**Assignment – 2**

1. **What is Exploratory testing?**

Exploratory testing is a concurrent process where,

* the test design, execution and logging happen simultaneously.
* Makes use of experience, heuristics and test patterns
* Testing is based on a test charter that may include
* Scope of the testing
* A brief description of how tests will be performed
* Expected problems

Though the current trends 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.

In Exploratory testing functionalities are checked in a ad-hoc testing.

1. **What is traceability matrix?**

* Traceability matrix (also known as Requirement Traceability Matrix-RTM) is a table which is used to trace the the requirements during the Software development life cycle. It can be used for forward tracing(i.e from Requirements to design or coding) or backward(i.e from coding to requirements). There are many user defined templates for RTM.
* A requirements traceability matrix is a document that traces and maps user requirements(requirements Ids from requirement specification document)with the test cases Ids. Purpose is to make sure that all the requirements are covered in test cases so that while testing no functionality can be missed.
* **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 Referenced from test cases to basis documentation and vice versa.

**Advantages:-**

1)To make sure that all requirements included in the test cases

2)To make sure that developers are not creating features that no one has requested

3)Easy to identify the missing functionalities.

4)If there is a change request for a requirements, then we can easily find out which test cases need to update.

**Disadvantages:-**

* 1. No traceability or incomplete traceability results are:
  2. Poor or unknown test coverage, more defects found in production
  3. Difficult project planning and tracking, misunderstanding between different teams over project dependencies, delays, etc.

**3)What is boundary value testing?**

* 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.
* Boundary value analysis(BVA)uses the same analysis of partitions as EP and is usually used in conjunction with EP in test cases design.
* BVA Test cases are designed to exercise the software on and at either side of boundary values.
* Find the boundary and then test one value above and below it.
* Always results in two test cases per boundary for valid inputs and three tests cases per boundary for all inputs.
* E.g- Boundary of ‘a>10.0’ should results in test values of 10.0, 10.1 & 10.2) only applicable for numeric (and date) fields

**4)What is Equivalence partitioning testing?**

* Equivalence Partitioning can be used for all levels of testing
* If we want to test the following IF statement: If values is between 1 and 100(inclusive) (e.g value>=1 and value<=100)
* The number fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition(EP) or Equivalence Class
* In EP we must identify Valid Equivalence partitions and Invalid Equivalence partitions where applicable (typically in range tests)
* The valid partitions is bounded by the values 1 and 100
* Pass there are 2 invalid partitions
* EP can help reduces the number of tests from a list of all possible inputs to a minimum set that would still test each partition
* EP is used to achieve good input and output coverage, knowing exhaustive testing is often impossible
* It can be applied to human input, input via interface to a system, or interface parameters in integration testing

**5)What is Integration Testing?**

* Integration Testing- Testing performed to expose defects in the interfaces and in the interactions between integrated components or system.
* 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.
* There are two levels of Integration Testing

1)Component Integration Testing

2)System Integration Testing

**1)Component Integration Testing:-**

* Testing performed to expose defects in the interfaces and interactions between integrated components.
* All individual components should be integration tested prior to system testing.
* If tests the interactions between software components and is done after component testing.
* The following testing techniques are appropriate for Integration Testing.

1. Functional Testing using Black Box testing techniques against the interfacing requirements for the components under test.
2. Non-Functional Testing.

**2)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.
* As testing for dependencies between different components is a primary function of SIT Testing, this area is often most subject to Regression Testing.

There is two types methods of Integration Testing:

1)Big Bang Integration Testing

2)Incremental Integration testing

Top Down Approach

Bottom Up Approach

Integration Testing is performed after Unit testing and before system testing.

Either Developers themselves or independent testers perform Integration Testing.

**1)Big Bang Integration Testing:-**

* 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 .

**2)Incremental Integration Testing:-**

* The Incremental approach has the advantage that the defects are found early in a smaller assembly when it is relatively easy to detect the cause.
* A disadvantage is that it can be time-consuming since stubs and drivers have to be developed and used in the test.
* Stubs and drivers do not implement the entire programming logic of the software module but just simulate data communication with the calling module.

**Stubs**: Is called by the Module under test.

**Driver**: calls the module to be tested.

**Top Down Approach:**

Testing takes place from top to bottom, following the control flow or architectural structure.

It takes help of stubs for testing.

Advantages: 1)Fault localization is easier.

2)Possibility to obtain an early prototype.

Disadvantage: 1)Needs many stubs.

2)Modules at lower level are tested inadequately.

**Bottom Up Approach:**

* In the bottom up strategy, each module at lower levels is tested with higher modules until all modules are tested. It takes help of Drivers for testing.

Advantages: 1)Fault localization is easier.

2)No time is wasted waiting for all modules to be developed unlike Big-Bang approach.

Disadvantage: Early prototype is not possible.

**6)What determines the level of risk?**

* A properly designed test that passes, reduces the overall level of Risk is a system.
* Risk- A factor that could result in future negative consequences; usually expressed as impact and likelihood.
* The quality of system can be improved through Lesson learned from previous projects.

Types of Risk:- Risk are of two types

1)Project Risks

2)Product Risks

**Project Risks:-** Project risks is Senior Team Member leaving the project abruptly.

Eg :- Every risk is assigned a likelihood i.e. chance of it occurring, typically on a scale of 1 to 10. Also the impact of that risk is identified on a scale 1-10.

**Product Risks:-** Product risks would be Flight Reservation system not installing in test environment.

**7) What is Alpha Testing ?**

* It is always performed by the developers at the software development site.
* 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.
* It comes under the category of both white box testing and black box testing .
* Alpha testing is always performed at the time of Acceptance Testing when developers test the product and project to check whether it meets the user requirements or not.
* Unit testing, integration testing and system testing when combined are known as alpha testing.

**8) What is Beta Testing ?**

* It is always performed by the customers at their own site.
* 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 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.
* Beta testing can be considered “pre-release” testing.

**9) What is component Testing ?**

* A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”
* The testing of individual software components.
* 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.
* The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.
* Unit testing is performed by using the White Box Testing method.
* 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.

**10)What is 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

**1)Requirement Based functional Testing-**

* Testing against requirements and specifications
* Starts by using the most appropriate black-box testing techniques
* May support this with white-box techniques (e.g. menu structures, web page navigation)

**2)Business process based testing-**

* Test procedures and cases derived from:

1)Expected user profiles

2)Business scenarios

3)Use cases

* Testing should reflect the business environment and processes in which the system will operate.
* Therefore, test cases should be based on real business processes.

**11)What is non-functional testing?**

* Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability
* It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.
* To address this issue, performance testing is carried out to check & fine tune system response times.
* Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

**Examples:-**

**Web Based Testing :**

* Identify the software processes that directly influence the overall performance of the system.
* In website number of user/customer will increase , how the website will handled to every customer/user.

**Mobile Based Testing :**

* In mobile , automatically will switch off without any reason.
* To stop the application which is not in our hand.

**Game Based Testing :**

* Confirms workability and stability of the software.
* Validate whether the user interface of the app is as per the screen size of the device and ensure high quality

**12)What is GUI Testing?**

* 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 the alignment of the text is proper
* Check that the images have good clarity
* Check that the images are properly aligned

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 - 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.

**Examples:**

**Web Based Testing & Desktop Based Testing :**

* The scrollbar should be enabled only when necessary.
* The description text box should be multi-lined.

**Mobile Based Testing :**

* If mobile is in every orientation mode so display image , video properly.
* Every app will display in responsive type .

**13)What is Adhoc Testing?**

* Adhoc testing is an informal testing type with an aim to break the system.
* 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.
* This is why an error guessing approach, used after more formal techniques have been applied to some extent, can be very effective.
* Using experience to postulate errors.
* Use Error Guessing to Complement Test Design Techniques.

**Types of Adhoc Testing:**

**1)Buddy Testing:-** Two buddies mutually work on identifying defects in the same module. Buddy testing helps the testers develop better test cases and development team can also make design changes early.

**2)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.

**3)Monkey Testing:-** Randomly test the product or application without test cases with a goal to break the system.

**14)What is load testing?**

* Load testing - 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 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 simultaneously.

**Pros and Cons of Load Testing:-**

Pros:

* Performance bottlenecks identification before production
* Improves the scalability of the system

Cons:

* Need programming knowledge to use load testing tools.
* Tools can be expensive as pricing depends on the number of virtual users supported.

Need For Load Testing:

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.

**Goals of Load Testing:** Loading testing identifies the following problems before moving the application to market or Production:

* Response time for each transaction
* Performance of System components under various loads
* Performance of Database components under different loads

**15) what is stress testing?**

* Stress testing - System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
* 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 For Stress Testing:-**

1. During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.
2. To check whether the system works under abnormal conditions.
3. Displaying appropriate error message when the system is under stress.

**Goal of Stress Testing:-**

* 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.
* The main purpose of stress testing is to make sure that the system recovers after failure which is called as recoverability.

**Types of Stress Testing:-**

1. Application Stress Testing:
2. Transactional Stress Testing:
3. Systemic Stress Testing:
4. Exploratory Stress Testing:

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

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’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

**White box testing techniques/Types of coverage:-**

1. Statement coverage
2. Decision coverage
3. Condition coverage

**1)Statement/Segment Coverage:-**

* The statement coverage is also known as line coverage or segment coverage.
* The statement coverage covers only the true conditions.
* The statement coverage can be calculated as shown below:

number of statements exercised

statement coverage= —---------------------------------------------------- x 100

Total number of statements

**2)Decision/Branch Coverage:-**

* Decision coverage also known as branch coverage or all-edges coverage.
* It covers both the true and false conditions unlikely the statement coverage.
* Aim is to demonstrate that all Decisions have been run at least once
* The decision coverage can be calculated as shown below:

number of decision outcomes exercised

Decision coverage = —------------------------------------------------------------------- x 100

Total number of decision outcomes

**3)Condition Coverage:-**

* This is closely related to decision coverage but has better sensitivity to the control flow.
* full condition coverage does not guarantee full decision coverage.
* Condition coverage reports the true or false outcome of each condition.
* Condition coverage measures the conditions independently of each other.

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

* Black-box testing: 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 technique of testing without having any knowledge of the interior workings of the application is Black Box testing.

**Techniques of Black Box Testing:-**

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing

**1)Equivalence Partitioning(E.P.):-**

* Aim is to treat groups of inputs as equivalent and to select one representative input to test them all
* 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...”
* The numbers fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition (EP) or Equivalence Class
* In EP we must identify Valid Equivalence partitions and Invalid Equivalence partitions where applicable (typically in range tests)

**2)Boundary Value Analysis(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
* Boundary value analysis is a method which refines equivalence partitioning.
* Boundary value analysis 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

**3)Decision Table:-**

* The other two specification-based software testing techniques, decision tables and state transition testing are more focused on business logic or business rules.
* A decision table is a good way to deal with combinations of things (e.g. inputs).
* Using the 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)

**18)Mention what are the categories of defects?**

**Types of Defect:-**

**1)Data Quality/Database Defects:**Deals with improper handling of data in the database.

Examples: Improper/wrong/null values inserted in place of the actual values

**2)Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application.

Examples: - Exceptions

**3)Functionality Defects:** These defects affect the functionality of the application.

Examples: All JavaScript errors

**4)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

**5)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

**19)Mention what bigbang testing is?**

* 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.

**Advantages:**

* Convenient for small systems.

**Disadvantages:**

* Fault Localization is difficult.
* Given the sheer number of interfaces that need to be tested in this approach, some interfaces links to be tested could be missed easily.

**20)What is the purpose of exit criteria?**

How do we know when to stop testing?

1. Run out of time?
2. Run out of budget?
3. The business tells you it went live last night!
4. Boss says stop?
5. All defects have been fixed?
6. When out exit criteria have been met?

Purpose of exit criteria is to define when we STOP testing either at the:

1. End of all testing – i.e. product Go Live
2. End of phase of testing (e.g. hand over from System Test to UAT)

Exit Criteria typically measures:

1. Thoroughness measures, such as coverage of requirements or of code or risk
2. coverage
3. Estimates of defect density or reliability measures. (e.g. how many defects open by
4. category)
5. Cost.
6. Residual Risks, such as defects not fixed or lack of test coverage in certain areas.
7. Schedules - such as those based on time to market.

**21)When should “Regression Testing” be performed?**

* Regression Testing: 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.
* You also need to ensure that the modifications have not caused unintended side-effects elsewhere and that the modified system still meets its requirements – Regression Testing

**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

**22)What is 7 key principles? Explain in deatil?**

**7 Key Principle**

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 Faults
* As we find more defects, the probability of undiscovered defects remaining in a system reduces.

**2.Exhaustive Testing is Impossible!:-**

* 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.

Why do not Testing Everything?

* Exhaustive testing of complex software applications:
* requires enormous resources
* is too expensive
* takes too long
* It is therefore impractical

**3.Early Testing:-**

* Testing activities should start as early as possible in the development life cycle
* These activities should be focused on defined objectives – outlined in the Test Strategy
* Remember from our Definition of Testing, that Testing doesn’t start once the code has been written!

**4.Defect Clustering:-**

* 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’
* An important consideration in test prioritisation!

**5.Pesticide Paradox:-**

* 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.
* As bugs are eliminated by the programmers, the software improves
* Therefore we must learn, create and use new tests based on new techniques to catch new bugs

**6.Testing is Context Dependent:-**

* Testing is basically context dependent.
* Different kinds of sites are tested differently.
* For example

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

**7.Absence of Errors Fallacy:-**

* If the system built is unusable and does not fulfill the user’s needs and expectations then finding and fixing defects does not help.
* Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectations

**23)Difference between QA v/s QC v/v Tester**

| **Sr No.** | **Quality Assurance** | **Quality Control** | **Testing** |
| --- | --- | --- | --- |
| 1 | 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. |
| 2 | Process oriented activities. | Product oriented activities. | Product oriented  activities. |
| 3 | Preventive activities. | It is a corrective process. | It is a preventive process. |
| 4 | 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. |
| 5 | 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. |

**24)Difference between smoke and sanity?**

| **Smoke 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 |
| 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 |

**25)Difference between verification and validation?**

| **Sr No.** | **Verification** | **Validation** |
| --- | --- | --- |
| 1 | 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. |
| 2 | Are we building the product right? | Are we building the right product? |
| 3 | Plans, Requirement Specs, Design  Specs, Code, Test Cases | The actual product/software. |
| 4 | Reviews  ∙ Walkthroughs  ∙ Inspections | Testing |

**26)Explain types of performance Testing?**

**Types of Performance Testing:-**

1)Load testing

2)Stress testing

**1)stress Testing:-**

* System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
* 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.

**Need For Stress Testing:-**

* During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.
* When a blog is mentioned in a leading newspaper, it experiences a sudden surge in traffic.

**Goal of Stress Testing:-**

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:

**2)Load Testing:-**

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.

**Need For Load Testing:-**

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.

**Goals of Load Testing:-**

Loading testing identifies the following problems before moving the application to market or Production:

* Response time for each transaction
* Performance of System components under various loads
* Performance of Database components under different loads

**Pros and Cons of Load Testing:-**

**pros:-**

1. Performance bottlenecks identification before production
2. Improves the scalability of the system
3. Minimize risk related to system down time

**cons:-**

1. Need programming knowledge to use load testing tools.
2. Tools can be expensive as pricing depends on the number of virtual users supported.

**27)What is Error, Defect, Bug and failure?**

“A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is 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.

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

**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.

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

**28)Difference between priority and severity?**

| **Sr No.** | **Severity** | **Priority** |
| --- | --- | --- |
| 1 | severity is associated with functionality | priority is associated with scheduling |
| 2 | It indicate the seriousness of defect | It indicate how soon the bug should be fixed |
| 3 | QA engineer determine the severity level | priority of defect id consultation with the client |
| 4 | severity is driven by functionality | priority is driven by business level |
| 5 | severity levels are: critical, major ,minor, moderate and cosmetic | priority levels are : critical , high, medium, low |

**29)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 ‘Defect Life Cycle’.
* When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and closed.
* When a bug reaches one of the Closed or Terminal states, its lifecycle ends. Each state has one or more valid states to move to.

**Defect Stages:-**

**1)New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.

**2)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

**3)Open:** The developer starts analyzing and works on the defect fix

**4)Fixed:** When a developer makes a necessary code change and verifies the change ,he or she can make bug status as “Fixed.”

**5)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.”

**6)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.”

**7)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.

**8)Closed:** If the bug is no longer exists then tester assigns the status “Closed.”

**9)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

**30)Explain the Difference between functional and non-functional testing?**

| Sr No. | Functional testing | Non-functional testing |
| --- | --- | --- |
| 1 | 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. |
| 2 | Functional testing is executed first | Non functional testing should be performed after functional testing |
| 3 | Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| 4 | Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. |
| 5 | Functional testing describes what the product does | Nonfunctional testing describes how good the product works |
| 6 | Easy to do manual testing | Tough to do manual testing |
| 7 | Types of Functional testing are  Unit Testing  ∙ Smoke Testing  ∙ Sanity Testing  ∙ Integration Testing | Types of Nonfunctional testing are  ∙ Performance Testing  ∙ Load Testing  ∙ Volume Testing  ∙ Stress Testing |

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

| **Sr No.** | **SDLC** | **STLC** |
| --- | --- | --- |
| 1 | Focuses on building a product | Focuses on testing a product |
| 2 | A parent process | A child of SDLC process |
| 3 | Building a product as user requirement | Ensuring the product is working as expected |
| 4 | SDLC phases are completed before testing | STLC phases start after SDLC phases are completed |
| 5 | End goal is to deploy a high quality product to user | End goal is to finding and fixing the bugs/defects |

**32)What is priority?**

* 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.

**Priority can be of following types:**

**1)Low:** The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.

**2)Medium:** The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

**3)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.

**4) Critical**: Extremely urgent, resolve immediately

**33)What is severity?**

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.

**Severity can be of following types:**

**1)Critical:** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data.

**2)Major (High):** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data.

**3)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.

**4)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.

**5)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.

**34)Bug Categories are….**

Bug Category: Security, Database, Functionality (Critical/General), UI

**35)What are the different Methodologies in Agile Development Model?**

* 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.
* Agile is a philosophy, i.e., a set of values and principles to make a decision for developing software.

**36)Explain what Test plan is? What is the information that should be covered.**

* All projects require a set of plans and strategies which define how the testing will be conducted.
* A document describing the scope, approach, resources and schedule of intended test activities
* Defining the amount, level of detail, structure and templates for the test documentation.
* There are number of levels at which these are defined:

Test policy:- Defines how the organisation will conduct testing

Master Test plan:- Defines how the project will conduct testing

Functional Test plan, System Integration Test Plan, UAT test plan:- Defines how each level of testing will be conducted

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

| **Test script** | **Test scenario** | **Test cases** |
| --- | --- | --- |
| 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. | 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. |
| There are various means for executing test scripts.  Manual Testing  Automation Testing | The scenarios are derived from use cases. | Test cases are derived (or written) from test scenario. |

**38)Advantage of bugzilla.**

1. Key features of Bugzilla includes
2. Advanced search capabilities
3. E-mail Notifications
4. Modify/file Bugs by e-mail
5. Time tracking
6. Strong security
7. Customization
8. Localization
9. Bugzilla is a defect tracking tool
10. It is written in Perl and uses MYSQL database.