TOPS TECHNOLOGIES

**Software testing assignment**

**Module-2(Manual testing)**

**1 What is Exploratory testing ?**

= Exploratory testing is an approach to software testing that is often described as simultaneous learning, testing design and execution. In focuses on discover and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other tests. Exploratory testing is mostly used if the requirements are incomplete and time to release the software is less.

**2 What is traceability matrix ?**

= Traceability matrix is a table which is used to trace the requirements during the software development life cycle. It can be used for forward tracing or backward. A traceability matrix is a document that details the technical requirements for a given test scenario and its current state. It helps the testing team understand the level of testing that is done for a given product. Using this document the person can track the requirement based on the defect id.

* **Types of traceability matrices:**

**1 Forward traceability:** Mapping of requirements to test cases.

**2 Backward traceability:** Mapping of test cases to requirements.

**3 Bi-Directional traceability:** A good example of a bi-directional traceability matrix used in software testing is the references from test cases to basis documentation and vice versa.

**3 What is boundary value testing ?**

= software testing technique in which tests are designed to include representatives of boundary values. It is performed by the QA testing teams. Boundary value analysis is black-box testing technique used to check the errors at the boundaries of an input domain. The name comes from the boundary, which means the limits of an area.

**4 What is Equivalence partitioning testing ?**

= Equivalence partitioning testing is software testing technique that divides the input data of a software unit into partitions of data from which test cases can be derived.it is usually performed by the QA teams. Equivalence partitioning method is also known as equivalence class partitioning. Values from each partition must be tested at least once. Partitions with valid values are used for positive testing. While, partitions with invalid values are used for negative testing.

**5 What is integration testing ?**

= Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system. Integration testing is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity. Integration testing is a level of the software testing process where individual units are combined and tested as a group. Integration testing is done by 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**

**1 Component integration testing**

**2 System integration testing**

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

= Risk is determined by a combination of probability and severity; the main area of the ,atrix reveals the risk levels. The levels are low, medium, high. To have a low level of risk, we must have a somewhat limited probability and level of severity. Risk should be evaluated at the business level, technological level, project level and testing level.

**7 What is alpha testing ?**

= Alpha testing is performed by the testers within the organization .

= Alpha testing is performed in virtual environment .

= Alpha testing is performed by independent testing team.

= Alpha testing is conducted for the software application and project .

= Alpha testing is performed at developer’s site.

= Alpha testing is the from of acceptance testing .

= Alpha testing involves both white box and black box testing.

= Alpha testing identifies possible errors.

= Alpha testing requires testing environment.

= Alpha testing requires long execution cycle.

= Alpha testing is to ensure the quality of the product before moving to beta testing.

**8 What is beta testing?**

= Beta testing is performed by the end users.

= Beta testing is performed in real time environment .

= Beta testing is not performed by independent testing team .

= Beta testing is usually conducted for software product .

= Beta testing is performed at client’s location .

= Beta testing is also the form of acceptance testing .

= Beta testing mainly involves black box testing .

= Beta testing checks the quality of the product .

= Beta testing doesn’t require any lab environment ir testing environment.

= Beta testing requites only few weeks of execution.

**9 What is component testing ?**

= Component testing is also known as unit testing.

= Component testing is the testing of individual software components.

= Component is a minimal software item that can be tested is isolation. It means “A unit is the smallest testable part of software.”

= Unit testing is the first level of testing and is performed prior to integration testing.

= Unit testing is performed by using the white box testing method.

= Entry criteria for component testing.

* **Exit criteria for component testing:**

1. The functionality of all the components must be working as expected;
2. There should be no critical , high , medium severity and priority defect/bug left the software.

**10 What is functional system testing ?**

= Functional testing is a type of software testing in which the system is tested against the functional requirements and specifications. There is two types of techniques. Requirement based functional testing. Process based testing

* **Types of functional system testing:**

**1 Unit testing**

**2 Smoke testing**

**3 Sanity testing**

**4 Integration testing**

**5 White box testing**

**6 Black box testing**

**7 User acceptance testing**

**8 Regression testing**

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

= Non-functional testing assesses application properties that aren’t critical to functionality but contribute to the end-user experience. Non-functional testing has a goal to validate the performance of the software.

* **Types of Nonfunctional testing are:**

**1. Performance Testing**

**2. Load Testing**

**3. Volume Testing**

**4. Stress Testing**

**5. Security Testing**

**6. Installation Testing**

**7. Penetration Testing**

**8. Compatibility Testing**

**9. Migration Testing**

**12. What is GUI Testing?**

= GUI testing is a testing technique in which the application's user interface is tested.Whether the application performs as expected with respect to user interface behaviour.

* **GUI Testing Guidelines:**

**1. Check Screen Validations**

**2. Verify All Navigations**

**3. Check usability Conditions**

**4. Verify Data Integrity**

**5. Verify the object states**

**6. Verify the date Field and Numeric Field Formats**

* **Approach of GUI Testing:**

**1. MANUAL BASED TESTING**

**2. RECORD AND REPLAY**

**3. MODEL BASED TESTING**

**13. What is Adhoc testing?**

= Adhoc Testing is also known as Error Guessing.Adhoc testing is an informal testing type with an aim to break the system.When a software testing performed without proper planning and documentation,

it is said to be Adhoc Testing.It does not follow any test design techniques to create test cases.Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of application.Error guessing can be done by the people having enough experience on the system to.“guess” the most likely source of errors.

* **There are different types of Adhoc testing and they are listed as below:**

**1. Buddy Testing**

**2. Pair testing**

**3. Monkey Testing**

**14. What is load testing?**

= Load testing helps to determine the behavior of an application when large numbers of users access the application simultaneously.It is a type of performance testing which identifies the maximum operating capacity of an application.This helps to know whether the existing infrastructure is capable to run the application or not.Load testing is conducted to determine the number of concurrent users that an application can support without deterioration in performance.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.Load Testing helps identify the bottlenecks in the system under heavy user stress

scenarios before they happen in a production environment.

**15. What is stress Testing?**

= Stress Testing is a software testing technique that determines the robustness of software by testing beyond the limits of normal operation.Stress testing is particularly important for critical software but is used for all types of software.Stress testing emphasizes robustness, availability, and error handling under a heavy load rather than what is correct behavior under normal situations.It also checks whether system demonstrates effective error management under extreme conditions.

**Example:**

**The application under testing will be stressed when 5GB data is copied from the website and pasted in notepad.**

**Notepad is under stress and gives ‘Not Responded’ error message.**

* **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 is a testing technique, that examines the program structure and derives test data from the program logic/code.It is also called as glass box testing, clear box testing, open box testing, logic

driven testing or path driven testing or structural testing.

* Advantages of White Box Testing:

Code optimization by finding hidden errors.

White box tests cases can be easily automated.

Testing is more thorough as all code paths are usually covered.

Testing can start early in SDLC even if GUI is not available.

* Disadvantages of White Box Testing:

White box testing can be quite complex and expensive.

Developers who usually execute white box test cases detest it. The white box testing by developers is not detailed can lead to production errors.

White box testing requires professional resources, with a detailed

understanding of programming and implementation.

* Types of code coverage:

Statement Coverage

Decision Coverage

Condition Coverage

* Types of other White Box testing:

Branch Condition Testing

Branch Condition Combination Testing

Modified Condition Decision Testing

Data Flow Testing

Linear Code Specification And Jump Testing

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

= Black box Testing is either functional or non-functional, without reference to the internal structure of the component or system.Black Box Testing is a software testing method in which the functionalitys of

software applications are tested without having knowledge of internal code structure, implementation details and internal paths.In black-box testing, the tester is concentrating on what the software does, not how it does it.

* **Advantages**

Well suited and efficient for large code segments.

Code Access not required.

Clearly separates user's perspective from the developer's perspective

through visibly defined roles

Large numbers of moderately skilled testers can test the application with

no knowledge of implementation, programming language or operating

systems.

* **Disadvantages :**

Limited Coverage since only a selected number of test scenarios are

actually performed.

Inefficient testing, due to the fact that the tester only has limited knowledge about an application.

Blind Coverage, since the tester cannot target specific code segments or

error prone areas.

The test cases are difficult to design.

* **There are four specification-based or black-box technique:**

**1.Equivalence partitioning**

**2. Boundary value analysis**

**3. Decision tables**

**4. State transition testing**

**5. Use-case Testing**

**6. Other Black Box Testing**

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

= **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

**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

**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 big bang testing is?**

= Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system.

When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

Big Bang integration testing all the modules are integrated without performing any integration testing and then it’s executed to know whether all the integrated modules are working fine or not.

**Advantage of Big Bang Integration Testing:**

1. Big Bang testing has the advantage that everything is finished before

integration testing starts. Disadvantages of Big Bang Integration Testing:

**The major disadvantage is that in general it is very time consuming.**

1. It is very difficult to trace the cause of failures because of this late integration.

2. The chances of having critical failures are more because of integrating all the components together at same time.

3. If any bug is found then it is very difficult to detach all the modules in order to find out the root cause of it.

4. There is high probability of occurrence of the critical bugs in the production environment.

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

= Exit Criteria ensure that the testing of the application is completed and ready for release.

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

All the planned requirements must be met

All the high Priority bugs should be closed

All the test cases should be executed

End of all testing

End of phase of testing

**21. When should "Regression Testing" be performed?**

= Regression testing should be performed whenever your code base has been modified or altered in any way as well as to verify any previously discovered issues marked as fixed.

Regression testing will help your developers fix the reported defects on time, and your project to avoid any long-term pitfalls and technical debt caused by poor code quality.

However, even though an occasional project might have the resources to perform the tests after the slightest changes have been introduced to the code base, for most projects designing and maintaining such multiplicity of regression tests may simply be infeasible. Therefore, it is important to understand when you need to start regression testing.

The most common reason to run regression tests is the introduction of new functionality.

Regression testing can save developers a lot of time with timely detection of bugs that would otherwise cause the project a lot of pain in the long run.

Last but not least, regression testing has proven to be an effective means of ensuring seamless and bug-free integration with external systems.

**22.What is 7 key principles? Explain in detail?**

**= 7 key Principles of Testing:**

**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 talks about the presence of defects and doesn’t talk about the absence of defects”.

The probability of undiscovered defects remaining in a system.

The testing process does not guarantee that software is 100% error-free.

**2. Exhaustive Testing is Impossible!**

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

The inputs and outputs alone have an infinite number of combinations, so it is 100% not possible to test an application from every angle.

So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.

**3.Early Testing**

Early testing means incorporating testing as early as possible in the development process.

Early Testing should start as early as possible in the Software Development Life Cycle.

So that any defects in the requirements or design phase are captured in early stages.

**4.Defect Clustering**

Defect Clustering which states that a small number of modules contain most of the defects detected during pre-release testing or are responsible.

This is because defects are not evenly distributed within a system but are clustered.

**5.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.

This is based on the theory that when you repeatedly spray the same pesticide on crops in order to eradicate insects, the insects eventually develop an immunity,making the pesticide ineffective.

**6.Testing is Context Dependent**

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application.

According to this principle, testing depends on the context of the software developed, and this is entirely true.

**7.Absence of Errors Fallacy**

The software which we built not only must be 99% bug-free software but also it must fulfil the business, as well as user requirements otherwise it will become unusable software.

Even bug-free software may still be unusable if incorrect requirements are incorporated into the software, or if the software fails to meet the business needs.

Software testing is not finding defects, but also to check that software addresses the business needs.

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

| No. | Quality Assurance | Quality Control | Tester |
| --- | --- | --- | --- |
| 1 | Quality assurance is  process oriented. | Quality control,  alternatively, is product  oriented. | Testing is a subset of  Quality Control. |
| 2 | Quality assurance is all  about preventing  defects by ensuring the  processes used to  manage and create  deliverable works. | Quality control is the  function of software  quality that determines  the ending result is what  was expected. | Testing is the process of  executing a system in  order to detect bugs in the  product so that they get  fixed. |
| 3 | Quality Assurance is  Preventive activities. | Quality Control is a  corrective process. | Testing is a preventive  process. |
| 4 | QA is about engineering  processes that assure  quality is achieved in an  effective and efficient  way. | QC detects bugs by  inspecting and testing  the product. | Testing is an integral part  of QC as it helps  demonstrate that the  product runs the way it is  expected and designed for. |
| 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  requirements. | Activities which ensure the  identification of  bugs/error/defects in the  Software. |
| 6 | 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. |
| 7 | 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. |

**24. Difference between Smoke and Sanity?**

| **No.** | **Smoke Testing** | **Sanity Testing** |
| --- | --- | --- |
| 1 | Smoke Testing has a goal to verify stability. | Sanity Testing has a goal to verify rationality. |
| 2 | Smoke Testing is done by both  developers or testers. | Sanity Testing is done by testers. |
| 3 | Smoke Testing verifies the critical  functionality of the system. | Sanity Testing verifies the new functionality like bug fixes. |
| 4 | Smoke testing is a subset of  acceptance testing. | Sanity testing is a subset of Regression  Testing. |
| 5 | Smoke testing verifies the entire  system from end to end. | Sanity Testing verifies only a particular  component. |
| 6 | Smoke testing is usually documented  and scripted. | Sanity testing is not documented and is  unscripted. |
| 7 | It is a well elaborate and planned  testing. | This is not a planned test and is done  only when there is a shortage of time. |
| 8 | This is a wide and deep testing. | This is a wide and shallow testing. |

**25. Difference between verification and Validation?**

| No. | Verification | Validation |
| --- | --- | --- |
| 1 | Verification is static testing. | Validation is dynamic testing. |
| 2 | Verification means Are we building  the product right? | Validation means Are we building the  right product? |
| 3 | It includes checking documents,  design, codes and programs. | It includes testing and validating the  actual product. |
| 4 | Verification does not involve code  execution. | Validation involves code execution. |
| 5 | Methods used in verification are  reviews, walkthroughs, inspections  and desk-checking. | Methods used in validation are Black  Box Testing, White Box Testing and  non-functional testing. |
| 6 | Verification checks whether the  software confirms a specification. | Validation checks whether the software  meets the requirements and  expectations. |
| 7 | Verification finds the bugs early in the  development cycle. | Validation finds the bugs that  verification cannot catch. |
| 8 | Verification process targets on  software architecture, design,  database, etc. | Validation process targets the actual  software product. |
| 9 | Verification is done by the QA team. | Validation is done by the involvement of  testing team with QA team. |
| 10 | Verification process comes before  validation. | Validation process comes after  verification. |

**26. Explain types of Performance testing.**

= Performance testing is a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload.

**Types of Performance Testing:**

1. Load testing

2. Stress testing

3. Endurance testing

4. Spike testing

5. Volume testing

6. Scalability testing

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

= Testing is the process of identifying defects, where a defect is any variance between actual and expected results.

**Error:** A bug is the result of a coding error. An Error found in the development environment before the product is shipped to the customer.A programming error that causes a program to work poorly, produce incorrect results or crash. An error in software or hardware that causes a program to malfunction. A bug is the terminology of Tester.

**Defect:** It can be simply defined as a variance between expected and actual. The defect is an error found AFTER the application goes into production. It commonly refers to several troubles with the software products, with their external behaviour or with its internal features. In other words, a Defect is a difference between expected and actual results in the context of testing. It is the deviation of the customer requirement.

**Bug:** A bug is the result of a coding error. An Error found in the development environment before the product is shipped to the customer. A programming error that causes a program to work poorly, produce incorrect results or crash. An error in software or hardware that causes a program to malfunction. A bug is the terminology of Tester.

**Failure:** A failure is the inability of a software system or component to perform its required functions within specified performance requirements. When a defect reaches the end customer it is called a Failure. During development, Failures are usually observed by testers.

**28. Difference between Priority and Severity?**

| No | Priority | Severity |
| --- | --- | --- |
| 1 | Priority is a term that defines how fast  we need to fix a defect. | Severity is a term that denotes how  severely a defect can affect the  functionality of the software. |
| 2 | Priority is basically a parameter that  decides the order in which we should  fix the defects. | Severity is basically a parameter that  denotes the total impact of a given  defect on any software. |
| 3 | Priority relates to the scheduling of  defects to resolve them in software. | Severity relates to the standards of  quality. |
| 4 | The product manager basically  decides a defect’s priority level. | The testing engineer basically decides  a defect’s severity level. |
| 5 | Priority is the order in which developer  has to fix the bug. | Severity is how seriously the bug is  affecting the application. |
| 6 | If high priority is mentioned then the  developer has to fix it at the  earliest.priority status is set based on  the customer requirements. | Severity type is defined by the tester  based on the written test cases and  functionality. |
| 7 | Its value is subjective. | Its value is objective. |
| 8 | Its value changes from time to time. | Its value doesn’t change from time to  time |
| 9 | Priority is a parameter to decide the  order in which defects should be fixed. | Severity is a parameter to denote the  impact of a particular defect on the  software. |
| 10 | Priority means how fast defect has to  be fixed. | Severity means how severe defect is  affecting the functionality. |
| 11 | Priority is related to scheduling to  resolve the problem. | Severity is related to the quality  standard. |
| 12 | Product manager decides the  priorities of defects. | Testing engineer decides the severity  level of the defect. |
| 13 | Priority is of 3 types: Low, Medium,  and High. | Severity is of 5 types: Critical, Major,  Moderate, Minor, and Cosmetic. |

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

= When the expected and actual behavior is not matching, an incident needs to be raised. An incident may be a Bug. It is a programmer's fault where a programmer intended to implement a certain behavior, but the code fails to correctly conform to this behavior because of incorrect implementation in coding. It is also known as Defect.

**The different phases of Bug life cycle are :**

New or Opened

Assigned

Fixed

Tested

Closed

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

**Functional testing?**

| No | Functional Testing | Non-Functional Testing |
| --- | --- | --- |
| 1 | It verifies the operations and actions  of an application. | It verifies the behaviour of an  application. |
| 2 | It is performed before non-functional  testing. | It is performed after the functional  testing. |
| 3 | It is based on customer’s  requirements. | It focuses on customer’s expectation. |
| 4 | It is easy to define functional  requirements. | It is difficult to define the  requirements for non-functional  testing. |
| 5 | Helps to validate the behaviour of the  application. | Helps to validate the performance of  the application. |
| 6 | Carried out to validate software  actions. | It is done to validate the performance  of the software. |
| 7 | Functional testing is carried out using  the functional specification. | This kind of testing is carried out by  performance specifications |
| 8 | Functional testing is easy to execute  by manual testing. | It’s very hard to perform  non-functional testing manually. |
| 9 | It tests what the product does. | It describes how the product does. |
| 10 | Functional testing is based on the  business requirement. | Non-functional testing is based on the  performance requirement. |
| 11 | Examples of Functional Testing:  1. Unit Testing  2. Smoke Testing  3. Integration Testing  4. Regression Testing | Examples of Non-Functional Testing:  1. Performance Testing  2. Load Testing  3. Stress Testing  4. Scalability Testing |

**31. What is the difference between the STLC (Software Testing Life**

**Cycle) and SDLC (Software Development Life Cycle)?**

| No | STLC | SDLC |
| --- | --- | --- |
| 1 | STLC is mainly related to software  testing. | SDLC is mainly related to software  development. |
| 2 | It focuses only on testing the  software. | Besides development other phases  like testing is also included. |
| 3 | STLC involves only five phases or  steps.  1. Test Planning and Controlling  2. Test Analysis and Design  3. Test Implementation and  Execution  4. Evaluating Exit Criteria and  Reporting  5. Test Closure Activities | SDLC involves total six phases or  steps.  1. Requirements Gathering  2. Analysis  3. Design  4. Implementation  5. Testing  6. Maintenance |
| 4 | In STLC, less number of members  (testers) are needed. | In SDLC, more number of members  (developers) are required for the  whole process. |
| 5 | In STLC, testing team(Test Lead or  Test Architect) makes the plans and  designs | In SDLC, development team makes  the plans and designs based on the  requirements. |
| 6 | Goal of STLC is to complete  successful testing of software. | Goal of SDLC is to complete  successful development of software. |
| 7 | It helps in making the software  defects free. | It helps in developing good quality  software. |
| 8 | STLC phases are performed after  SDLC phases. | SDLC phases are completed before  the STLC phases. |
| 9 | Regression tests are run by QA team  to check deployed maintenance code  and maintains test cases and  automated scripts. | Post deployment support ,  enhancement , and update are to be  included if necessary. |
| 10 | A tested software system is the end  result of STLC. | Creation of reusable software  systems is the end result of SDLC. |

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

| No | Test Scenarios | Test Cases | Test Script |
| --- | --- | --- | --- |
| 1 | Test Scenario is any  functionality that can be  tested. | Test cases is set of  actions executed to verify  particular features or  functionality. | Test script is a set of  instruction to test an app  automatically. |
| 2 | Test Scenario is derived  from test artifacts like  Business Requirement  Specification(BRS) and  Software Requirement  Specification. | Test Cases is mostly  derived from test  scenarios. | Test Script is mostly  derived from test cases. |
| 3 | Test Scenario helps test  the end to end  functionality in an agile  way. | Test Cases helps in  exhaustive testing of an  application. | Test Script helps to test  specific things repeatedly. |
| 4 | Test Scenario is more  focused on what to test. | Test Cases is focused on  what to test and how to  test. | Test Script is focused on  the expected result. |
| 5 | Test Scenario takes less  time and fewer  resources to create. | Test Cases requires more  resources and time. | Test Script requires less  time for testing but more  resources for scripts  creating and updating. |
| 6 | Test Scenario includes  an end to end  functionality to be  tested. | Test Cases includes test  steps,data,expected  results for testing. | Test Script includes  different commands to  develop a script. |
| 7 | Test Scenario allows  quickly accessing the  testing scope. | Test Cases allows  detecting errors and  defects. | Test Script allows carrying  out an automatic  execution of test cases. |

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

= A Test Plan is a detailed document that describes the test strategy, objectives,schedule, estimation, deliverable, and resources required to perform testing for a software product.Test Plan helps us determine the effort needed to validate the quality of the application under test.

**Types of Test Plan :**

**-->There are three types of the test plan**

Master Test Plan

Phase Test Plan

Specific Test Plans

**Master Test Plan**

Master Test Plan is a type of test plan that has multiple levels of testing. It includes a complete test strategy.

**Phase Test Plan**

A phase test plan is a type of test plan that addresses any one phase of the testing strategy. For example, a list of tools, a list of test cases, etc.

**Specific Test Plans**

Specific test plan designed for major types of testing like security testing, load testing, performance testing, etc. In other words, a specific test plan designed for non- functional testing.

**The information that should be covered in Test Plan**

=Introduction to the Test Plan document

=Assumptions when testing the application

=List of test cases included in Testing the application

=What sort of Approach to use when testing the software

=The resources allocated for testing the application

**34. What is priority?**

=Priority is defined as the order in which the defects should be resolved.The priority status is usually set by the testing team while raising the defect against the dev team mentioning the time frame to fix the defect.If high priority is mentioned then the developer has to fix it at the earliest.

**Priority can be marked as either of the following states:**

**1. Low**

**2. Medium**

**3. High**

**4. Critical**

**1) Low -** This defect can be fixed after the critical ones are fixed.

**2) Medium -** The defect should be resolved in the subsequent builds.

**3) High -** The defect must be resolved immediately because the defect is affecting the application to a considerable extent and the relevant modules cannot be used until it's fixed.

**4) Critical -** The defect must be resolved immediately because the defect is affecting the application or the product severely and the product cannot be used until it has been fixed.

**35. What is severity?**

= Severity is absolute and Customer-Focused.Severity is the extent to which the defect can affect the software.It defines the impact that a given defect has on the system.

**Severity can be of following types:**

**1. Critical**

**2. Major**

**3. Moderate**

**4. Minor**

**5. Cosmetic**

**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. 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.

**2) Major:** 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. 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.

**3) Moderate:** 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:** 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.

**36. Advantage of Bugzilla?**

=**The Advantages of Bugzilla are:**

→It is an open-source widely used bug tracker

→It is easy in usage and its user interface is understandable for people without technical knowledge

→It easily integrates with test management instruments

→It integrates with an e-mailing system

→It automates documentation

→Automatic Duplicate Bug Detection.

→Search option with advanced features.

→File/Modify Bugs By Email.

→Move Bugs Between Installs.

→Multiple Authentication Methods (LDAP, Apache server).

→Time Tracking.

→Automated bug reporting; has an API to interact with system.

**37. What are the different Methodologies in Agile Development Model?**

= There are various methodologies present in agile testing and those are listed below:

**1. Scrum**

**2. eXtreme Programming**

**3. eXtreme Programming**

**4. Dynamic System Development Method (DSDM)**

**5. Test Driven Development (TDD)**

**6. Feature Driven Development(FDD)**

**7. XBreed**

**8. Crystal**

**→Scrum :** SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team based development environment. Scrum is derived from activity that occurs during rugby match. Scrum believes in empowering the development team and advocates working in small teams.

→**eXtreme Programming :** This is a light weight agile testing methodology in which development and testing happen in parallel. Business requirements are gathered in terms of stories

→**Dynamic System Development Method (DSDM) :** This is an Iterative and incremental approach that emphasizes on the continuous user involvement.

→**Test Driven Development (TDD) :** This is a technique which has short iterations where new test

→**Feature Driven Development(FDD) :** This is an iterative and incremental software development process and this can aim depends on the features.

→**XBreed :** Agile enterprise previously known as Xbreed .It is agile way of managing,architecting and monitoring the enterprise.

→**Crystal :** Crystal is an adaptive technique mainly used for software development methodologies.

**38. Explain the difference between Authorization and Authentication in Web testing.?**

| No | Authorization | Authentication |
| --- | --- | --- |
| 1 | Authorization determines what  resources a user can access. | Authentication verifies who the user is. |
| 2 | 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. |
| 3 | Authorization is the process of giving permission to access the resources. | Authentication is the process of identifying a user to provide access to a system. |
| 4 | In this, it is verified that if the user is allowed through the defined policies and rules | In this, the user or client and server are verified. |
| 5 | It is usually done once the user is successfully authenticated. | It is usually performed before the authorization. |
| 6 | Example: After employees  successfully authenticate themselves,they can access and work on certain functions only as per their roles and  profiles. | Example: Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software. |

**39. When to used Usability Testing?**

=Usability testing is an essential part of product development.You test before a redesign, you test during the redesign and then you test afterwards too.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.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.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.

**40.What is the procedure for GUI Testing?**

=The first method of GUI testing is Manual based testing.The simplest way of executing the GUI testing is purely using the application manually. Usually, manual-based testing is implemented by enthusiastic parament test engineers.

**Procedure for GUI :**

**1. MANUAL BASED TESTING**

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

**2. 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.

**3. 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.