* What is Exploratory Testing ?

Ans - Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits

* Is not random testing but it is Adhoc testing with purpose of find bugs .
* Is structured and rigorous.
* Is highly teachable and manageable.
* Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently.
* 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.

* A software process should help you keeping the virtual table up-to-date.
* Simple technique may be quite valuable (naming convention).
* What is the 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 (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design.

* 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.

* If one value finds a bug, the others probably will too.
* If one doesn't find a bug, the others probably won't either.
* What is Integration testing ?

Ans - Integration Testing - 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

* What determines the level of risk ?

Ans - Risk – ‘A factor that could result in future negative consequences; usually expressed as impact and likelihood’

* When testing does find defects, the Quality of the software system increases when those defects are fixed
* Risks are of two types

1) Project Risks

2) Product Risk

* 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 always performed within the organization.
* 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.
* It comes under the category of both White Box Testing and Black Box Testing.
* What is beta testing ?

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

* 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.
* It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area.
* Beta testing can be considered “pre-release” testing.
* Pilot Testing is testing to product on real world as well as collect data on the use of product in the classroom.
* What is component testing ?

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

* Component Testing – The testing of individual software components.
* Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.
* Unit testing is the first level of testing and is performed prior to Integration Testing.
* Sometimes known as Unit Testing, Module Testing or Program Testing
* Unit testing frameworks, drivers, stubs and mock or fake objects are used to assist in unit testing.
* Functional and Non-Functional testing
* Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.
* Unit testing is performed by using the White Box Testing method.
* Three steps:
* 1. Design test that defines how you think a small part of the software should behave (Incremental development).
* 2. Make the test run as easily and quickly as you can. Don't be concerned about the design of code, just get it to work!
* 3. Clean up the code. Now that the code is working correctly, take a step back and refactor to remove any duplication or any other problems that were introduced to get the test to run.
* What is functional system testing ?

Ans - Functional System Testing : 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 tasks |

* What is Non-Functional Testing ?

Ans - It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.

* The term non-functional testing describes the tests required to measure characteristics of systems and software that can be quantified on a varying scale, such as response times for performance testing.
* To address this issue, performance testing is carried out to check & fine tune system response times. The goal of performance testing is to reduce response time to an acceptable level
* Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system.
* 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
* 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.
* What is Adhoc testing ?

Ans - Adhoc testing is an informal testing type with an aim to break the system.

* It does not follow any test design techniques to create test cases.
* In fact is does not create test cases altogether!
* This testing is primarily performed if the knowledge of testers in the system under test is very high.
* Types of Adhoc Testing
* There are different types of Adhoc testing and they are listed as below:

1. Buddy Testing :-

Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the testers develop better test cases and development team can also make design changes early. This testing usually happens after unit testing completion.

1. Pair testing :-

Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the tests and another person can take notes on the findings. Roles of the persons can be a tester and scriber during testing.

3. Monkey Testing :-

Randomly test the product or application without test cases with a goal to break the system.

* What is load testing ?

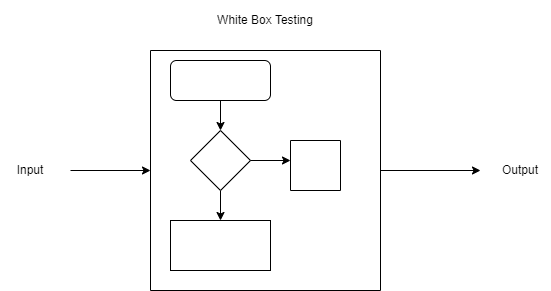
Ans - Load Testing is to test the system behavior under normal workload conditions, and it is just testing or simulating with the actual workload.

* Load testing identifies the bottlenecks in the system under various workloads and checks how the system reacts when the load is gradually increased.
* Load testing does not break the system
* What is stress testing ?

Ans - Stress testing is to test the system behavior under extreme conditions and is carried is out till the system failure.

* Stress testing determines the breaking point of the system to reveal the maximum point after which it breaks.
* Stress testing tries to break the system by testing with overwhelming data or resources.
* What is white box testing and list the types of white box testing ?

Ans -

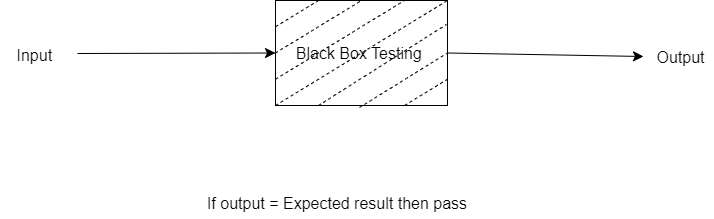


* Based on code and the design of the system
* The tests provide the ability to derive the extent of coverage of the whole application
* Techniques

1. Statement coverage
2. Branch Coverage
3. Decision Coverage

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

Ans -



* Based on requirements
* From the requirements, tests are created
* Specification Models can be used for systematic test case design
* Techniques :-

1. Equivalence Partitioning

2) Boundary Value Analysis

3) Decision Tables

4) State Transition Testing

5) Use Case Testing

* Mention what are the categories of defects ?

Ans – Categories of defect :-

* Error of Commission
* Error of Omission
* Error of Clarity
* Error of speed or Capacity
* Mention what big bang 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.

* What is the purpose of exit criteria?

Ans -Purpose of exit criteria :-

- End of all testing – i.e. product Go Live

* End of phase of testing (e.g. hand over from System Test to UAT)
* Successful Testing of Integrated Application.
* Executed Test Cases are documented.
* All High prioritized bugs fixed and closed .
* When should "Regression Testing" be performed ?

Ans – Change in requirements and code is modified according to the requirement

* New feature is added to the software
* Defect fixing
* Performance issue fix
* What is 7 key principles? Explain in detail ?

Ans – 7 key principles

* General Testing Principles

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

1. Testing shows presence of Defects :-

* Testing can show that defects are present, but cannot prove that there are no defects.
* We test to find Fault

1. Exhaustive Testing is Impossible! :-

* Testing everything including all combinations of inputs and preconditions is not possible.
* Why do not Testing Everything?
* Exhaustive testing of complex software applications:

- requires enormous resources

- is too expensive

* - takes too long
* Why do not Testing Everything?
* Examples:

**System has 20 screens**

**Average 4 menus / screen**

**Average 3 options / menu**

**Average of 10 fields / screen**

**2 types of input per field**

**Around 100 possible values**

**Approximate total for exhaustive testing**

**20 x 4 x 3 x 10 x 2 x 100 = 480,000 tests**

**Test length = 1 sec then test duration = 17.7 days**

**Test length = 10 sec then test duration = 34 weeks**

**Test length = 1 min then test duration = 4 years**

**Test length = 10 mins then test duration = 40 years!**

1. Early Testing :-

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

4) Defect Clustering :-

- Defects are not evenly spread in a system

- They are ‘clustered’

5) Pesticide Paradox

* If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
* To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

6) Testing is Context Dependent :-

- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software

- 1 to 3 failures per KLOC typical for industrial software

- 0.01 failures per KLOC for NASA Shuttle code!

7) Absence of Errors Fallacy :-

- Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectation

* Difference between QA v/s QC v/s Testing ?

Ans – ***QA v/s QC v/s Testing :-***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Sr.no*** | ***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. |

* Difference between Smoke and Sanity ?

Ans - ***Smoke v/s Sanity :-***

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

* Difference between verification and Validation

Ans - ***Verification & Validation Phase :-***

|  |  |  |
| --- | --- | --- |
| ***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 |

* Explain types of Performance testing ?

Ans - Types of Performance Testing

1 Load testing

2 Stress testing

3 Endurance testing

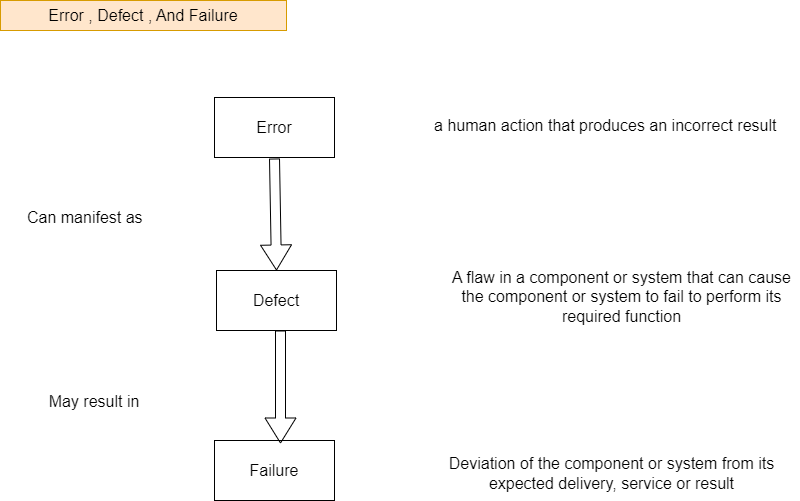
4 Spike testing

5 Volume testing

6 Scalability testing

* What is Error, Defect, Bug and failure?

Ans -



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.

* Difference between Priority and Severity ?

Ans - ***Priority and Severity***

|  |  |
| --- | --- |
| ***Priority*** | ***Severity*** |
| Priority is a parameter to decide the order in which defects should be fixed. | Severity is parameter to denote the impact of particular defect on the software. |
| Priority means how fast defect has to be fixed. | Severity means how severe defect is affecting the functionality. |
| Priority is related to scheduling to resolve the problem. | Severity is related to the quality standard. |
| Product manager decides the priorities of defects. | Testing engineer decides the severity level of the defect. |
| It value is subjective. | It value objective. |
| Its value changes from time to time. | It value doesn’t change from time to time. |
| Priority is of 3 types:   1. Low 2. Medium 3. High | Severity is of 5 types:   1. Critical 2. Major 3. Moderate 4. Minor 5. Cosmetic |

* 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.”

* As you can see from above diagram, a defect‘s state can be divided into Open or Closed.

* 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.”
* 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”.
* Explain the difference between Functional testing and Non Functional testing ?

Ans ***- Functional testing and Non Functional testing :-***

|  |  |
| --- | --- |
| ***Functional Testing*** | ***Non-Functional Testing*** |
| Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements. | Non-Functional 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 Testin  • 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 |

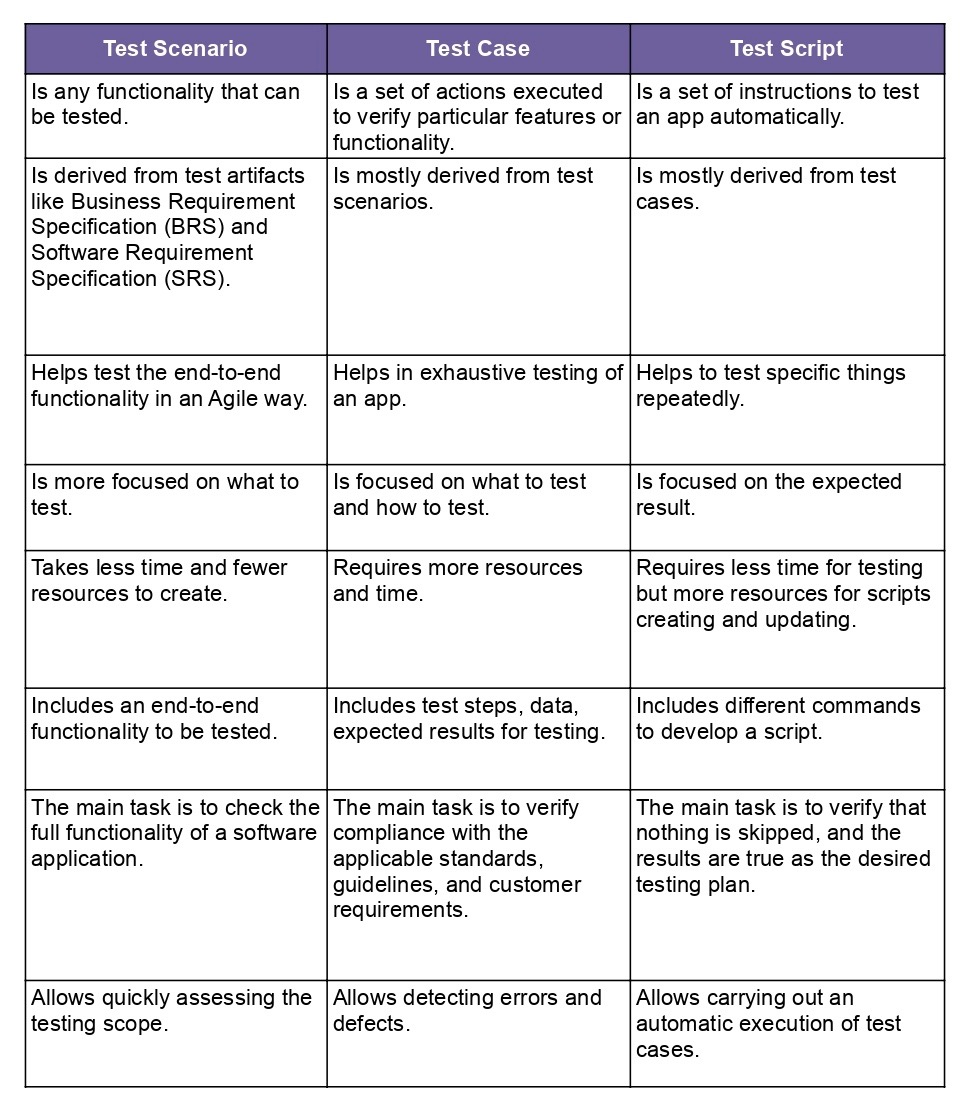
* What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

Ans ***- Difference between SDLC and STLC:***

| ***SDLC*** | ***STLC*** |
| --- | --- |
| SDLC is mainly related to software development. | STLC is mainly related to software testing. |
| Besides development other phases like testing is also included. | It focuses only on testing the software. |
| SDLC involves total six phases or steps. | STLC involves five only phases or steps. |
| In SDLC, more number of members (developers) are required for the whole process. | In STLC, less number of members (testers) are needed. |
| In SDLC, development team makes the plans and designs based on the requirements. | In STLC, testing team(Test Lead or Test Architect) makes the plans and designs. |
| Goal of SDLC is to complete successful development of software. | Goal of STLC is to complete successful testing of software. |
| It helps in developing good quality software. | It helps in making the software defects free. |
| SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |
| Post deployment support , enhancement , and update are to be included if necessary. | Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts. |
| Creation of reusable software systems is the end result of SDLC. | A tested software system is the end result of STLC. |

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

Ans - Test Scenario ,Test Case and Test Script



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

Ans – A test plan is a detailed document which describes software testing areas and activities . It outlines the test strategy, objectives, test schedule, required resources (human resources, software and hardware), test estimation and test deliverables. The test plan is a base of every software’s testing.

* What is priority?

Ans – Priority is the order in which developer has to fix the bug. 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.

* What is severity?

Ans – Severity is how seriously the bug is affecting the application. The severity type is defined by the tester based on the written test cases and functionality. Software Testing Question & Answer For Interview.

* Bug categories are…

Ans – Bug categories : Security , Database , Functionality (Critical/General) , UI

* Advantage of Bug zila .

Ans - 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.

* This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the datamanagement chain.
* Key features of Bugzilla includes
* Advanced search capabilities
* E-mail Notifications
* Modify/file Bugs by e-mail
* Time tracking
* Strong security
* Customization Localization
* Difference between priority and severity ?

Ans – Priority is the order in which developer has to fix the bug. 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.

* Severity is how seriously the bug is affecting the application. The severity type is defined by the tester based on the written test cases and functionality. Software Testing Question & Answer For Interview.
* What are the different Methodologies in Agile Development Model ?

Ans – The Agile methodologyis 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 workbegins, teams cycle

through a process of planning, executing, and evaluating. Agile is a philosophy, i.e., a set of values and principles to make a decision developing software.

There are 5 main Agile methodologies:

* Scrum,
* Kanban
* Extreme Programming (XP)
* Lean Development e Crystal.
* Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

Ans ***– Authorization And Authentication***

|  |  |
| --- | --- |
| ***Authorization*** | ***Authentication*** |
| Authorization determines what resources a user can access. | Authentication verifies who the user is. |
| Authorization works through settings that are implemented and maintained by the organization. | Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user. |
| Authorization always takes place after authentication. | Authentication is the first step of a good identity and access management process. |
| Authorization isn’t visible to or changeable by the user. | Authentication is visible to and partially changeable by the user. |
| Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization. | Example: By verifying their identity, employees can gain access to an HR application that includes their personal pay information, vacation time, and 401K data. |
| Popular  Authorization Techniques-   * Role-Based Access Controls (RBAC) * [JSON web token (JWT) Authorization](https://www.geeksforgeeks.org/json-web-token-jwt/) * SAML Authorization * OpenID Authorization * OAuth 2.0 Authorization | Popular Authentication Techniques-   * Password-Based Authentication * Passwordless Authentication * 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) * [Single sign-on (SSO)](https://www.geeksforgeeks.org/introduction-of-single-sign-on-sso/) * Social authentication |
| The user authorization is not visible at the user end. | The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc. |

**Below are five web application testing challenges faced by web developers during the development process.**

* Integration. Integration testing exposes problems with interfaces among different program components before deployment. ...
* Interoperability. ...
* Security. ...
* Performance. ...
* When to used Usability Testing?

Ans – If possible, usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you've begun the strategy work around a brand new site or app

* What is the procedure for GUI Testing?

Ans - 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.