**SOFTWARE TESTING**

**MODULE-2(MANUAL TESTING)**

**What is software testing?**

Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

ANSI/IEEE 1059 standard, Testing can be defined as A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

**What is exploratory testing**?

Directed from requirement and exploring during testing.

The current trend in testing is to push for automation . exploratory testing is new way of thinking

**What is traceability matrix?**

To protect against changes you should be able to trace back from every system component to the original requirement that cause its presences.

**-**Traceability matrix also known as requirements

Traceability matrix

RTM is a table which is used to trace the requirements during the software development life cycle. It can be used for forward tracing or(requirement to coding) and backward (coding to requirements.

Each requirement in the RTM document is linked with its associated test case so that testing can be done as per the mentioned requirements.

Bug id also include and linked with its associated requirements and test case

Main goal of matrix

Make sure software is developed as per the mention requirements.

Helps is finding the root cause of any bug.

1. **what is boundary value?**

Boundary value analysis is methodology for designing the test case that concentrates software testing effort on case near the limit of valid range.

* Always result in two test case per boundary for valid input and three test case per boundary for all input.
* BVA test case are designed to exercise the software on and at either side of boundary value.
* BVA use the same analysis of partition as EP and is usually used in conjunction with EP in test case design.

**What is equivalence partition testing?**

The number fall into a partition where would have the same or equivalent,result an equivalence partition or equivalence class.

In EP we must identify valid equivalence partition and invalid partition where applicable.

The valid partition is bounded by the value 1 and 100

Plus there are invalid partitions

**What is integration testing?**

Testing performed to expose defect in the interfaces and in the interaction between integrated components or system

Integration testing is level of the software testing process where individual unit are combined and tested as a group.

There are two level of integration testing

1. component integration testing
2. system integration testing

integration testing is done by a specific intergration tester and testing team

**what determines the level of risk?**

Risk – ‘A factor that could result in future negative consequences; usually expressed as impact and likelihood’

A Risk could be any future event with a negative consequence .You need to identify the risks associated with your project

Risks are of two types

Project Risks

Product Risk

**What is alpha testing?**

t is always performed by the developers at the software development site.

t is always performed by the developers at the software development site.

Alpha Testing is not open to the market and public

It is conducted for the software application and project

It is always performed in Virtual Environment.

It is always performed within the organization.

It is the form of Acceptance Testing

Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers

It comes under the category of both White Box Testing and Black Box Testing

It is always performed at the developer’s premises in the absence of the users.

**What is beta testing?**

It is always performed by the customers at their own site

It is not performed by Independent Testing Team.

Beta Testing is always open to the market and public.

It is performed in Real Time Environment.

It is also the form of Acceptance Testing. Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data

It is always performed at the user’s premises in the absence of the development team. It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area.

Beta testing can be considered “pre-release” testing

**What is component testing?**

Component (Unit) – A minimal software item that can be tested in isolation. It means

“A unit is the smallest testable part of software.”

Component Testing – The testing of individual software components

Unit Testing is a level of the software testing process where individual units/components

of a software/system are tested. The purpose is to validate that each unit of the software

performs as designed.

Unit testing is the first level of testing and is performed prior to Integration Testing

Sometimes known as Unit Testing, Module Testing or Program Testing

Unit testing frameworks, drivers, stubs and mock or fake objects are used to assist in

unit testing.

Test cases derived from component specification

Functional and Non-Functional testing

Unit tests are typically written and run by software developers to ensure that code meets

its design and behaves as intended with debugging tool.

It usually has one or a few inputs and usually a single output. In procedural programming a

unit may be an individual program, function, procedure, etc.

In object-oriented programming, the smallest unit method.

**What is functional system testing?**

.**Functional System Testing : A requirement that specifies a function that a system or**

**system component must perform**

A Requirement may exist as a text document and/or a model

There is two types of techniques

Requirement Based Functional Testing

Process Based Testing

**Functional System Testing Functionality As below:**

|  |  |
| --- | --- |
| **Accuracy** | Provision of right or agreed results or effects |
| **Interoperability** | Ability to interact with specified systems |
| **Compliance** | Adhere to applicable standards, conventions, regulations or laws |
| **Auditability** | Ability to provide adequate and accurate audit data |
| **Suitability** | Presence and appropriateness of functions for specified tasks |

**What is non functional testing?**

Non-Functional Testing: Testing the attributes of a component or system that do not

relate to functionality, e.g. reliability, efficiency, usability, interoperability,

maintainability and portability

May be performed at all Test levels (not just Non Functional Systems Testing)

Measuring the characteristics of the system/software that can be quantified on a varying

scale- e.g. performance test scaling

Non-functional testing includes, but is not limited to, performance testing, load testing,

stress testing, usability testing, maintainability testing, reliability testing and portability

testing.

**What is gui testing?**

System under Test. GUI testing involves checking the screens with the controls like menus,

buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

**Approach of GUI Testing**

**MANUAL BASED TESTING**

Under this approach, graphical screens are checked manually by testers in conformance with

the requirements stated in business requirements document.

**RECORD AND REPLAY**

GUI testing Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the

can be done using automation tools. This is done in 2 parts. During Record , test

steps are captured into the automation tool. During playback, the recorded test steps are

executed on the Application under Test. Example of such tools - QTP.

**MODEL BASED TESTING**

A model is a graphical description of system’s behavior. It helps us to understand and

predict the system behavior. Models help in a generation of efficient test cases using the

system requirements

**What is adhoc testing?**

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

It does not follow any test design techniques to create test cases.

In fact is does not create test cases altogether!

This testing is primarily performed if the knowledge of testers in the system under test is

very high.

Testers randomly test the application without any test cases or any business requirement

document.

Adhoc Testing does not follow any structured way of testing and it is randomly done on any

part of application.

Main aim of this testing is to find defects by random checking.

Adhoc testing can be achieved with the testing technique called Error Guessing.

Error guessing can be done by the people having enough experience on the system to

“guess” the most likely source of errors.

The Error guessing is a technique where the experienced and good testers are

encouraged to think of situations in which the software may not be able to cope.

Some people seem to be naturally good at testing and others are good testers because they

have a lot of experience either as a tester or working with a particular system and so are

able to find out its weaknesses.

**What is load testing?**

**Load testing** - Its a performance testing to check system behavior under load. Testing an

application under heavy loads, such as testing of a web site under a range of loads to

determine at what point the system’s response time degrades or fails.

Load testing is a kind of performance testing which determines a system’s performance

under real-life load conditions. This testing helps determine how the application behaves

when multiple users access it simultaneously.

**Pros and Cons of Load Testing**

**Pros:**

Performance bottlenecks identification before production

Improves the scalability of the system

Minimize risk related to system down time

Reduced costs of failure

Increase customer satisfaction

**Cons:**

Need programming knowledge to use load testing tools.

Tools can be expensive as pricing depends on the number of virtual users supported.

**What is stress testing?**

Stress testing - System is stressed beyond its specifications to check how and when it fails.

Performed under heavy load like putting large number beyond storage capacity, complex

database queries, continuous input to system or database load.

Stress testing is used to test the stability & reliability of the system. This test mainly

determines the system on its robustness and error handling under extremely heavy load

conditions.

It even tests beyond the normal operating point and evaluates how the system works

under those extreme conditions.

Stress Testing is done to make sure that the system would not crash under crunch

situations.

Stress testing is also known as endurance testing

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

White Box Testing based on an analysis of the internal structure of the component or system.  Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.  
Types of white box testing are:-  
∙ Structural Testing  
 ∙ Test/Code Coverage  
      \* Statement coverage

       \* Decision coverage

       \* Condition coverage  
∙ Branch Condition testing

∙ Branch Condition Combination testing

∙ Modified Condition Decision testing

∙ Dataflow testing

∙ Linear Code Sequence And Jump (LCSAJ) testing

**What is black box testing? What are different black box t**echnique?

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

Equivalence partitioning

Boundary value analysis

Decision tables

State transition testing

Use-case Testing

Other Black Box Testing

Syntax or Pattern Testing

Equivalence Partitioning (E.P.)

Aim is to treat groups of inputs as equivalent and to select one representative input to test

them all

EP can be used for all Levels of Testing

Boundary Value Analysis (B.V.A.)

Boundary value analysis is a methodology for designing test cases that concentrates

software testing effort on cases near the limits of valid ranges

Boundary value analysis is a method which refines equivalence partitioning.

Boundary value analysis generates test cases that highlight errors better than equivalence

partitioning.

The trick is to concentrate software testing efforts at the extreme ends of the equivalence

classes.

At those points when input values change from valid to invalid errors are most likely to

Decision Table

The techniques of equivalence partitioning and boundary value analysis are often applied to

specific situations or inputs.

However, if different combinations of inputs result in different actions being taken, this can

be more difficult to show using equivalence partitioning and boundary value analysis, which

tend to be more focused on the user interface.

The other two specification-based software testing techniques, decision tables and state

transition testing are more focused on business logic or business rules.

A decision table is a good way to deal with combinations of things (e.g. inputs).

This technique is sometimes also referred to as a ’cause-effect’ table. The reason for this is

that there is an associated logic diagramming technique called ’cause-effect graphing’

Decision Table

The techniques of equivalence partitioning and boundary value analysis are often applied to

specific situations or inputs.

However, if different combinations of inputs result in different actions being taken, this can

be more difficult to show using equivalence partitioning and boundary value analysis, which

tend to be more focused on the user interface.

The other two specification-based software testing techniques, decision tables and state

transition testing are more focused on business logic or business rules.

A decision table is a good way to deal with combinations of things (e.g. inputs).

This technique is sometimes also referred to as a ’cause-effect’ table. The reason for this is

that there is an associated logic diagramming technique called ’cause-effect graphing’

**Mention what are categories of defect?**

 Quality control professionals typically classify quality defects into three main categories: minor, major and critical.

**Mention what big bang testing?**

n Big Bang integration testing all components or modules are integrated simultaneously, after which everything is tested as a whole.  Big Bang testing has the advantage that everything is finished before integration testing starts.  The major disadvantage is that in general it is time-consuming and difficult to trace the cause of failures because of this late integration. Here all components are integrated together at once and then tested.

**What is the purpose of exit crite area?**

∙ To check the test logs against the exit criteria specified in test planning.

∙ To assess if more tests are needed or if the exit criteria specified should be changed.

 ∙ To write a test summary report for stakeholders.

∙ If the exit criteria have not been met

∙ Assess if more tests are needed

 ∙ Assess which test activities may need to be repeated

**Regrestion testing be performed?**

∙ when the system is stable and the system or the environment changes

 ∙ when testing bug-fix releases as part of the maintenance phase

 ∙ when change in requirements and code is modified according to the requirement

 ∙ when new feature is added to the software

 ∙ Defect fixing

 ∙ Performance issue fix

**What is 7 key principle? Explain in details?**

Testing shows presence of Defects

Exhaustive Testing is Impossible!

Early Testing

Defect Clustering

The Pesticide Paradox

Testing is Context Dependent

Absence of Errors Fallacy

**Testing shows presence of Defects**

Testing can show that defects are present, but cannot prove that there are no defects.

Testing reduces the probability of undiscovered defects remaining in the software buu

even if no defects are found, it is not a proof of correctness.

We test to find Faults

As we find more defects, the probability of undiscovered defects remaining in a system

reduces.

**Exhaustive Testing is Impossible!**

Testing everything including all combinations of inputs and preconditions is not

possible.

Example;

System has 20 screens

Average 4 menus / screen

Average 3 options / menu

Average of 10 fields / screen

2 types of input per field

Approximate total for exhaustive testing

20 x 4 x 3 x 10 x 2 x 100 = 480,000 tests

Test length = 1 sec then test duration = 17.7 days

Test length = 10 sec then test duration = 34 weeks

Test length = 1 min then test duration = 4 years

**Early Testing**

Testing activities should start as early as possible in the software or system development life

cycle, and should be focused on defined objectives.

Testing activities should start as early as possible in the development life cycle

These activities should be focused on defined objectives – outlined in the Test Strategy

Remember from our Definition of Testing, that Testing doesn’t start once the code has been

**Testing is Context Dependent**

Testing is done differently in different contexts

Different kinds of sites are tested differently.

For example

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

**Absence of Errors Fallacy**

If the system built is unusable and does not fulfill the user’s needs and expectations then

finding and fixing defects does not help.

If we build a system and, in doing so, find and fix defects....

It doesn’t make it a **good** system

Even after defects have been resolved it may still be unusable and/or does not fulfil the user

**Difference between QA V/S V/S tester.**

|  |  |
| --- | --- |
| QA | QC |
| Activities which ensure  the implementation of  processes, procedures and  standards in context to  verification of developed  software and intended  requirements. | Activities which ensure the  verification of developed  software with respect to  documented (or not in some  cases) requirements. |
| Focuses on processes and  rocedures 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. |
| Process oriented  activities. | Product oriented activities. |

Difference beetweem smoke and sanity?

|  |  |
| --- | --- |
| Smoke | sanity |
| Smoke Testing is performed to ascertain  that the critical functionalities of the  program is working fine | Sanity Testing is done to check the  new functionality / bugs have been fixed |
| The objective of this testing is to verify  the "stability" of the system in order to  proceed with more rigorous testing | The objective of the testing is to verify  the "rationality" of the system in order  to proceed with more rigorous testing |
| This testing is performed by the developers  or testers | Sanity testing is usually performed by testers |
| Smoke testing is usually documented  or scripted | Sanity testing is usually not documented and  is unscripted |
| Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |
| Smoke testing exercises the entire system  from end to end | Sanity testing exercises only the  particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health  check up |
|  |  |

Difference between verification and validation?

|  |  |  |
| --- | --- | --- |
| **Criteria** | verification | validation |
| **Definition** | The process of evaluating  work-products (not the actual  final product) of a development  phase to determine whether they  meet the specified requirements  for that phase. | The process of evaluating software during  or at the end of the development process  to determine whether it satisfies  specified business requirements |
| **Objective** | To ensure that the product is  being built according to  the requirements and  design specifications. In other  words, to ensure that work  products meet their  specified requirements. | To ensure that the product actually meets  the user’s needs, and that the  specifications were correct in the first  place.  In other words, to demonstrate that  the product fulfills its intended use  when placed in its intended environment |
| **Question** | Are we building the product right? | Are we building the right product |
| **Evaluation**  **Items** | Plans, Requirement Specs, Design  Specs, Code, Test Cases | The actual product/software |
| **Activities** | Reviews  Walkthroughs  Inspections | Testing |

**Explain types of performance testing?**

Types of Performance Testing

Load testing

Stress testing

Endurance testing

Spike testing

Volume testing

Scalability testing

**What is error,defect,bug and failure?**

**Error:** A discrepancy between a computed, observed, or measured value or condition and

the true, specified, or theoretically correct value or condition. This can be a

misunderstanding of the internal state of the software, an oversight in terms of memory

management, confusion about the proper way to calculate a value, etc.

**Defect:** Commonly refers to several troubles with the software products, with its external

behavior or with its internal features.

**Bug**: A fault in a program which causes the program to perform in an unintended or

unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology

of Tester.

**Failure:** The inability of a system or component to perform its required functions within

specified performance requirements. See: bug, crash, exception, and fault.

**Difference between priority and severity?**

|  |  |
| --- | --- |
| **Severity** | **Priority** |
| Severity is associated with functionality | Priority is associated with scheduling |
| It indicate the seriousness of defect | It indicate how soon the bug should be fixed |
| QA engineer determine the severity level | Priority of defect is consultation with the client |
| Severity is driven by functionality | Priority is driven by business level |
| Severity levels are: Critical, major, minor, moderate & Cosmetic | Priority levels are: Critical, high, medium, low |

**What is bug life cycle?**

**New :** The Bug is newly found out and entered in the Bug tracking or Bug Reporting tool.

**Open:** The Development or Test Lead reviews the defect. If it is determined to be a true

defect, he or she adjusts the severity and priority and changes the status to open. A status of

Open indicates that the defect is a true defect but that it has not yet been assigned to a

developer for correction.

**Assigned :** The bug is assigned to the Developer.

**Tested :** The bug is tested by the Software tester.

**Verified :** The bug is verified by the QA Lead.

**Closed:** The tester verifies that the defect has been resolved and changes the status to

Closed. A status of Closed indicates that the defect has been fixed and re-tested to the

satisfaction of the person who first logged the defect.

**Rejected:** If the Test Lead finds that the system is working according to the specifications

or the defect is invalid as per the explanation from the development team, he/she rejects the

defect and marks its status as ‘Rejected’. A status of Rejected indicates that the defect is

invalid, and therefore closed. No further work will be done on it.

**Deferred:** In some cases the client may determine that a particular defect stands less

important and can be deferred to a later stage. In that case it may be marked with ‘Deferred’,

with a comment indicating when it should be reviewed again. A status of Deferred indicates

that no further work will be done on the defect until that later date.

Reopened: If after retesting the defect, the problem is not solved, the tester reopens changes

the status to ‘Reopened’. A status of re-opened indicates that the developer failed to

satisfactorily fix the defect and that it needs to be re-assigned to a developer for fixing.

**Difference between functional testing and non functional?**

|  |  |
| --- | --- |
| **Functional testing** | **Non functional testing** |
| Functional testing is performed using the  functional specification provided by the client verifies the system against the functional requirments | Non-Functional testing checks the  Performance, reliability, scalability and other non fuctional aspects of the software system |
| Functional testing is executed first | Non functional testing should be performed  after functional testing |
| Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| Easy to do manual testing | Tough to do manual testing |
| Types of Functional testing are  Unit Testing  Smoke Testing  Sanity Testing  Integration Testing  White box testing  Black Box testing  User Acceptance testing  Regression Testing | Types of Nonfunctional testing are  Performance Testing  Load Testing  Volume Testing  Stress Testing  Security Testing  Installation Testing  Penetration Testing  Compatibility Testing  Migration Testing |

**To create HLR &test case of instagram facebook only first page?**

**What is the difference between stlc and sdlc?**

|  |  |
| --- | --- |
| **Sdlc** | **Stlc** |
| **Process followed by the development team within the software organization to develop a software product.** | **Process of carring out various activity to ensure the quality of software** |
| **Stands for software development life cycle.** | **Stands for software testing life cycle.** |
| **Goal of sdlc to complete succesfull development of software.** | **Goal of stlc is to complete successful testing of software.** |

**What is the difference between test senarios ,test case and test script?**

**Test scenarios:-** A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.  Test Scenario is ‘What to be tested’  Test scenario is nothing but test procedure.  The scenarios are derived from use cases.  Test Scenario represents a series of actions that are associated together.  Scenario is thread of operations .

**Test cases:-** Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.  Test Case is ‘How to be tested’  Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition.  Test cases are derived from test scenario.

**Test script:-** The Test Procedures Specification specifies the sequence of actions for a test, i.e. one or more Test Cases  It is also known as a Test Script  The Test Script can be manual or automated.

**Explain what is test plan is? What is the information that should be cover?**

**Test Plan :-** A document describing the scope, approach, resources and schedule of intended test activities.

\*  Determining the scope and risks, and identifying the objectives of testing.  Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

**What is priority?**

**Priority is Relative and Business-Focused.** Priority defines the order in which we should

resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester

to the developer mentioning the time frame to fix the defect. If high priority is mentioned

then the developer has to fix it at the earliest. The priority status is set based on the customers

requirements

Priority can be of following types:

**Low:** The defect is an irritant which should be repaired, but repair can be deferred

until after more serious defect has been fixed.

**Medium:** The defect should be resolved in the normal course of development

activities. It can wait until a new build or version is created.

**High:** The defect must be resolved as soon as possible because the defect is affecting

the application or the product severely. The system cannot be used until the repair

has been done.

**Critical:** Extremely urgent, resolve immediately

**What is severity?**

**Severity is absolute and Customer-Focused.** It is the extent to which the defect can affect

the software. In other words it defines the impact that a given defect has on the system

**Severity can be of following types:**

**Major (High):** The defect that results in the termination of the complete system or

one or more component of the system and causes extensive corruption of the data.

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.

**Moderate (Medium):** The defect that does not result in the termination, but causes

the system to produce incorrect, incomplete or inconsistent results then the severity

will be stated as moderate.

**Minor (Low):** The defect that does not result in the termination and does not damage

the usability of the system and the desired results can be easily obtained by working

around the defects then the severity is stated as minor.

**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

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

**Test Plan :-** A document describing the scope, approach, resources and schedule of intended test activities.

\*  Determining the scope and risks, and identifying the objectives of testing.  Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

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

**Ans.** There are various methodologies present in agile testing and those are listed below:

**\* Scrum**

**\* eXtreme Programming**

Below listed methodologies are used less frequently:

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

**Test Driven Development (TDD):-** This is a technique which has short iterations where new test cases covering the desired improvement or new functionality are written first.

**Feature Driven Development:-**  This is an iterative and incremental software development process and this can aim depending on the features.

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

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

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

**Ans. Authentication:-** The process of establishing the identity of the user. Authentication can take many forms including but not limited to: passwords, biometrics, radio frequency identification, etc. **Authorization:-** The process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorizati

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

**Ans.** Factors that make Web Application testing challenging:

* Apps should be ADA compliant
* Slow network speed and poor bandwidth
* Firewalls Aspect
* Security of the data
* Limited web testing schedule
* Intranet versus Internet based Applications
* Several Application flows (Ins and outs) are possible