**Module – 2**

**( Manual Testing )**

1. What is software testing ?

**Ans. Software testing is the process of finding errors in the developed product . it also check whether the real outcomes can match expected result, as well as aids in the identification of defect, missing requirements, or gaps. Testing is the penultimate step before the launch of the product to the market.**

1. What is Error, defect , bug and failure ?

**Ans.**

* **Error = A mistake in coding is called error.**
* **Defect = Error found by tester is called defect.**

**The variation between the actual results and expected results is known as defect.**

* **Bug = Defect accepted by developer team then it is called bug.**
* **Failure = when a defect reaches the end customer it is called a failure.**

1. What is 7 key principles ? Explain in detail ?

**Ans.**

**Ans.**

1. **Testing shows presence of defect = Hence ,testing principle states that – testing talks about the absence of defects. i.e. software testing reduces the probability of undiscovered defect remaining in the software but even if no defect are found , it is not a proof of correctness.**

* **But what if, you work extra hard , taking all precautions & make your software product 99% bug free , and the software does not meet the needs & requirement of the clients.**

* **This leads us to our next principle , which states that – Absence of error.**

**b) Exhaustive testing is impossible = yes ! Exhaustive testing is not possible . instead , we need the optimal amount of testing based on the risk assessment of the application.**

**- And the million dollar question is , how do you determine the risk ?**

**- To answer let do a exercise**

**- In your opinion , which operation is most likely to cause your operating system fail ?**

**- I am sure most of you would have guessed , opening 10 different application all at the same time.**

**- So if you were testing this operating system , you would realize that defect are likely to be In multi asking activities and need to be tested thorough which bring us to our next principle defect clustering .**

1. **Defect clustering = Defect clustering which states that a small number 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 20% of the modules.**
* **By experience , you can identify such risky module .**

**but this approach has its own problems.**

* **If the same tests are repeated over and over again**

**,eventually the same test cases will no longer find new bugs.**

**d) Early testing = Testing should start as early as possible in the software development life cycle . so that any defect in the requirement or design phase are captured in early stages. It is much cheaper to fix defect in the early stage testing ? it is recommended that you start finding the bug the moment the requirement are defined . more on this principle in a later training tutorial.**

**e) The pesticide paradox =**

**if the same test repeated over and over again eventually the same set of test cases will no longer find any new defects to overcome this use “ PESTICIDE PARADOX “**

**Test cases needs to be regularly reviewed and revised and new and different tests needs to be written to exercise different parts of the software or system to potentially find more defects.**

**Testing identified bugs and programmers respond to fix them , as bugs are eliminated by the programmers the software improves, as software improves the effectiveness of previous tests erodes.**

**f) Testing is context depends = Different kinds of sites are tested differently .**

**Different methodologies techniques and types and nature of the application.**

**g) Absence of error fallacy = it is possible that software which is 99% bug free 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 users needs & requirements.**

**To solve this problem , the next principle of testing stated that early testing.**

1. Difference between QA v/s QC v/s Tester

**Ans.**

|  |  |  |
| --- | --- | --- |
| **QA (quality assurance)** | **QC ( Quality control)** | **Tester** |
| **Subset of SDLC.** | **Subset of QA** | **Subset of QC** |
| **Process oriented** | **Product oriented** | **Project oriented** |
| **Preventive process** | **Corrective process** | **Corrective process** |
| **Focus on processes** | **Focus on Required quality** | **Focus on actual testing** |
| **Proactive process** | **Reactive process** | **Reactive process** |
| **Verifies the quality** | **Validates the quality** | **Validates the Quality** |
| **The whole project team involved .** | **Testing team involved** | **Testing team involved** |

1. What is boundary value testing ?

**Ans. Boundary – value analysis is a software testing technique in which tester are designed to include representative of boundary values in a range . The idea comes from the boundary .given that we have a set of test the system , a topology can be defined on that set.**

1. What is Equivalent partitioning testing ?

Ans. Equivalence partitioning is a software testing technique that divides the input data from which test cases be derived . In principal , test cases are designed to cover each partition at least once.

1. What is integration testing ?

Ans. Integration testing is the phases in software testing in which individual software modules are combined and testing as a group. Integration testing is conduct to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing.

1. What determine the level of risk ?

Ans. “Risk can uncertain future event with a probability of occurrence and occurrence and potential for loss”

In software testing risk is the possibility of a negative or undesirable outcome.

1. What is component testing ?

Ans. Component testing is also known unit testing.

A minimal software item that can be tested in isolation . it means “ a unit the smallest testable part of software”

Component testing - the testing individual software components .

1. What is functional system testing ?

Ans. 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

1. What is non – function testing ?

Ans. Testing the attributes of a component or system that do not relate to functionality , e.g , reliability , efficiency , usability , interoperability , maintainability and portability.

1. What is GUI testing ?

Ans. GUI testing is the process of testing the system’s Gul of the system under test GUI testing involves checking the screen with controls like menus , buttons, icons & all types of bar- tool bar , menu bar , dialog boxes and windows etc.

1. What is adhoc testing ?

Ans. Adhoc testing informal testing type with an aim to break the system .

* it does not follow any test case any test design techniques to create test cases.
* This testing is primarily performed if the knowledge of testers in the system under test is very high.
* Testers randomly test the application without any test cases or any business requirement document.
* Adhoc testing can be achieved with the testing techniques called error guessing.

1. What is white box testing and list the types of white box testing ?

Ans. Testing based on analysis of the internal structure of the component or system.

* Structured based testing techniques is also known as ‘ white – box or ‘ glass-box’ or ‘open-box’ testing techniques because here the testers require knowledge of how the software is implemented , how it works.
* In white box testing the tester is concentrating on how the software does it.

Types of white box testing:

1. **Statement coverage :**

* The statement coverage is also known as line coverage or segment coverage
* The statement coverage covers only true conditions

1. **Decision / branch coverage :**

* A decision is an IF statement , a loop control statement or a case statement where there are two or more outcome from statement.

1. **Condition coverage :**

* This is closely related to decision coverage but has better sensitivity to control flow.
* Condition coverage report the true or false outcome of each condition.

1. What is black box testing ? What are the different black box testing techniques ?

Ans. 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 input and outputs.

Techniques of black box testing :

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing
5. Mention what big bang testing ?

Ans. Big – bang integration testing – it is the simplest integration testing approach, where all the modules are combined and the functionality is verified after the completion of individual module testing.

In simple words, all the modules of the system are simply put together and tested.

1. What is the purpose of exit criteria ?

Ans. Successful testing or integrated application

* Executed test cases are documented
* All high prioritized bugs fixed and closed
* Technical documents to be submitted followed by release note.

1. Difference between smoke and sanity ?

Ans.

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity testing** |
| We check stability of the software | Checks the new functionality |
| Check the critical functionality | It checks the critical stage |
| It is done in initial stage | It is done after 30 built |
| Part of acceptance testing | Part of regression testing |
| It check only system end to end | It checks only a particular function of entire system |
| Who does : done by tester & developer | Who does : Done by tester |
| EX : general health check up | EX: advance health check up |
|  |  |

1. Explain difference between functional testing and non-functional testing ?

Ans.

|  |  |
| --- | --- |
| **Functional testing** | **Non Functional testing** |
| Functional testing is executed first | Non – functional testing should be performed after |
| Manual testing or automation tools can be used for F.T. | Using tools will be effective for this testing |
| Business requirement are the inputs to F.T | Performance parameters like speed, scalability are inputs to N.F.T |
| Functional testing describes what the product does. | Non – functional testing describes how good the product works. |

1. What is alpha testing ?

Ans. It is always performed by the developers at the software development site

* Sometimes it is also performed by independent testing team.
* Alpha testing is not open to the market public
* It is conducted for the software application and project
* It is always performed within the organization
* It is from of acceptance testing
* Alpha testing is definitely performed and carried out the developing organization locations with the investment of developer’s
* IT comes under the category of both white box and black box testing.

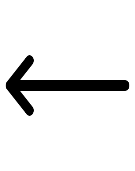
1. What is beta testing ?

Ans. It is always performed by the customer at their own site

* It is not performed by independent testing team.
* Beta testing is always open to the market and public.
* It is usually conducted for software product
* IT is performed in real time environment
* IT is 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 sit using customer data.
* IT is only a kind of black box testing.

1. What is bug life cycle ?

Ans. A computer bug is an error , flaw ,mistake , failure , or fault in a computer program that prevents it from working correctly or produces an incorrect result , bugs arise from mistake and errors , made by people in either a program’s source code or its design . “

 New

Assigned

Duplication

Rejected

Deffered

Not a bug

Open

Fixed

Pending retest

Retest

Verified

Closed

Defect stages

* New : when a new defect is logged and posted for the first time . it is assigned a status as new.
* Assigned : once the bug is posted by the tester the lead of tester approves the bug and assigned the bug to the developers team.
* Open : the developer starts analysing and works on the defect fix.
* Fixed : when a developer makes a necessary code change and verifies the change he or she can make bug status as “ fixed”
* ~ pending
* Retest : once the defect is fixed the developer gives a particular code for retesting the code to the tester . since the software testing remains pending from the testers and the status assigned is “ pending retest “ .
* Retest : tester does the retesting of the code at this stage to check whether the defect is fixed by the developer o not and changes the status to “re-test”.
* Verified : The tester re-test 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 “.
* 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.
* Closed : IF the bug is no longer exists then testers assigns the status “ closed “ .
* Duplicate : IF the defect is repeated twice or the defect corresponds to the same concept of the bug , the status is changed to “ duplicate “.
* Rejected : IF the developer feels the defect is not a genuine defect then it changes the defect to " rejected “ .
* Deferred : IF the present bug is not a prime priority and if it is expected to get fixed in the next release , then status “ Deferred “ is assigned to such bugs.
* Not a bug : If it does not affect the functionality of the application then the status assigned to a bug is “ Not a bug “ .

23 ) Difference between priority and severity ?

Ans.

|  |  |
| --- | --- |
| Severity | Priority |
| * Defect priority has defined the order in which the developer should resolve a defect . | * Defect severity is defined as the degree of impact that a defect has on the operation of the product. |
| * Priority is categorize into three types  1. Low 2. Medium 3. High | * Severity is categorized into five types  1. Critical 2. Major 3. Moderate 4. Minor 5. cosmetic |
| * priority is associated with scheduling | * severity is associated with functionality or standards |
| * priority of defect is decided in consultation with the manager /client | * QA engineer determines the severity level of the defect |
| * Priority is driven by business value | * Severity is driven by functionality |
| * Priority status is based on customer requirement | * Severity status is based on the technical aspect of the product |
|  |  |

1. What is load testing ?

ANS . load testing is to test the system behaviour under normal workload condition and it is just testing or simulating with actual workload.

* Load testing identifies the bottlenecks breaking the system under various workloads and point check how the system reacts when the point load is gradually increased.
* Load testing does not break the system.

1. What is stress testing b?

Ans. Stress testing is to test the system behaviour under extreme conditions and is carried out fill the system failure.

* In stress testing determines the of the system to reveal the maximum after which it breaks.
* Stress testing tries to break the system by testing with over whelming data or resources.

1. Difference between smoke and sanity ? -

|  |  |
| --- | --- |
| smoke | Sanity |
| * Smoke testing is performed to ascertain that the critical functionalities of the program are working fine. | * Sanity testing is done at random to verify that **each functionality** is **working as expected** |
| * Smoke testing exercises the **entire system** from end to end. | * Sanity testing exercises only the **particular component** of the entire system. |
| * The main objective of the testing is to verify the **stability** of the system. | * The main objective of the testing is to verify the **rationality** of the system. |
| * Smoke testing is usually **documented and scripted.** | * Sanity testing **is not** documented and is unscripted. |
| * This testing is performed by the **developers or testers.** | * Sanity testing in software testing is usually performed by **testers.** |
| * 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. |
| * Smoke testing is a subset of [Acceptance testing.](https://www.practitest.com/qa-learningcenter/best-practices/what-is-uat-testing/) | * Sanity testing is a subset of **Regression Testing.** |

1. Difference between verification and validation ?

Ans.

|  |
| --- |
|  |
| | Verification | Validation | | --- | --- | | It includes checking documents, design, codes and programs. | It includes testing and validating the actual product. | | Verification is the static testing. | Validation is the dynamic testing. | | It does *not* include the execution of the code. | It includes the execution of the code. | | 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. | | It checks whether the software conforms to specifications or not. | It checks whether the software meets the requirements and expectations of a customer or not. | | It can find the bugs in the early stage of the development. | It can only find the bugs that could not be found by the verification process. | | The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. | | Quality assurance team does verification. | Validation is executed on software code with the help of testing team. | | It comes before validation. | It comes after verification. | | It consists of checking of documents/files and is performed by human. | It consists of execution of program and is performed by computer. | |

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

|  |  |  |
| --- | --- | --- |
| Test scenarios | Test case | Test script |
| Is ant functionality that can be tested. | Is a set actions executed to verify particular features or functionality | Is a set of instruction to let an app automatically |
| Is derived from test artifacts like business requirement specifications and software requirements speciation. | Is mostly derived from test scenarios. | Is mostly derived from test cases. |
| Helps test the end to end functionality in an agile way | Helps in exhaustive testing of an app. | Helps to test specific things repeatedly |
| Is more focused on what to test | Is focused on what to test and how to test. | Is focused on the expected result |
| Takes less time and fewer resources to create. | Requires more resources and time | Require less time for testing but more resources for scripts creating and updating. |
| Includes an end to end functionality to be tested. | Includes test steps , data , expected results for testing. | Includes different commands to develop a script. |
| Allows quickly assessing the testing scope. | Allows detecting errors and defect | Allows carrying out an automatic execution of test cases. |

1. What is different between the STLC and SDLC ?

Ans.

|  |  |
| --- | --- |
| STLC | SDLC |
| * Testing Life Cycle | * Development Life Cycle |
| * The only objective of the STLC phase is testing. | * The main object of SDLC life cycle is to complete successful development of the software including testing and other phases. |
| * In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan | * In SDLC the business analyst gathers the requirements and create Development Plan |
| * In STLC, the test analyst creates the Integration Test Plan | * In SDLC, the development team creates the high and low-level design plans |
| * The testing team prepares the test environment and executes them | * The real code is developed, and actual work takes place as per the design documents. |
| * Testers, execute regression suits, usually automation scripts to check maintenance code deployed. | * SDLC phase also includes post-deployment supports and updates. |
|  |  |

1. Mention what are the categories of defect ?

Ans. 1 . Data quality / Database defects :- Deals with improper handling of data in the database

Example: values not deleted / inserted into the database properly.

2. Critical functionality defects :- the occurrence of these bugs hamper the crucial functionality of the application

* Example: exception

1. Functionality defects :- these defects affects the functionality of the application

* example: all java script errors

buttons like save ,cancel , delete not performing their intended functions

1. Security defects :- application security defects generally involve improper handling of data sent from the user to the application these defects ate the most severe and given highest priority for a fix

* Example: accepting an invalid username/password

1. UI defects

:- as the name suggested , the bugs deal with problems related to UI are usually considered less severe

* Example: improper error / warning / UI messages

1. When should “ regression testing “ be performed ?

Ans. Regression testing can be performed on a new build when there is significant change in the original functionality . it ensure that the code still work even when the changes are occurring .regression means re test those part of the application which are unchanged .

1. Explain what test plan ? what is the information that should cover ?

Ans. A test plan is a detailed document which describes software testing areas and activities . It outline the test strategy , objective , test schedule , required resources , test estimations and test deliverables . the test plan is base of every software testing .

1. Bug categories are ……..

Ans. Bug categories are following :

1. Performance bug
2. Security bug
3. Unit level bug
4. Functional bug
5. Usability bug
6. Syntax bug
7. Compatibility error
8. Logic bug

34 ) What are the different methodologies in agile development model ?

Ans. 1. Scrum

2. Kanban

3. lean

4. XP (Extreme programing )

5. crystal

6. FDD ( feature driven development )

7. DSDM ( dynamic systems development method )

35) explain the different between authorization and authentication in web testing . what are the common problem faces in web testing ?

Ans. Authentication :- accepting an invalid username /password

* Authorization :- accessibility to pages though permission not given

Below are five web application testing challenges faced by web developer during the development process .

* + - * Integration testing
      * Interoperability
      * Security
      * Performance
      * Usability
      * Quality testing
      * Exceptional services