***TOPS******Technologies Pvt.Ltd***

***Software testing Assignment***

**Module–2 (Manual Testing)**

* **What is software testing?**

**Ans: Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product **is**[**Defect**](https://www.guru99.com/defect-management-process.html)**free** and Error free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

* **What is Exploratory Testing?**

**Ans: “Exploratory testing”** – as the name suggests, is a simultaneous learning, test design, and test execution process. We can say that in this testing test planning, analysis, design and test execution, are all done together and instantly.

This testing is about exploring the system and encouraging real-time and practical thinking of a tester.

* **What is traceability matrix?**

**Ans: Traceability matrix** is a Requirements traceability is the ability to connect requirements to other artifacts such as different [types of software tests](https://www.perforce.com/resources/alm/types-software-testing) or bugs. It's used to track requirements and prove that requirements have been fulfilled.

* **What is Boundary value testing?**

**Ans: Boundary Value Testing** is one of the popular software testing mechanisms, where testing of data is done based on boundary values or between two opposite ends where the ends may be like from start to end, or lower to upper or from maximum to minimum. This testing process was introduced to select boundary values that came from the boundary based on the inputs at different ends of testing values. This black box testing strategy was introduced after equivalence class partitioning where the partition of classes takes place first followed by a partition at the boundaries.

* **What is Equivalence partitioning testing?**

**Ans:** **Equivalence partitioning testing** is a [software testing](https://en.wikipedia.org/wiki/Software_testing) technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once. This technique tries to define test cases that uncover classes of errors, thereby reducing the total number of test cases that must be developed. An advantage of this approach is reduction in the time required for testing software due to lesser number of test cases.

Equivalence partitioning is typically applied to the inputs of a tested component, but may be applied to the outputs in rare cases. The equivalence partitions are usually derived from the requirements specification for input attributes that influence the processing of the test object.

* **What is Integration testing?**

**Ans: Integration testing** is done to test the modules/components when integrated to verify that they work as expected i.e. to test the modules which are working fine individually does not have issues when integrated.

When talking in terms of testing large application using black box testing technique, involves the combination of many modules which are tightly coupled with each other. We can apply the Integration testing technique concepts.

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

**Ans:** Sometimes in projects, things go wrong. The project may go way over budget, the software team may hit a wall with development, or the product might require and then fail safety testing. Risk in project management refers to the chance that things will go wrong during the course of the project.

* **What is Alpha testing?**

**Ans: Alpha Testing** is a type of software testing performed to identify bugs before releasing the software product to the real users or public. It is a type of [acceptance testing.](https://www.guru99.com/user-acceptance-testing.html) The main objective of alpha testing is to refine the software product by finding and fixing the bugs that were not discovered through previous tests.

This testing is referred to as an alpha testing only because it is done early on, near the end of the development of the software, and before Beta Testing. Check [Differences between Alpha testing](https://www.guru99.com/alpha-beta-testing-demystified.html).

* **What is beta testing?**

**Ans: Beta testing** is a type of user acceptance testing where the product team gives a nearly finished product to a group of target users to evaluate product performance in the real world.

There is no standard for what a beta test should look like and how to set up beta testing. The actual testing procedure should be relevant to your testing goals. However, there are a few requirements that a product needs to comply with in order to be ready for beta testing.

* **What is component testing?**

**Ans: Component testing** is defined as a software testing type, in which the testing is performed on each individual component separately without integrating with other components. It’s also referred to as Module Testing when it is viewed from an architecture perspective. Component Testing is also referred to as Unit Testing, Program Testing or Module Testing.

* **What is functional system testing?**

**Ans:** **Functional testing** is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

* **What is Non-Functional Testing?**

**Ans: Non-Functional Testing** is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per non-functional parameters which are never addressed by functional testing.

* **What is GUI Testing?**

**Ans: GUI** testing refers to testing the functions of an application that are visible to a user. In the example of a calculator application, this would include verifying that the application responds correctly to events such as clicking on the number and function buttons.

* **What is Adhoc testing?**

**Ans: Adhoc Testing** is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

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

**Ans:** We are discussing here white box testing which also known as glass box is **testing, structural testing, clear box testing, open box testing and transparent box testing**. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs.

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

**Ans:** Black Box testing s a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.

**Techniques of Black Box testing are:**

* Equivalence Partitioning
* Boundary Value Analysis
* Decision Table Testing
* State Transition Testing
* Error Guessing
* Graph-Based Testing Methods
* Comparison Testing
* **Mention what are the categories of defects?**

**Ans:** Software Defect is some kind of error, flaw or some kind of mistake from the development team which prevent the software from the smooth working. It directly affects software quality; software quality is something how smooth and reliable your software is. Smoothness and reliability is how less defects your software have.

**Categories of defects:**  
**Categories of defects are:** Errors of commissions, Errors of omissions, Errors of clarity, and Error of speed and capacity.

* **Mention what bigbang testing is?**

**Ans:** 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.

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

**Ans:** Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

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

**Ans:** Whenever a new feature is developed, or when an existing feature is improved or if there are any UI updates made, ideally there is a dier need to perform software regression testing.

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

**Ans:** It is important that you achieve optimum test results while conducting software testing without deviating from the goal. But how you determine that you are following the right strategy for testing? For that, you need to stick to some basic testing principles. Here are the common seven testing principles that are widely practiced in the software industry.

* **7 key of Principles:**

1. **Exhaustive testing is not possible:** Yes! Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application.
2. **Defect Clustering:** Defect Clustering which states that a small nuof mber of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.
3. **Pesticide Paradox:** Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.
4. **Testing shows a presence of defects:** Hence, testing principle states that – Testing talks about the presence of defects and don’t talk about the absence of defects. i.e. Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.
5. **Absence of Error – fallacy:** It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user’s needs & requirements.
6. **Early Testing:** Early Testing – 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. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

* **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. All the developed software’s are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.
* **Difference between QA v/s QC v/s Tester**

**Ans:** While QA testing focuses on providing assurance that quality requested will be achieved, QC testing focuses on fulfilling the quality requested. QA focuses on preventing defect while QC focuses on identifying the defect.

* **Difference between Smoke and Sanity?**

**Ans: Smoke Testing** is a software testing technique performed post software build to verify that the critical functionalities of software are working fine. It is executed before any detailed functional or regression tests are executed. The main purpose of smoke testing is to reject a software application with defects so that QA team does not waste time testing broken software application.

**Sanity testing** is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

* **Difference between verification and Validation**

**Ans: Verification** is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is static testing.   
Verification means **Are we building the product right?**

**Validation** is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product. Validation is the dynamic testing.   
Validation means **Are we building the right product?**

* **Explain types of Performance testing.**

**Ans: Performance Testing** is a software testing process used for testing the speed, response time, stability, reliability, scalability and resource usage of a software application under particular workload. The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application. It is a subset of performance engineering and also known as “Perf Testing”.

The focus of Performance Testing is checking a software program’s

**Types of Performance Testing**

**Load testing –** checks the application’s ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.

**Stress testing –** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.

**Endurance testing –**is done to make sure the software can handle the expected load over a long period of time.

**Spike testing –** tests the software’s reaction to sudden large spikes in the load generated by users.

**Volume testing –** Under Volume Testing large no. of. Data is populated in a database and the overall software system’s behavior is monitored. The objective is to check software application’s performance under varying database volumes.

**Scalability testing –** The objective of scalability testing is to determine the software application’s effectiveness in “scaling up” to support an increase in user load. It helps plan capacity addition to your software system.

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

**Ans:** “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.”

* **Difference between Priority and Severity**

**Ans:**

**Difference Between Severity and Priority in Testing**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Severity in Testing** | **Priority in Testing** |
| Definition | Severity is a term that denotes how severely a defect can affect the functionality of the software. | Priority is a term that defines how fast we need to fix a defect. |
| Parameter | Severity is basically a parameter that denotes the total impact of a given defect on any software. | Priority is basically a parameter that decides the order in which we should fix the defects. |
| Relation | Severity relates to the standards of quality. | Priority relates to the scheduling of defects to resolve them in software. |
| Value | The value of severity is objective. | The value of priority is subjective. |
| Change of Value | The value of Severity changes continually from time to time. | The value of Priority changes from time to time. |
| Who Decides the Defect | The testing engineer basically decides a defect’s severity level. | The product manager basically decides a defect’s priority level. |
| Types | There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical. | There are 3 types of Priorities: High, Medium, and Low. |

* **What is Bug Life Cycle?**

**Ans:** Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

* **Explain the difference between Functional testing and Non-Functional testing**

**Ans: Functional Testing:**

Functional testing is a type of software testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on simulation of actual system usage but does not develop any system structure assumptions.

It is basically defined as a type of testing which verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned about the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output and comparing the actual output with the expected output.

**Non-functional Testing:**

Non-functional testing is a type of software testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects which are not tested in functional testing.

Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per non-functional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing.

* **To create HLR & TestCase of**

1. **(Instagram, Facebook) only first page**



1. **Facebook Login Page :** [**https://www.facebook.com/**](https://www.facebook.com/)



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

|  |  |  |
| --- | --- | --- |
| **SDLC** |  | **STLC** |
| SDLC is mainly related to software development. |  | STLC is mainly related to software testing. |
| Besides development other phases like testing is also included. |  | It focuses only on testing the software. |
| SDLC involves total six phases or steps. |  | STLC involves only five phases or steps. |
| In SDLC, more number of members (developers) are required for the whole process. |  | In STLC, less number of members (testers) are needed. |
| In SDLC, development team makes the plans and designs based on the requirements. |  | In STLC, testing team(Test Lead or Test Architect) makes the plans and designs. |
| Goal of SDLC is to complete successful development of software. |  | Goal of STLC is to complete successful testing of software. |
| It helps in developing good quality software. |  | It helps in making the software defects free. |
| SDLC phases are completed before the STLC phases. |  | STLC phases are performed after SDLC phases. |
| Creation of reusable software systems is the end result of SDLC. |  | A tested software system is the end result of STLC. |

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

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Script** |
| A test scenario contains high-level documentation which describes an end to end functionality to be tested. | Test cases contain definite test steps, data, expected results for testing all the features of an application. | Test Script is set of instructions or a short program to test any functionality of software application/product. |
| It focuses on more “what to test” than “how to test”. | A complete emphasis on “what to test” and “how to test.”. | Test Script is an automatic approach of software testing. |
| Test scenarios are a one-liner. So, there is always the possibility of ambiguity during the testing. | Test cases have defined a step, pre-requisites, expected result, etc. Therefore, there is no ambiguity in this process. | It is a program developed by the tester, intended to test any specific function of the software product. |
| Test scenarios are derived from test artifacts like BRS, SRS, etc. | Test case is mostly derived from test scenarios. Multiple Test case can be derived from a single Test Scenario | Automatic testing approach is beneficial for constant execution. |
| It helps in an agile way of testing the end to end functionality | It helps in exhaustive testing of an application | Test Scripting is done by scripting format. |
| Test scenarios are high-level actions. | Test cases are low-level actions. | Test script is developed in form of scripting. |
| Comparatively less time and resources are required for creating & testing using scenarios. | More resources are needed for documentation and execution of test cases. | Test Script are characterized as manual test script and automation test scripts. |

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

**Ans:** A test plan must contain data on three things; test coverage, test methods and test responsibilities. In an earlier article, [What is a Test Plan in Software Testing?](https://qa.world/test-plan-software-testing/) we explained what a test plan is and why it is essential to create one. This article lists contents for your test plans and offers a few good templates, with help on how to fill in each of the sections, with examples.

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

**Ans:**

**1. Kanban**

This type of methodology fulfills all of the Agile model’s 12 different principles. It’s an incremental process wherein transparency in software development is the primary aspect.

**2. SCRUM**

The set of development principles that this methodology type possesses is more complex, making it a highly prescriptive framework.

**3. Extreme Programming**

Extreme Programming (XP) focuses on customer satisfaction through constant development. This methodology emphasizes feedback, communication, and teamwork. Similar to SCRUM, short development cycles or sprints are also present in this framework. It creates a highly efficient and productive environment for software development teams.

**4. Crystal**

Also known as ‘lightweight methodology,’ crystal has less documentation and project review overhead. It doesn’t have a set of rules or any approach to follow, unlike all the previous frameworks we’ve mentioned so far. The Crystal methodology depends on many factors, which are categorized by several different colors

**5. Dynamic Systems Development**

Dynamic Systems Development Method (DSDM) was developed to address swift software delivery’s need for a standardized industry charter. This methodology believes that project modifications are always expected. It also emphasizes that quality with timely delivery is a must. Those are all based on a business-driven approach.

**6. Feature-driven Development**

This Agile methodology is in contrast to other frameworks like SCRUM and XP. That’s because it centers on strict operations involving domain walkthroughs. It also focuses on design, code, and inspection. Feature-Driven Development (FDD) is centered on the developer and involves turning models into builds at iterations performed every two weeks.

**7. Lean Software Development Methodology**

This Agile methodology type is all about using a holistic approach to give valuable services to customers. Waste reduction is the Lean software development framework’s basic concept.

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

**Ans:** Both **Authentication and Authorization** area units are utilized in respect of knowledge security that permits the safety of an automatic data system. Each area unit terribly crucial topics usually related to the online as key items of its service infrastructure. However, each of the terms area units is completely different with altogether different ideas. whereas indeed, they’re usually employed in an equivalent context with an equivalent tool, they’re utterly distinct from one another. In the **authentication process,** the identity of users is checked for providing the access to the system. While in the authorization process, a person’s or user’s authorities are checked for accessing the resources. Authentication is done before the authorization process, whereas the authorization process is done after the authentication process.

* **What are the common problems faced in Web testing?**

**Ans:** 1) Apps should be ADA compliant, 2) Slow network speed and poor bandwidth, 3) Firewalls Aspect, 4) Security of the data, 5) Limited web testing schedule, 6) Intranet versus Internet based Applications, 7) Several Application flows (Ins and outs) are possible, 7) Tips, techniques and tools for effective web app testing.