**Unit-05**

**Q.1) Explain performance testing with its attributes.**

* **Performance:** Performance Testing is a type of software testing that ensures software applications to perform properly under their expected workload. Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload. It’s a key step in ensuring software quality.
* **Attributes of Performance:**
* **Speed:**It determines whether the software product responds rapidly.
* **Scalability:** It determines amount of load the software product can handle at a time.
* **Stability:**It determines whether the software product is stable in case of varying workloads.
* **Reliability:**It determines whether the software product is secure or not.

**Q.2) Explain factor affecting the performance testing.**

* **Factor affecting the performance testing:**
* **Most commonly scenarios:**It means that we can find the performance scenarios based on the scenarios, which commonly used like in the Gmail application; we will perform login, inbox, send items, and compose a mail and logout.
* **Most critical scenarios:**Critical scenarios mean regularly used and important for the business-like in Gmail application login, compose, inbox, and logout.
* **Huge data transaction:** If we have huge data means that n-number of the users using the application at the same time. Once we identify the performance scenarios, we will move to the next step.

**Q.3) Describe performance testing process.**

* **Performance testing process:** The performance testing cannot be done manually since: We need a lot of resources, and it became a costlier approach. And the accuracy cannot maintain when we track response time manually.
* **The Performance testing process will be completed in the following steps:**

1. **Identify performance scenarios:** Firstly, we will identify the performance scenarios based on these below factors:

* **Most commonly scenarios:**It means that we can find the performance scenarios based on the scenarios, which commonly used like in the Gmail application; we will perform login, inbox, send items, and compose a mail and logout.
* **Most critical scenarios:**Critical scenarios mean regularly used and important for the business-like in Gmail application login, compose, inbox, and logout.
* **Huge data transaction:** If we have huge data means that n-number of the users using the application at the same time.

1. **Plan and design performance test script:** In this step, we will install the tools in the Test Engineer Machine and access the test server and then we write some script according to the test scenarios and run the tool. Once we are done with writing the script, we will go to the next step.
2. **Configure the test environment & distribute the load:** After writing the test scripts, we will arrange the testing environment before the execution. And also, manage the tools, other resources and distribute the load according to the "Usage Pattern" or mention the duration and stability.
3. **Execute test scripts:** Once we are done with distributing the load, we will execute, validate, and monitor the test scripts.
4. **Result**: After executing the test scripts, we will get the test result. And check that the result meeting the goal in the given response time or not, and the response time could be maximum, average, and minimum. If the response is not meeting the required time response, then we will go for the negative flow where will perform the below steps:
5. **Analysis result:** First, we will analyze the test result whether it meets with the response time or not.
6. **Identify the Bottleneck:** After that, we will identify the bottleneck (bug or performance issue). And the bottleneck could occur because of these aspects like the problem in code, hardware issue (hard disk, RAM Processor), network issues, and the software issue (operating system). And after finding the bottleneck, we will perform tuning (fix or adjustment) to resolve this bottleneck.
7. **Re-run test**: Once we fix the bottlenecks, re-run the test scripts and checks the result whether it meets the required goal or not.

**Q.4) What is regression testing? Explain its types.**

* **Regression** testing is a black–box testing technique performed by executing units of code repeatedly. To ensure that the on-going code modifications do not impact the system’s functionality. Alterations to the application can occur in various forms, be it new functionality, bug fixes, integrations, functionality enhancements, interfaces, patches, among others. It is advisable for regression tests to be executed as often as possible throughout the software development life cycle.
* **Types:**

1. **Unit regression:** Unit regression testing, executed during the unit testing phase, tests the code as a single unit. It has a narrow and focused approach, where complex interactions and dependencies outside the unit of code in question are temporarily blocked.
2. **Partial regression:** Partial regression is performed after impact analysis. In this testing process, the new addition of the code as a unit is made to interact with other parts of older existing code. Doing so determines that even with code modification, the system functions in silos as desired.
3. **Complete regression:** Complete regression testing is often carried out when the code changes for modification or the software updates seep way back into the roots. It is also carried out in case there are multiple changes to the existing code. A sort of a “final” regression testing is implemented to certify that the build (new lines of code) has not been altered for a period of time. This final version is then deployed to the end users.

**Q.5) Explain ad-hoc testing and its type.**

* When a software testing performed without proper planning and documentation, it is said to be Adhoc Testing. Ad-hoc testing is a type of software testing which is performed informally and randomly after the formal testing is completed to find out any loop hole in the system. Ad-hoc testing has – No Documentation, No Test cases, No Test Design.
* **Types of Ad-hoc Testing :**

1. **Buddy testing :** Buddy testing is a type of Ad-hoc testing where two bodies will be involved one is from Developer team and one from tester team. So that after completing one module and after completing Unit testing the tester can test by giving random inputs and the developer can fix the issues too early based on the currently designed test cases.
2. **Pair Testing :** Pair testing is a type of Ad-hoc testing where two bodies from the testing team can be involved to test the same module. When one tester can perform the random test and another tester can maintain the record of findings. So when two testers get paired they exchange their ideas, opinions and knowledge so good testing is performed on the module.
3. **Monkey Testing :** Testing in which the system is tested based on random inputs without any test cases. And the behavior of the system is tracked and all the functionalities of the system is working or not is monitored. -As the randomness approach is followed there is no constraint on inputs so it is called as Monkey testing.

**Q.6) Describe Exploratory testing in your own words.**

* **Exploratory testing:** A 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. In exploratory testing, software developers use their personal learning, knowledge, skills and abilities to test the software developed by themselves.

**Q.7) What is internationalization testing? Why it is necessary?**

* **Internationalization testing:** The most important factor is when an IT company develops a software product/service beyond any specific region. Then the company has to perform a test to check if the product will work at different regions of world or not as it will be used by different cultural people. Internationalization testing is a process of ensuring the adaptability of software to different cultures and languages around the world accordingly without any modifications in source code. Internationalization simply makes applications ready for localization.
* **Why is internationalization testing done**: To ensure the proper encoding of characters when a language is converted to another language. To check, if the search query or string is not supported with the targeted language then the software will not crash or malfunction. To attract audiences globally by providing convenience on using the application in their preferred languages. To make sure that the look and feel of the font and font size are rendered accordingly.

**Q.8) Where internationalization testing is performed? Explain in detail.**

* Internationalization testing is done on several important aspects that are classified into the following two parts.

1. **Internationalization testing at the Front end :**
2. **Content localization –**Localization of the static contents like labels, buttons, tabs and other fixed elements in applications, and the dynamic contents like dialogue boxes, pop-ups, toolbars, etc. Cultural awareness testing has to be done to ensure the appropriate rendering of time, date, currencies, telephone numbers, zip codes, special events and festivals on calendars used in different regions.
3. **Feature-based Testing:** Several features of an application work for certain regional users and not for others. So those features should be hidden for non-applicable users and it should be visible and functional to the users for whom they work. This is ensured by Feature-based testing.
4. **File transferring and rendering:** Property files of different languages need to be tested whether the interface of file transfer is localized as per the language being selected. Rendering means providing or displaying contents (scripts) that are appropriately displayed without misalignment or random words.
5. **Internationalization testing at the Back end :** Internationalization testing at the back end requires an in-depth understanding of the database. This testing includes the support of Unicode characters in the database.This testing also facilitates the back end (server-side) of an application to handle different languages, currencies,  encoding, site search and form data submission.

**Q.9) Explain iterative testing with advantage.**

* **Advantage :** Increased visibility and reach of target audience around the world with personalized content rendering.
* Single source code with the international standard for all versions of the application.
* The global release of the product (application) with lesser cost and time.
* Improved good quality and architecture with simpler maintenance.
* Reduced ownership cost for the various versions of the product with compliance with international standards.

**Q.10) What is agile testing? Explain its principle.**

* **Agile testing:** Agile testing is a software development practice that promotes frequent testing of new code as it is completed and stipulates that defects should be fixed as soon as they are found. Agile testing means testing software for defects or any other issues quickly or within the context of agile and give quick feedback for better and faster development of the project. The tester and developers respond to quick changes in the application rather than following a fixed plan.
* **The different principles involved in Agile Testing : Testing is continuous:** Agile team performs testing continuously because it is the only way to ensure continuous progress of the product.
* **Continuous feedback:** Agile testing provides feedback on an ongoing basis so that your product can meet the business needs.
* **Ests performed by the whole team:** In the software development life cycle, only the test team is responsible for testing. But in agile testing, the developers and the business analysts also test the application.
* **Decrease time of feedback response:** The business team is involved in each iteration in agile testing. So continuous feedback reduces the time of feedback response.

**Q.11) What is defect seeding? What care should be taken while seeding?**

* **A Defect seeding** is a practice in which defects are intentionally inserted into a program by one group for detection by another group. The purpose of this method is that  while finding the known seeded defects, the unseeded defects can also be found. Defects which  are seeded are similar to real defects. Therefore they are not very obvious and easy to detect. Defect seeding act as a method  to check the efficiency of the testing process. It serve as a confidence measure to know the percentage of defect removal rates .
* **Defect seeding we should take care of following points:** All seeded defects should be removed before release of the product. Code for defects should be written in such a way that errors can be easily identified. Number of lines should be minimum to seed defects so that efforts during defect removal becomes less. We should estimate the effort required for defect clean up and identification.

**Q.12) Describe extreme programming? When it’s applicable?**

* XP is a set of engineering practices. Developers have to go beyond their capabilities while performing these practices. That’s where the “extreme” in the framework’s title comes from. To get a better understanding of these practices, we’ll start with describing XP’s lifecycle and the roles engaged in the process.
* **When Applicable:**

1. Dynamically changing software requirements.
2. Risks caused by fixed time projects using new technology.
3. Small, co-located extended development team.
4. The technology you are using allows for automated unit and functional tests.
5. The XP framework normally involves.
6. phases of the development process that iterate continuously: **Planing, Designing, Coding, Testing.**
7. Listeningis all about constant communication and feedback. The customers and project managers are involved to describe the business logic and value that is expected.

**Q.13) What are 5 value of XP. Explain.**

* **Communication:** Everyone on a team works jointly at every stage of the project.
* **Simplicity**: Developers strive to write simple code bringing more value to a product, as it saves time and effort.
* **Feedback:**Team members deliver software frequently, get feedback about it, and improve a product according to the new requirements.
* **Respect**: Every person assigned to a project contributes to a common goal.
* **Courage:**Programmers objectively evaluate their own results without making excuses and are always ready to respond to changes.

**Q.14) Principles of XP.**

* **Principles of extreme programming**
* **Rapid feedback**: Team members understand the given feedback and react to it right away.
* **Assumed simplicity**: Developers need to focus on the job that is important at the moment.
* **Incremental changes:**Small changes made to a product step by step work better than big ones made at once**.**
* **Embracing change:**If a client thinks a product needs to be changed, programmers should support this decision and plan how to implement new requirements.
* **Quality work:**A team that works well, makes a valuable product and feels proud.

**Q.15) What is OO testing & its type.**

* **OO Testing** is a software testing process that is conducted to test the software using object-oriented paradigms like, encapsulation, inheritance, polymorphism, etc.

The software typically undergoes many levels of testing, from unit testing to system or acceptance testing.

* In higher order testing (e.g, acceptance testing), the entire system is tested with the focus on testing the functionality or external behavior of the system.
* This testing method is a data-centric technique rather than algorithmic.
* It is a technique that is based on the hierarchy of classes and well-defined objects.
* Types of OO Testing :

1. **Fault Based Testing:** It tries to identify possible faults. This is method of testing does not find all types of errors. However, incorrect specification and interface errors can be missed.
2. **Class Testing Based on Method Testing:** Each method of the class performs a well defined cohesive function. Therefore all the methods of a class can be involved at least once to test the class.
3. **Random Testing:** It is supported by developing a random test sequence that tries the minimum variety of operations typical to the behavior of the categories
4. **Partition Testing:** This methodology categorizes the inputs and outputs of a category so as to check them severely. This minimizes the number of cases that have to be designed.
5. **Scenario-based Testing:** It primarily involves capturing the user actions then stimulating them to similar actions throughout the test. These tests tend to search out interaction form of error.

**Q.16) Differentiate OO & conventional testing.**

|  |  |
| --- | --- |
| **Object-Oriented Testing** | **Conventional Testing** |
| In object-oriented Testing, a class is considered as a unit. | In conventional testing, the module or subroutine, or procedure are considered as a unit. |
| Here, we cannot test a single operation in isolation but rather as part of a class. | Here, a single operation of a procedure can be tested. |
| It focuses on composition. | It focuses on decomposition. |
| It uses an incremental approach in the testing process. | It uses a sequential approach in the testing process. |
| This testing requires at every class level wherein each class is tested individually. | This testing is following the waterfall life cycle in its testing process. |
| This testing has a hierarchical control structure. | This testing does not have any hierarchical control structure. |
| Top-down or bottom-up integration is possible in this testing. | Here, any ordering is not possible to follow. |
| In object-oriented testing, it has unit, integration, validation, and system testing as its levels of testing. | Conventional Testing also has the same levels of testing but the approach is different. |

**Q.17) What is web testing & explain its types.**

* **Website testing** is checking your web application or website for potential bugs before its made live and is accessible to general public.Web Testing checks for functionality, usability, security, compatibility, performance of the web application or website. During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.
* **Types of Web-testing :**

1. **Functionality Testing -** The below are some of the checks that are performed but not limited to the below list:Verify there is no dead page or invalid redirects, First check all the validations on each field, Wrong inputs to perform negative testing, Verify the workflow of the systemVerify the data integrity.
2. **Usability testing -**To verify how the application is easy to use with, Test the navigation and controls, Content checking, Check for user intuition.
3. **Interface testing** - Performed to verify the interface and the dataflow from one system to other and one application to another within the same system.
4. **Compatibility testing-** Compatibility testing is performed based on the context of the application, Browser compatibility, Operating system compatibility, Compatible to various devices like notebook, mobile, etc.
5. **Performance testing** - Performed to verify the server response time and throughput under various load conditions.

* **Load testing :** It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc. are also monitored.
* **Stress testing:** It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
* **Soak testing:** Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.
* **Spike testing:** Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the work load.
* **Security testing:** Performed to verify if the application is secured on web as data theft and unauthorized access. Below are some of the techniques to verify the security level of the system. 1)Injection 2)Broken Authentication and Session Management 3)Cross-Site Scripting (XSS) 4)Insecure Direct Object References 5)Security Misconfiguration

**Q.18) Explain security testing? Explain its techniques.**

* **Security testing:** Performed to verify if the application is secured on web as data theft and unauthorized access. Below are some of the techniques to verify the security level of the system.
* Injection
* Broken Authentication and Session Management
* Cross-Site Scripting (XSS)
* Insecure Direct Object References
* Security Misconfiguration

|  |  |
| --- | --- |
| **Manufacturer** | **Testing Tools** |
| **Segue** | 1. **SilkTest** 2. **SilkPerformer** 3. **SilkCentral** |
| **IBM/ Rational** | 1. **RequirementPro** 2. **Robot** 3. **ClearCase** |
| **Mercury Interactive** | 1. **WinRunner** 2. **LoadRunner** 3. **TestDirector** |
| **Compuware** | 1. **Reconcile** 2. **QALoad** 3. **QARun** |