. | It is a corrective process. | It is a preventive process. |
| 5 | It is a subset of Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance. | Testing is the subset of Quality Control. |

1. Difference between Smoke and Sanity?

Ans:-

|  |  |  |
| --- | --- | --- |
| Sn | Smoke Testing | Sanity Testing |
| 1 | Verifies Critical Functionalities Application Starts Successfully? | Verifies new functionality, bug fixes in the build |
| 2 | This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
| 3 | Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| 4 | Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |
| 5 | Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| 6 | Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

1. Difference between verification and Validation

Ans:-

|  |  |  |
| --- | --- | --- |
| Criteria | Verification | Validation |
| Definition | The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase | The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements. |
| Objective | To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements. | To ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment. |
| Question | Are we building the product right? | Are we building the right product? |
| Evaluation Items | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| Activities | ∙ Reviews  ∙ Walkthroughs  ∙ Inspections | ∙ Testing |

1. Explain types of Performance testing.

Ans:-

1. Load Testing
2. Stress testing
3. Endurance testing
4. Spike testing
5. Volume testing
6. Scalability testing

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

Ans:- Error:- A mistake in coding is called error

* A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.

Defect:- error found by tester is called defect

* Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

Bug:- defect accepted by development team then it is called bug

* A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.

Failure:- build does not meet the requirements then it is failure

* The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

1. Difference between Priority and Severity

Ans:-

|  |  |  |
| --- | --- | --- |
| Sn | Priority | Severity |
| 1 | Priority is Relative and Business-Focused. | Severity is absolute and Customer-Focused |
|  | Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements. | It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system. |
| Ex | If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it. | If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low. |

1. What is Bug Life Cycle?

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

The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as ‘Defect Life Cycle’.

* Defect Stages
* 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 the 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 testers end, the status assigned is “pending retest.”
* Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
* Defect Stages(Cont…)
* Verified: The tester re-tests the 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 is “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 tester assigns the status “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 of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs
* Not a bug:If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

1. Explain the difference between Functional testing and NonFunctional testing

Ans:-

|  |  |
| --- | --- |
| Functional Testing | Non-Functional Testing |
| Functional testing is performed using functional specification provided by the client and the system against the functional requirements. | Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system. |
| Functional testing is executed first | Non functional testing should be performed after functional testing |
| Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. |
| Functional testing describes what the product does | Nonfunctional testing describes how good the product works |
| Easy to do manual testing | Tough to do manual testing |
| Types of Functional testing are  ∙ Unit Testing  ∙ Smoke Testing  ∙ Sanity Testing  ∙ Integration Testing  ∙ White box testing  ∙ Black Box testing  ∙ User Acceptance testing  ∙ Regression Testing | Types of Nonfunctional testing are  ∙ Performance Testing  ∙ Load Testing  ∙ Volume Testing  ∙ Stress Testing  ∙ Security Testing  ∙ Installation Testing  ∙ Penetration Testing  ∙ Compatibility Testing  ∙ Migration Testing |

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

Ans:-

|  |  |  |
| --- | --- | --- |
| SN | STLC | SDLC |
| 1 | Stlc full form is software testing life cycle | Sdlc full form is software development life life cycle |
| 2 | This article is about**Software Testing Life Cycle** (STLC), which involves a series of activities to ensure software quality goals are met and consists of six major phases: Requirement Analysis, Test Planning, Test case development, Test Environment setup, Test Execution and Test Cycle closure. | SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are a number of different development models. |
| 3 | STLC Phases  1. Requirement Analysis  2. Test Planning  3. Test case development  4. Test Environment setup  5. Test Execution  6. Test Cycle closure | SDLC Phases  Requirements Collection/Gathering  Analysis  Design  Implementation  Testing  Maintenance |

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

Ans:-

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Test scenarios | Test cases | Test script |
| 1 | A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility. | Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks. | A test script in software testing is a set of instructions that will be performed on the system under test to test that the system functions as expected. |
| 2 | Test Scenario is ‘What to be tested’ | Test Case is ‘How to be tested’ | One script is written to explain how to simulate each business scenario |
| 3 | Test scenario is nothing but test procedure. | Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition. | The Test Script can be manual or automated |
| 4 | The scenarios are derived from use cases. | Test cases are derived (or written) from test scenario. | The Test Procedures Specification specifies the sequence of actions for a test, i.e. one or more Test Cases. It is also known as a Test Script |

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

Ans:- Test Planning in STLC is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project.

* The Test Plan gets prepared and finalized in the same phase.
* Moreover, the resources, test environment, test limitations and the testing schedule are also determined.
* The Test Case Development Phase involves the creation, verification and rework of test cases & test scripts after the test plan is ready.
* Test Environment Setup decides the software and hardware conditions under which a work product is tested.
* Test Execution Phase is carried out by the testers in which testing of the software build is done based on test plans and test cases prepared.
* Test Cycle Closure phase is completion of test execution which involves several activities like test completion reporting, collection of test completion matrices and test results.

1. What is priority?

Ans:- Priority is Relative and Business-Focused.

* Priority defines the order in which we should resolve a defect.
* Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect.
* If high priority is mentioned then the developer has to fix it at the earliest.
* The priority status is set based on the customer requirements.

1. What is severity?

Ans:- Severity is absolute and Customer-Focused.

* It is the extent to which the defect can affect the software.
* In other words it defines the impact that a given defect has on the system.

1. Bug categories are…

Ans:-

* Security,
* Database,
* Functionality (Critical/General),
* UI

1. Advantage of Bugzila

Ans:-

1. Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.
2. Bugzilla is a defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other test case management tools like Quality Center, Testlink etc.
3. This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the datamanagement chain
4. Key features of Bugzilla includes
5. Advanced search capabilities
6. E-mail Notifications
7. Modify/file Bugs by e-mail
8. Time tracking
9. Strong security
10. Customization
11. Customization
12. What are the different Methodologies in Agile Development Model?

Ans:- The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

* the different Methodologies in Agile Development Model
* Scrum
* Kanban
* DSDM (Dynamic Software Development Method)
* FDD (Feature Driven Development)

1. When to used Usability Testing?

Ans:- Usability testing is a useful way to identify design issues and measure how easily users can accomplish given tasks when using your product, so you can make improvements to your designs. You can run usability tests at**any point in the design process**. In fact, the earlier the better.

* Aesthetics and design are important. How well a product looks usually determines how well it works
* There are many software applications / websites, which miserably fail, once launched, due to following reasons –
* Where do I click next?
* Which page needs to be navigated?
* Which Icon or Jargon represents what?
* Error messages are not consistent or effectively displayed
* Session time not sufficient.
* It helps uncover usability issues before the product is marketed.
* It helps improve end user satisfaction
* It makes your system highly effective and efficient
* It helps gather true feedback from your target audience who actually use your system during usability test. You do not need to rely on “opinions” from random people.

1. What is the procedure for GUI Testing?

Ans:- Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

* WHAT DO YOU CHECK IN GUI TESTING?
* Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
* Check you can execute the intended functionality of the application using the GUI
* Check Error Messages are displayed correctly
* Check for Clear demarcation of different sections on screen
* Check Font used in application is readable
* Check the alignment of the text is proper
* Check the Color of the font and warning messages is aesthetically pleasing
* Check that the images have good clarity
* Check that the images are properly aligned
* Check the positioning of GUI elements for different screen resolution.
* Approach of GUI Testing
* MANUAL BASED TESTING
* Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.
* RECORD AND REPLAY
* GUI testing can be done using automation tools. This is done in 2 parts. During Record , test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test. Example of such tools – QT
* MODEL BASED TESTING
* A model is a graphical description of system’s behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements.