Module – 2 (manual testing)

1) What is Software testing?

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

- software testing is a process executing a programme or application with intend of finding the software of bug.

2) What is Exploratory Testing?

Exploratory testing is concurrent process where,

- test design, execution and logging happen simultaneously
- testing is often not recorded.
- makes use of experience heuristics and test patterns.
- Testing is based on a test charter that may include
- Expected problems
- more structured than error guessing

3) What is Traceability Metrix ?

- Development life cycle. It can be used for forward tracing or backward. There are many unedified templates for RTM. Traceability Metrix is a table which is used to trace the requirement during the software .
- A requirement traceability matrix document that trace add map user requirement within the taste case ids. Purpose is to make sure that all the requirements are covered in taste case so that while testing no functionally can be missed.

Types of Traceability Matrix

1. • Forward traceability
2. • Backward traceability
3. • Bi-Directional Traceability

4) What is Boundary value testing ?

- Boundary value analysis is a methodology for designing taste case that concentrates software

- testing efforts on case near the limits of valid ranges.
- Boundary value analysis is a method which refine equivalence partitioning.
- Boundary value analysis generates taste case that highlight errors better than equivalence classes.

5) What is Equivalence partitioning testing?

- There are more effective techniques that can be used to find bugs in such circumstances
- Equivalence partitioning can help reduce the number of test from a list of all possible inputs to minimum set that would still test each partition.
- Equivalence partitioning is used to archive good input and output coverage, knowing exhaustive
- testing is often impossible.
- 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 .

6) what is integration testing?

- Integration testing performed to expose defects in the interfaces and in the interaction between integrated components or systems.
- integration testing is a level of the software testing process where individual units are combined and tested as a group.
- integration testing is done by a specific integration tester or test team.

There are two levels of integration testing

1. Component Integration testing
2. System Integration testing

7) What determines the level of risk?

A properly designed test that passes, reduces the overall level of risk in system.

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

8) What is Alpha Testing?

- Alpha testing is always performed by the developers at the software development site.
- Alpha Testing is not open to the market and public.

- Alpha testing is always performed in Virtual Environment.
- During this phase, the following will be tested in the application:

1. Spelling Mistake
2. Broken Links
3. Cloudy Directions

- Alpha testing is always performed at the time of acceptance testing when developers test the product and project to check whether it meets the user requirement or not.
- Unit testing, integration testing and system testing when combined are known as alpha testing.

9) What is beta testing?

- Beta testing is always performed by the customers at their own site.
- Beta testing is performed in Real Time Environment.
- Beta testing is performed and carried out by users on you can say people at their own locations and site using customer data.
- Beta testing can be considered “pre-rerelease” testing.
- Beta testing is only a kind of black box testing.

10) What is component testing?

Unit testing in Extreme Programming involves the extensive use of testing frameworks. A unit test framework is used in order to create automated unit tests. Below we look at some of what extreme programming brings to the world of unit testing;

- Tests are written before the code
- Rely heavily on testing frameworks
- All classes in the application are tested
- Quick and easy integration is made possible

11) What is functional system testing?

- Functional System testing is 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

1. Requirement based Functional Testing
2. Process based testing

Functional system testing Functionality as below:

i. Accuracy-Provision of right or agreed results or effects
ii. Interoperability-Ability to interact with specified systems
iii. Compliance-Adhere to applicable standards, conventions, regulations or laws

iv. Auditability-Ability to provide adequate and accurate audit data

v. Suitability-Presence and appropriateness of functions for specified tasks

12) What is non-functional testing?

- Non-functional testing is attributes of a component or system that do not relate to functionality e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

- It is the testing of "How" the system works Non-Functional testing may be performed at all test levels.

- To address this issue, performance testing is carried out to check & fine tune system response times.

- Hence load testing is carried out to check system performance at different loads i.e. number of users accessing the system.

13) What is GUI testing ?

Graphical User Interface(GUI) testing is the process of testing then system's GUI of the System under test.

GUI testing involves checking the screens with the controls like menus, button, icons and all types of bars- tool bar, menu bar, dialog boxes and windows etc.

Approach of GUI testing :

- Manual based Testing : User this approach, graphical screens are checked manually by tester in conformance with the requirements stated in business requirement document.

- Record and Replay : GUI testing can be done using automation tools.

- Model Based Testing : A model is a graphical description of system's behaviour.

14) What is Adhoc Testing?

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

- This testing is primary performed if the Knowledge of testers in the system under test is very high.

- 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.

- The Error guessing is a technique where the experienced and good testers are encouraged to think of situation in which the software may not be able to cope.

3 Types of Adhoc testing :

1. Buddy Testing : Two buddies mutually work on identify defects in the same module.

2. Pair Testing : Two tester are assigned modules, share ideas and work on the same machines to find defects.

3. Monkey Testing : Randomly test the product or app. Without test case with a goal to break the system.

15) What is White box testing and list the types of white box .

- White Box testing based on 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 tester 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.

- White box testing is also called glass testing or open box testing.

-> list the types of white box testing :

1. Branch condition testing

2. Branch condition combination testing

3. Modified Condition Decision testing

4. Dataflow Testing

5. Linear Code Sequence And Jump (LCSAJ) testing

16) What is Black Box testing? What are the different of black box testing technique ?

- Black box testing is either functional or non functional, without reference to the internal structure of the component or system.

- Specification-based testing is also known as 'black-box' or input/output driven testing techniques because they view the software as a black box with inputs and outputs.

- The testers have no knowledge of how the system or component is structured inside the box.

Techniques of black box testing :

1. Equivalence partitioning

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

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

2. Boundary Value analysis

Boundary Value analysis is a methodology for designing test case that concentrates software testing effort on case near the limits of valid ranges.

Boundary value analysis is method which refines equivalence partitioning.

3. Decision table

The technique of equivalence partitioning and boundary value analysis are often applied to specific situation or input.

Inputs are usually defined in terms of action which are Boolean (true or false)

4. State transition testing

State transaction testing uses the same principles as the state Transition Diagramming design technique.

State transaction testing is used where some aspect of the system can be described in what is called a finite state machine.

5. Use case Testing

Use case are a method of describing requirements

Usually has one main flow

Particularly useful in determining tests for system test and for UAT.

6. Other black box testing

Syntax or pattern Testing

17) mention what are the categories of defects ?

Time pressure

Complex code

Complex infrastructure

Changed technologies

18) Mention that what is big bang testing?

In Big bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

Big bang testing has the advantages is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.

19) What is the purpose of exit criteria ?

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

- End of all testing

- End of phase of testing

20) When should regression testing be performed ?

Regression should perform when change in requirements and code is modified according to the requirement.

New feature is added to the software

Defect fixing

Performance issue fix.

If your software undergoes frequent changes, regression testing costs will escalate.

Automation of regression test case is the smart choice in such case.

21) What is 7 key principle ? explain in detail.

7 Key Principles of Testing :

1. Testing Shows presence of Defects
2. Exhaustive testing is impossible!
3. Early testing
4. Defect Clustering
5. The pesticide Paradox
6. Testing is context Dependent
7. Absence of Errors Fallacy

1. Testing Shows presence of defect.

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

Testing reduce the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.

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

2. Exhaustive testing is impossible!

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

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

We have learned that we can not test everything

That iOS we must priorities our testing effort using a Risk Based Approach.

3. Early testing

Testing activities should start as early as possible in the STLC, and should be focused on defined objectives.

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

This activities should be focused on defined objectives-outlined in the Testing Strategy

4. Defect Clustering

A small number of modules contain most of the defects discovered during pre-realise testing for the most operational failures.

Defects are not evenly spread in a system

Similarly, most operational failures of a system are usually confined to a small number of modules

5. The pesticide paradox

If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.

To overcome this “Pesticide Paradox” ,the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

6) Testing is context Dependent

Testing is done differently in different contexts

Different kinds of sites are tested defiantly.

7) Absence of error fallacy

If the system built is unusable and does not fulfil the user’s needs and expectations then finding and fixing defects does not help.

Even after defects have been resolved it may still be unusable and / or does not fulfil the users.

22) Difference between QA v/s QC v/s Tester

Quality Assurance	Quality control	Tester
Activities which ensure the implementation of process, procedures and standards in context to verification of developed software and intended requirements. **Focus on processes and procedures rather than conducting actual testing on the system.**	is which ensure the verification of developed software with respect to documented.	Activities which ensure the identification of bug/defects/error in the software
Focus on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing software with intend to identify bug/defect through implementation of procedures and process.	Focuses on actual testing
Process oriented activities.	Product oriented activities.	Product oriented activities
Preventive activities	It is a corrective process.	It is preventive process.
It is subset of software development life cycle (SDLC)	QC can be considered as the subset of Quality assurance	It is a subset of quality control.

23) Difference between smoke and sanity ?

Smoke Testing	Sanity testing
Smoke testing is performed to ascertain That the critical functionalities of the Programme is work fine.	Sanity testing is done to check the new functionality / bugs have been fixed
The objectives of this testing is to verify The "stability" of the system intruder 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 tester
Smoke testing is usually documented or scripted	Sanity testing is usually not documented and unscripted
Smoke testing is a subset or regression testing	Sanity testing is a subset of acceptance testing
Smoke testing exercise 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

24) Deference between verification & validation

Verification	Validation
The process of evaluating work-products 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 satisfied business requirements.
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 users needs and that the specifications were correct6 in the first place.
Are we building the product right?	Are We build the right product ?
Plans, Requirement specs, Design	The victual product / software
Reviews **Walkthroughs** **Inspection**	Testing

25) Explain the types of performance testing .

1. Load testing
2. Endurance testing
3. Volume testing
4. Scalability testing
5. Spike testing
6. Stress testing

26) What is error , bug, defect ?

“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 “

29) Deference between Functional testing & non functional testing

Functional	Non Functional
Functional testing is performed using the Functional specification provided by the client And Verifies the system against the functional Requirements.	Non-Functional testing checks the Performance, reliability, scalability and other Non-functional aspects Of the software system.
Functional testing is executed first	Non functional testing should be performed
Manual testing or automation tools can be used For functional testing	Using tools will be effective for this testing
Business requirements are the inputs to Functional testing	Performance parameters like speed , scalability Are Inputs to non-functional testing.
Functional testing describes what the product Does	Non-functional testing describes how good the Product works
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 Acceptances	Types of Non-functional testing are • Performance Testing • Load Testing • Volume Testing • Stress Testing

27) Deference between priority and severity

Priority	Severity
Priority is a parameter to decide the order in which defects should be fixed.	Severity is a parameter to denote the impact of a particular defect on the software.
Priority means how fast defect has to be fixed.	Severity means how severe defect is affecting the functionality.
Priority is related to scheduling to resolve the problem.	Severity is related to the quality standard.
Product manager decides the priorities of defects.	Testing engineer decides the severity level of the defect.
Priority is of 3 types: Low, Medium, and High.	Severity is of 5 types: Critical, Major, Moderate, Minor, and Cosmetic.

28) what is bug life cycle ?

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.

30) what are the Deference between SDLC (software development life cycle) and STLC (software testing 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
In SDLC, more number of members (developers) are required for the whole process.	In STLC, less number of members (testers) are needed.
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.
Creation of reusable software systems is the end result of SDLC.	A tested software system is the end result of STLC.

31) What is the Deference between Test case , Test scenario and Test Script ?

Test case	Test scenarios	Test scripts
Is a set of action executed	Is any functionality that can be testing to Verify particular Features or functionality	Is a set of instructions to test an app automatically
It mostly derived from test scenarios	Is derived from test artifects like business requirement specifications and software requirement specifications (SRS)	Is mostly derived from test case
Helps in exhaustive testing of an app	Helps test the end to end functionality in agile way	Helps to test specific thing repeatedly
Is focused on what to test and how to test.	Is more focus on what to test	Is focused on expected result.
Requires more resources and time	Take less time and fewer resources to create	Requires less time for testing but more resources for script creating and uploading
Include test data, steps, expected result for testing	Include an end to end functionality to be tested	Include different commands to develop the script
Allow detecting errors and defects	Allow quickly assessing the testing scope	Allow carrying out an automatic execution of test case

32) Explain what test plan is ? What is the information that should be covered?

- Specifies how the test strategy and project test plan
- A document describing the scope, approach, resources and schedule of intended test
- Activities apply to the software under test.
- Principally:

- ➢ verify the mission
- ➢ define the Test objectives
- ➢ Specify the Test Activities required to meet the mission and objectives

- Selecting metrics for monitoring and controlling test preparation and execution, defect

- Resolution and risk issues.

- Setting the level of detail for test procedures in order to provide enough information to

- Support reproducible test preparation and execution.

Test planning major Task :

To determine the scope and risks and identify the objectives of testing.
To determine the test approach.
To implement the test policy and/or the test strategy.

Test analysis and design:

- General testing objectives are transformed into tangible Test Conditions (An item or event

Of a component or system that could be verified by one or more test cases, e.g. a function, Transaction, feature, quality attribute, or structural element) and Test Designs (A document Specifying the test conditions (coverage items) for a test item, the detailed test approach and Identifying the associated high level test cases).

Tests should be designed using the test design techniques selected in the test planning

.

Test analysis and design major task :

To review the test basis. (The test basis is the information we need in order to start the test Analysis and create our own test cases. Basically it's a documentation on which test cases are Based, such as requirements, design specifications, product risk analysis, architecture and Interfaces. We can use the test basis documents to understand what the system should do Once built.)

To identify Test Conditions/Requirements and required test data from analysis of test Items

To design the tests (note – the detail, in the form of a Test Case, is developed in the next

Test Implementation and execution :

During test implementation and execution, we take the test conditions into test cases and procedures and other test ware such as scripts for automation, the test environment and any other test infrastructure. (Test cases are a set of conditions under which a tester will determine whether an application is working correctly or no.)

33) what are Deference methodology in agile development model?

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 on 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 depends on the features.

- XBreed

Agile enterprise previously known as Xbreed .It is agile way of managing, Architecting and monitoring the enterprise.

- Crystal

Crystal is an adaptive technique mainly used for software development

Methodologies.

Scrum

SCRUM is an agile development method which concentrates particularly on how to manage Tasks within a team based development environment. Basically, Scrum is derived from Activity that occurs during rugby match. Scrum believes in empowering the development Team and advocates working in small teams (say- 7 to 9 members). It consists of three roles And their responsibilities are explained as follows:

Scrum Master: Master is responsible for setting up the team, sprint meeting and Removes obstacles to progress

Product owner: The Product Owner creates product backlog, prioritizes the backlog And is responsible for the delivery of the functionality at each iteration

Scrum Team: Team manages its own work and organizes the work to complete the sprint or cycle.

Extreme programming :

This is a light weight agile testing methodology in which development and testing happen in

Parallel. Business requirements are gathered in terms of stories.

All those stories are stored in a place called parking lot.

In this type of methodology, releases are based on the shorter cycles called Iterations with

Span of 14 days' time period.

Each iteration include phases like coding, unit testing and system testing where at each

Phase some minor or major functionality will be built in the application.

34) deference between Authorization and authentication .

Authorization	Authentication
While in authorization process, a the person's or user's authorities are checked for accessing the resources.	In the authentication process, the identity of users are checked for providing the access to the system
While in this process, users or persons are validated.	In the authentication process, users or persons are verified.
While this process is done after the authentication process.	It needs usually the user's login details.
Generally, transmit information through an Access Token.	Generally, transmit information through an ID Token.
Popular Authorization Techniques- **Role-Based Access Controls (RBAC)** **SON web token (JWT) Authorization** **SAML Authorization** **Open ID Authorization** **OAuth 2.0 Authorization**	Popular Authentication Techniques- Password-Based Authentication Password less Authentication 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) Single sign-on (SSO) Social authentication
While it needs the user's privilege or security levels.	It needs usually the user's login details
Example: After an employee successfully authenticates, the system determines what information the employees are allowed to access.	Example: Employees in a company are required to authenticate through the network before accessing their company email.

36)What are the common problem faced in web testing ?

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

Integration. Integration testing exposes problems with interfaces among different program components before deployment

Interoperability

Security

Performance

Usability

Quality Testing, Exceptional Services.