1. **What is Exploratory Testing?**

* Exploratory testing is a concurrent process where,
* Test design, execution and logging happen simultaneously
* Testing is often not recorded
* Makes use of experience, heuristics and test patterns
* Testing is based on a test charter that may include
  + Scope of the testing (in and out)
  + The focus of exploratory testing is more on testing as a “thinking” activity.
* Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking.

1. **What is Traceability Matrix?**

* Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability.
* Traceability can be horizontal through all the test documentation for a given test level or it can be vertical through the layers of development documentation.
* To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.



1. **What is Boundary Value Testing (B.V.A)?**

* Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.
* It is a method which refines equivalence partitioning.
* It generates test cases that highlight errors better than 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.

1. **What is Equivalence Partitioning (EP) Testing?**

* 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.

1. **What is 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.
* Test drivers and test stubs are used to assist in Integration Testing.
* Integration testing is done by a specific integration tester or test team.
* Components may be code modules, operating systems, hardware and even complete systems.
* There are 2 levels of Integration Testing
  + Component Integration Testing
  + System Integration Testing

1. **What determines the level of risk?**

* Determining the level of risk usually involves trying to assess not only the likelihood of an identified risk from actually occurring, but also the potential magnitude the consequences this risk could have on an organisation and its stakeholder, should it occur.
* Once risks are identified through testing and analysis, they can then be assigned a ‘level of risk’ based on a set of pre-conceived criteria. For example, ‘High’, ‘Medium’ and ‘Low’ risks may be categorised.

1. **What is Alpha Testing?**

* 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.

1. **What is Beta Testing (Field Testing)?**

* 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 usually conducted for software product.
* It is performed in **Real Time Environment**.
* It is always performed outside the organization.
* 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.
* Beta Testing is always performed at the time when software product and project are marketed.
* 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 a product in real world as well as collect data on the use of product in the classroom.

1. **What is Component (Unit) Testing?**

* A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”
* Component 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 tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.
* Unit tests find problems early in the development cycle.
* Unit testing is performed by using the White Box Testing method.

1. **What is Functional System Testing?**

* Functional System Testing is 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 are two types of Test Approach
  + Requirement Based Functional Testing
  + 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 |

1. **What is Non Functional System Testing?**

* Non-Functional System Testing is 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.
* 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.
* Performance parameters like speed, scalability are inputs to non-functional testing.
* Nonfunctional testing describes how good the product works
* Examples of Non Functional Testing are:
  + WEB BASED TESTING
  + Identify the software processes that directly influence the overall performance of the system.
  + DESKTOP BASED TESTING
  + Guarantee that error messages are instructive and helpful for the client
  + MOBILE BASED TESTING
  + In mobile, automatically will switch off without any reason.
  + GAME BASED TESTING
  + Confirms workability and stability of the software.

1. **What is GUI Testing?**

* Graphical User Interface (GUI) testing is the process of testing the system’ 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.
* Following can be checked 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 - QTP.

* + 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.

1. **What is Adhoc Testing?**

* 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.
* Main aim of this testing is to find defects by random checking.
* Adhoc testing can be achieved with the testing technique called Error Guessing.
* 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.
* It also saves a lot of time because of the assumptions and guessing made by the experienced testers to find out the defects which otherwise won’t be able to find.
* There are different types of Adhoc testing and they are listed as below:
  + BUDDY TESTING
  + Two buddies mutually work on identifying defects in the same module.
  + PAIR TESTING
  + Two testers are assigned modules, share ideas and work on the same machines to find defects.
  + MONKEY TESTING
  + Randomly test the product or application without test cases with a goal to break the system.

1. **What is Load Testing?**

* 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 to test the system’s behavior under normal workload conditions and it is just testing or simulating with the actual workload.
* This testing usually identifies –
  + The maximum operating capacity of an application.
  + Determine whether current infrastructure is sufficient to run the application.
  + Number of concurrent users that an application can support.
  + It is a type of non-functional testing.
  + It is commonly used for the Client/Server Web based applications – both Intranet and Internet.
* Load Testing is helpful for following reasons:
  + 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.
* **Goals** of Load Testing:
  + Response time for each transaction
  + Performance of System components under various loads
  + Performance of Database components under different loads
  + Network delay between the client and the server
  + Software design issues
  + Server configuration issues like Web server, application server, database server etc.
  + Hardware limitation issues like CPU maximization, memory limitations, network bottleneck, etc.

1. **What is Stress Testing?**

* Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant.
* System is stressed beyond its specifications to check how and when it fails.
* It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.
* Stress Testing is done to make sure that the system would not crash under crunch situations.
* Stress testing is also known as endurance testing.
* Most prominent use of stress testing is to determine the limit at which the system or software or hardware breaks.
* **Need** of Stress Testing
  + During festival time, an online shopping site may witness a spike in traffic or when it announces a sale.
  + Displaying appropriate error message when the system is under stress.
  + System failure under extreme conditions could result in enormous revenue loss
* **Goal** of Stress Testing
  + The goal of stress testing is to analyze the behavior of the system after failure.
  + The main purpose of stress testing is to make sure that the system recovers after failure which is called as **recoverability**.
* **Types** Stress Testing:
  + Application Stress Testing
  + Transactional Stress Testing
  + Systemic Stress Testing
  + Exploratory Stress Testing

1. **What is White Box Testing and list the types of white box testing?**

* Testing based on an analysis of the internal structure of the component or system.
* Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ or ‘open box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.
* White box testing is the detailed investigation of internal logic and structure of the code.
* The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.
* **Types** of Black Box Testing:
  + WEB BASED TESTING
  + Analyze the logic by reading the code
  + Code optimization Suggest if any optimization can be done in the code which is better than the existing one
  + DESKTOP BASED TESTING
  + When we debug the code when we writing
  + MOBILE BASED TESTING
  + Android devices enabled for development and debugging, with appropriate USB drivers as necessary
  + GAME BASED TESTING
  + When we connect with remote device , so which device we connect will check in code

1. **What is Black Box Testing and what are the different black box testing techniques?**

* The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
* 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.
* Specification-based techniques are appropriate at all levels of testing where a specification exists.
* The tester is oblivious to the system architecture and does not have access to the source code.
* Typically, when performing a black box test, 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.
* Types of Black Box Testing:
  + WEB BASED TESTING
  + Takes more time to execute as testers look for game play issues, graphic issues, audio-visual issues, etc.
  + DESKTOP BASED TESTING
  + Installation Testing (Upgrade/Downgrade)
  + MOBILE BASED TESTING
  + In mobile, it will automatically switch off without any reason.
  + GAME BASED TESTING
  + The game tester must know how to play the game, utilization of the gamepad, know the game flow and the rules.
* Techniques of Black Box Testing:
  + Equivalence Partitioning
  + Boundary Value Analysis
  + Decision Tables
  + State Transition Testing
  + Use-Case Testing

1. **Mention what are the categories of defects?**

* Defect is the variance from a desired product attribute (it can be a wrong, missing or extra data).
* **Types**:
* **Data Quality/Database Defects:** Deals with improper handling of data in the database.
  + Examples:
  + Values not deleted/inserted into the database properly
  + Improper/wrong/null values inserted in place of the actual values
* **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application.
  + Examples: - Exceptions
* **Functionality Defects:** These defects affect the functionality of the application.
  + Examples:
  + All JavaScript errors
  + Buttons like Save, Delete, Cancel not performing their intended functions
  + A missing functionality (or) a feature not functioning the way it is intended to
  + Continuous execution of loops
* **Security Defects:** Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
  + Examples:
  + Authentication: Accepting an invalid username/password
  + Authorization: Accessibility to pages though permission not given
* **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.
  + Examples:
  + Improper error/warning/UI messages
  + Spelling mistakes
  + Alignment problems

1. **Mention what Big Bang testing is?**

* In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
* Here all component are integrated together at once and then tested.
* 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.
* **Advantages:**
  + Convenient for small systems.
* **Disadvantages:**
  + Since the integration testing can commence only after “all” the modules are designed, testing team will have less time for execution in the testing phase.

1. **What is the purpose of exit criteria?**

* Purpose of exit criteria is to define when to STOP testing either at the:
  + End of all testing – i.e. product Go Live
  + End of phase of testing (e.g. hand over from System Test to UAT)

1. **When should “Regression Testing” be performed?**

* Regression testing should be carried out:
  + when the system is stable and the system or the environment changes
  + when testing bug-fix releases as part of the maintenance phase

1. **What are 7 key principles? Explain in detail.**

* 7 key principles are;
  + Testing shows presence of Defects
  + Exhaustive Testing is Impossible
  + Early Testing
  + Defect Clustering
  + The Pesticide Paradox
  + Testing is Context Dependent
  + Absence of Errors Fallacy
* Testing shows presence of Defects
  + Testing can show that defects are present, but cannot prove that there are no defects.
  + 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
* Exhaustive Testing is Impossible
  + Testing everything including all combinations of inputs and preconditions is not possible.
* Early Testing
  + Testing activities should start as early as possible in the development life cycle
* Defect Clustering
  + Defects are not evenly spread in a system
  + They are ‘clustered’
  + In other words, most defects found during testing are usually confined to a small number of modules
* The 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.
  + 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
* Testing is Context Dependent
  + Testing is basically context dependent
  + Testing is done differently in different contexts
  + 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
* Absence of Errors Fallacy
  + Even after defects have been resolved, it may still be unusable and/or does not fulfill the users’ needs and expectations

1. **Difference between QA and QC and Tester.**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **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.**

|  |  |
| --- | --- |
| **SMOKE TESTING** | **SANITY TESTING** |
| Checks the critical functionality | Checks the new functionality |
| It is done in initial stage | It is done after 30 builds |
| It checks the stability | It checks the Sanity/Rationality |
| Part of acceptance testing | Part of regression testing |
| General health checkup | Advance health checkup |
| Done by tester and developer | Done by tester |
| It checks the system end to end | It checks only a particular function of entire system |

1. **Difference between Verification and Validation.**

|  |  |
| --- | --- |
| **VERIFICATION** | **VALIDATION** |
| The process of evaluating work-products of a development phase to determine whether they meet the specified requirements for 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 is to ensure that the product is being built according to the requirements and design specifications | Objective is to ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place |
| By this we check that whether we are building the product right or not | By this we check that whether we are building the right product or not |
| Evaluation items are Plans, Requirement Specs, Design  Specs, Code, Test Cases | Evaluation item is the actual product/software. |
| Activities are Reviews, Walkthroughs, Inspections | Activity is Testing |

1. **Explain types of Performance Testing.**

* Types of Performance Testing;
  + Load testing
  + Stress testing
  + Endurance testing
  + Spike testing
  + Volume testing
  + Scalability testing
  + LOAD TESTING: Load Testing is to test the system’s behavior under normal workload conditions and it is just testing or simulating with the actual workload.
  + STRESS TESTING (ENDURANCE TESTING): Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant.
  + SPIKE TESTING: Spike testing is to test the system’s behavior when there is spike in traffic.

E.g. - During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.

* + VOLUME TESTING: Volume testing is to test the system’s behavior when there is excessive volume in terms of either users or data.
  + SCALABILITY TESTING: Determines maximum user load the software application can handle.

1. **What are Error, Defect, Bug and Failure?**

* 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: Commonly refers to several troubles with the software products, with its external behavior or with its internal features.
* Bug:A fault in a program which causes the program to perform in an unintended or unanticipated manner.
* Failure: The inability of a system or component to perform its required functions within specified performance requirements.

1. **Difference between Priority and Severity.**

* **Priority:** Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. The 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.
  + For example: 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.
* Priority can be of following types:
  + Low: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.
  + Medium: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
  + High: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.
  + Critical: Extremely urgent, resolve immediately
* **Severity:** 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.
* For example: 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.
* Severity can be of following types:
  + Critical: The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.
  + Major (High): The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.
  + Moderate (Medium): The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.
  + Minor (Low): The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.
  + Cosmetic: The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

1. **What is Bug Life Cycle?**

* 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 ‘Bug Life Cycle’.

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

|  |  |
| --- | --- |
| **Functional Testing** | **Non Functional Testing** |
| Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements | Non-Functional 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  ∙ Regression Testing | Types of Non functional testing are:  ∙ Performance Testing  ∙ Load Testing  ∙ Stress Testing  ∙ Security Testing  ∙ Installation Testing  ∙ Penetration Testing  ∙ Compatibility Testing |

1. ***Create HLR and Test Case of Instagram and Facebook (first page only).***
2. **What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

|  |  |
| --- | --- |
| **STLC** | **SDLC** |
| STLC is mainly related to software testing. | SDLC is mainly related to software development. |
| It focuses only on testing the software. | Besides development other phases like testing is also included. |
| STLC involves only five phases or steps. | SDLC involves total six phases or steps. |
| In STLC, less numbers of members (testers) are required. | In SDLC, more number of members (developers) are required. |
| Goal of STLC is to complete successful testing of software. | Goal of SDLC is to complete successful development of software. |
| It helps in making the software defects free | It helps in developing good quality software. |
| STLC phases are performed after SDLC phases | SDLC phases are completed before the STLC phases. |

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

* **Test Scenario:**
  + A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.
  + Test Scenario is ‘What to be tested’
  + Test scenario is nothing but test procedure.
  + The scenarios are derived from use cases.
  + Test Scenario represents a series of actions that are associated together.
* **Test Case:**
  + Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.
  + Test Case is ‘How to be tested’
  + Test cases are set of input and output given to the System.
  + Test cases are derived (or written) from test scenario.
  + Test Case represents a single (low level) action by the user.
* **Test Script:**
  + A set of sequential instruction that detail how to execute a core business function
  + Script is written to a level of detail for which someone else would be able to easily execute
  + Identifies the test condition that is being satisfied for each step, if applicable
  + Identifies the test condition that is being satisfied for each step, if applicable
  + Identifies the input/test data that should be entered for each transaction

1. **Explain Test Plan? What information should be covered?**

* A document describing the scope, approach, resources and schedule of intended test activities
* All projects require a set of plans and strategies which define how the testing will be conducted.
* It covers information related to making decisions about:
  + What to test?
  + Who do testing? i.e. what roles will perform the test activities
  + When and how the test activities should be done and when they should be stopped (exit criteria)
  + How the test results will be evaluated
  + Assigning resources for the different tasks defined.

1. **What is priority?**

* **Priority:** Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. The 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?**

* **Severity:** 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…?**

* Bugs are categorized as below:
  + Security
  + Database
  + Functionality (Critical/General)
  + UI

1. **Advantage of Bugzilla?**
   * Open source, free bug tracking tool.
   * Automatic [Duplicate Bug Detection](https://cloudinfrastructureservices.co.uk/how-to-setup-bugzilla-issue-tracker-on-azure-aws-gcp/).
   * Search option with advanced features.
   * File/Modify Bugs By Email.
   * Move Bugs between Installs.
   * Multiple [Authentication](https://cloudinfrastructureservices.co.uk/adfs-vs-azure-ad-how-authentication-has-evolved/) Methods ([LDAP](https://cloudinfrastructureservices.co.uk/radius-vs-ldap-vs-kerberos/),[Apache server](https://cloudinfrastructureservices.co.uk/how-to-setup-apache-web-server-mysql-server-on-linux-in-azure-aws-gcp/)).
   * Time Tracking.
   * Automated bug reporting; has an API to interact with system.
   * Integrated email capabilities.
   * Detailed permissions system.
   * Optimized database structure to enhance performance.
   * Robust security.
   * Powerful query tool.
   * Ideal for small projects.
2. **What are the different Methodologies in Agile Development Model?**

* Agile Development Model methodologies are:
  + Waterfall Model/Methodology (Classical Software Cycle)
  + Iterative & Incremental Model/Methodology
  + Spiral Model/Methodology
  + Agile Model/Methodology
  + Use Case

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

* Authorization and Authentication are the types of defects
* Authorization is a type of defect which accepts an invalid username or password, whereas Authentication is a type of defect which allows access to pages though permission is not given
* Common problems faced in web testing are;
  + Integration
  + Interoperability
  + Security
  + Usability
  + Performance

1. ***Create HLR and Test Case of Web Based WhatsappWeb and Instagram Web.***
2. ***Create HLR and Test Case of the LINK.***
3. ***Write a scenario of only Whatsapp chat messages.***
4. ***Write a Scenario of Pen.***
5. ***Write a Scenario of Pen Stand.***
6. ***Write a Scenario of Door.***
7. ***Write a Scenario of ATM.***
8. **When to use Usability Testing?**

* Usability testing should be conducted before putting any design resources to work.
* This will quickly identify areas for opportunity and reduce the amount of assumptions your design team will make with regard to what the user wants.
* Additionally, after the usability tests analysis, the team should have the ability to pinpoint the steps needed to achieve the project goals with as little disruption as possible.

1. **What is the procedure for 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.

1. ***Write a scenario of Microwave Owen.***
2. ***Write a scenario of Coffee Vending Machine.***
3. ***Write a scenario of Chair.***
4. ***Create scenario (positive & negative) Facebook chat on mobile, Gmail (Receiving mail), online shopping to buy product (flipkart).***
5. ***Write a Scenario of Wrist Watch***
6. ***Write a Scenario of Lift (Elevator).***
7. ***Write a Scenario of Whatsapp Group (generate group).***
8. ***Write a Scenario of Instagram (video call with chat).***
9. ***Write a Scenario of Whatsapp Payment.***