Module 2

Introduction and Fundamental

Que – 1 what is Error, Defect, Bug and failure ?

Ans – “A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirement then it is failure”

Error :- Mistake in code.

A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically corrects value or condition. This can be a misunderstanding of the internal state of the software, an oversight in term of memory management, confusion about the proper way to calculate a value, etc.

Defect :- Error found by the tester.

commonly refers to several troubles with the software products, with its external behavior or with its internal features.

Bug :- Defect accepted by the developer.

A fault in a program which cause the program to perform in an unintended or unanticipated manner. see : anomaly, defect, error, exception, and fault. Bug is terminology of tester.

Fault :- Incorrect steps / process

An incorrect steps, process, or data definition in a computer program which cause the program to perform in an unintended or unanticipated manner. see: bug, defect, error, exception.

Failure : - Build does not meet the requirements.

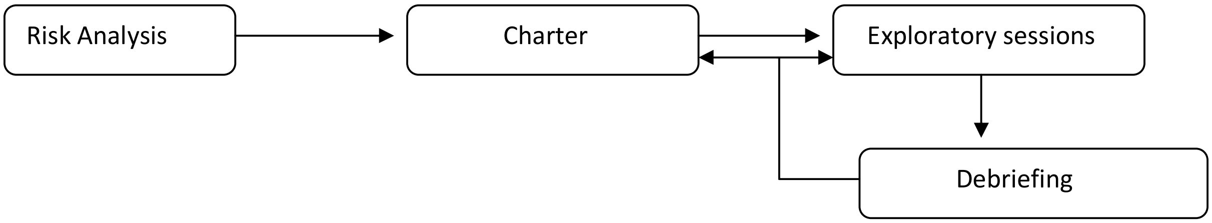
> " The inability of a system or component to perform its required function within specified performance requirements." see: bug, crash, exception and fault.

Que – 2 what is Exploratory Testing?

Ans – "To Explore application"

Exploratory testing is a concurrent process where

- Test design, execution and logging happen simultaneously
- Testing is often not recorded
- Make use of experience, heuristics and test patterns
- Testing is based on a test character that may include
  - scope of the testing
  - The focus of exploratory testing is more on testing as a "thinking" activity.
  - A brief description of how tests will be performed
  - Expected problem
- Is carried out in time boxed intervals
- More structured than Error guessing



- Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits

- is not random testing but it is Adhoc testing with purpose of find bus
- is structured and rigorous
- Is cognitively structured as compared to procedural structure of scripted testing.
- Is highly teachable and manageable.
- Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently.

Que – 3 what is Traceability matrix?

Ans – Traceability

“Mapping of the requirements with the test cases – RTM [Requirements Tractability Matrix]

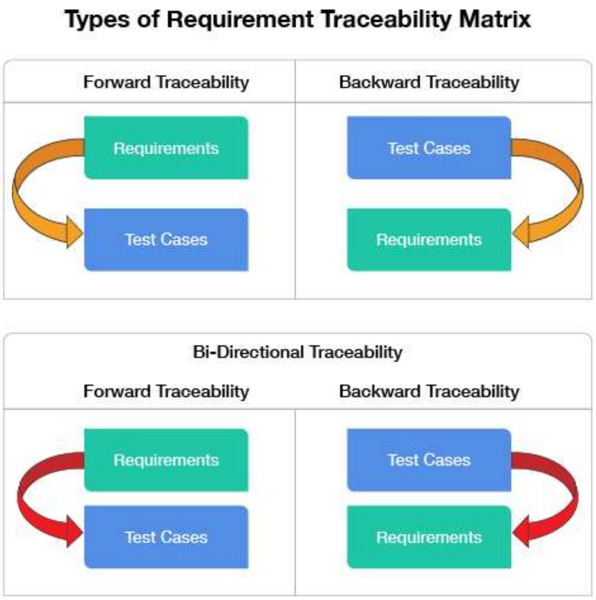
- Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability.
- Traceability can be horizontal thorough all the test documentation for a given test level or it can be vertical through the layers of development documentation (ex – from requirements to components)

Traceability matrix

- To protect against changes you should be able to trace back from every system component to the original requirement that causes its presence.
- A software process should helps you keeping the virtual up to date.
- Simple technique may be quite valuable (naming convention)

Types of Traceability Matrix :-

- Forward Traceability – Mapping of Requirements to Test cases
- Backward traceability – Mapping test cases to Requirements
- Bi – Direction traceability – A good traceability matrix is the reference from test cases to basic documentation and vice versa.



Types of Requirement Traceability Matrix

Pros of Traceability Matrix :-

- Make obvious to the client that the software is being developed as per the requirements.
- To make sure that all requirements included in the test cases.
- To make sure that developers are not creating feature that no one has requested.
- Easy to identify the missing functionalities.
- If there is a changes request for a requirement, then we can easily find out which test cases need to update.

Cons of Traceability Matrix :-

- No traceability or incomplete traceability results into :
- poor or unknown test coverage, more defect found in production
- It will lead to miss some bugs in earlier test cycles which may arise in later test cycles. Then a lot of discussions arguments with other teams and managers before release.
- Difficult projects planning and tracking, misunderstandings between different teams over project dependencies, delays etc.

Que – 4 what is Integration Testing?

Ans –“ Integration testing always perform after unit testing”

- Integration testing is a level of the software testing process where individual units are combined and tested as a group.
- ex - check the integration two or more models.

Integration testing always performed to expose defect in the interface and in the interaction between integrated components or system.

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

The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Integration testing is done by a specific integration tester or test team.

Que – 5 what is Functional System Testing?

Ans –“ Testing the attributes of the system that are directly affected to the functionalities of the system ”

Functional System Testing means a requirements that specifies a function that a system component must perform.

e.g – To check the accuracy, functions, Interoperability.

Functional Testing Example:-

1) Check OTP for any transaction
2) To check the score for every gaming level.

Que – 6 what is Non Functional Testing?

Ans –“ Testing the attributes of the system that are not directly affected to the functionalities of the system ”

Non functional Testing means Testing the attributes of a component or system that do not relate to functionality

e.g. – reliability, efficiency, usability, interoperability and portability.

Non Functional Testing Example:-

1) In mobile, automatically will switch off without any reason.

Que – 7 what is GUI Testing?

Ans –“ To check the graphical user interface by look and feel of the app ”

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

Example – Calculator

Que – 8 what is Adhoc Testing?

Ans –“ Adhoc testing is an informal testing type with an aim to break the system ”

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 cop.

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

Adhoc testing can be achieved with testing technique called error guessing.

Que – 9 what is Load Testing?

Ans –“ Load Testing is a type of performance testing that evaluates how a system, software product, or application performs under real – life load condition ” It simulates multiple users accessing the system simultaneously to identify bottlenecks and determine the maximum number of users or transaction the system can handle.

Stability + Response time + Applying load (app will with designed no of users)

Example :- App will handle 1000 users at every 5 sec

(You have to check 1000 or <= 1000 users with app)

Que – 10 what is Stress Testing?

Ans –“ It even test beyond the normal operating point and evaluate how the system work under those extreme condition.

Stress Testing is done to make sure that the system would not crash under crunch situations.

Stress testing is also known as endurance testing.

Stability + Response time + Applying load (App will with designed no of users)

Ex – App will 1000 users at every 5 sec.

You have to check 1000 or >= user with your app.

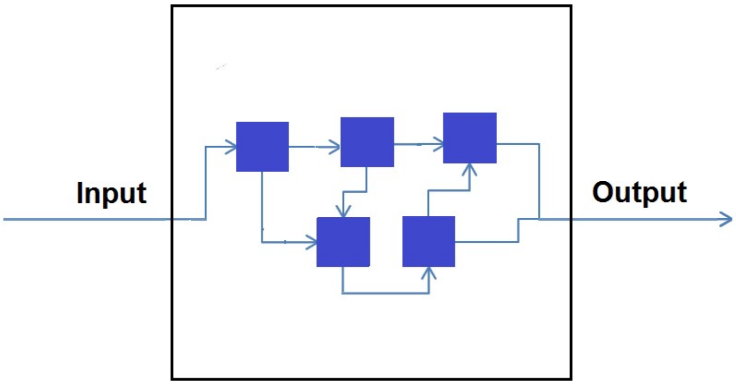
Que – 11 what is White Box Testing and list the type of white box testing?

Ans – White box testing means testing based on an analysis of the internal structure of component or system.

White box testing also known as " Glass box Testing " or "Open box Testing"

e.g – when we debug the code when we writing

Knowledge about OS, Programming lang, Technology



Types of white box testing?

1) Statement / Segment coverage
2) Decision / Branch coverage
3) Condition coverage

1) Statement / Segment coverage:-

"The statement coverage cover only true condition "

- This statement coverage is also known as line coverage or segment coverage.
- The statement coverage can be calculated as show below

$$\text{Statement coverage} = \frac{\text{Number of statements exercised}}{\text{Total number of statements}} \times 100\%$$

Advantage of statement coverage:-

- It verifies what the written code is expected to do and not to do.
- It measures the quality of code written
- It checks the flow of different paths in the program and it also ensures that whether those paths are tested or not.

Disadvantage of statement coverage:-

- It cannot test the false condition.
- It does not report that whether the loop reaches its termination condition.
-

2) Decision / Branch coverage:-

"It covers both true and false condition "

- Aim is to demonstrate that all decision have been run at least once

- With an IF statement, the exit can either be true or false, depending on the value of the logical condition that come after if.
- The decision coverage can be calculated as shown bellows :-

$$\text{Decision coverage} = \frac{\text{Number of decision outcomes exercised}}{\text{Total number of decision outcomes}} \times 100\%$$

Advantage of Decision / Branch coverage:-

- To validate that all the branches in the code are reached
- To ensure that no branches lead to any abnormality of the program's operation

Disadvantage of Decision / Branch coverage:-

- This metric ignores branches within Boolean expression which occur due to short-circuit operation.

3) Condition coverage:-

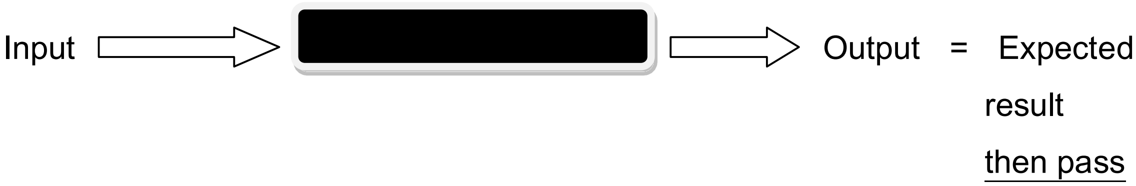
"Full condition coverage (all true/all false") does not guarantee full decision coverage.

Que – 12 what is Black Box Testing ? and What are the different Black box testing techniques?

Ans – Black Box Testing - The technique of testing without having any knowledge of the interior working of the application is a Black Box Testing.

- Directly executing the executable code without source code.

- The testers have no knowledge of how the system or component is structured inside the box.
- A tester will interact with system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.



Input

Output = Expected result then pass

Advantage :-

- Well suited and efficient for large code segments.
- Code Access not required.
- Clearly separates user's perspective from the developers perspectives through visibly defined roles.

Disadvantage :-

- Limited Coverage since only a selected number of test scenarios are actually performed
- Blind Coverage, since the tester cannot target specific code segments or error prone areas.
- The test cases are difficult to design.

Techniques / Types of Black Box Testing.

1) Equivalent Partitioning
2) Boundary Value Analysis
3) Decision Table

4) State Transition Tech

1) Equivalent Partitioning :-

e.g. emp_no : ______58______ (1 to 100)

Range (E.P)	Representative (input)	Result
1 – 20	15	pass (in bound –valid)
21- 40	22	pass
41- 60	58	pass
61- 80	77	pass
81 – 100	91	pass
101 – 120	105	Fail (out bound-invalid)

We are checking a range of numbers by this method.

In this method, Divide the range by equivalent partition, then select one representative value from each partition to the test the whole partition as pass (in range) or fail (out of range).

Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.

2) Boundary Value Analysis :-

The method refines the Equivalence partitioning method, where we can just always the boundary to get the valid range.

e.g. 1 to 50

Lower boundary (1) :	a – 1	a	a + 1	
(invalid)	0	1	2	
Upper boundary (1) :	a – 1	a	a + 1	
	0	1	2	(invalid)

3) Decision Table :-

The method to test the various combinations of inputs.

More focused on business logic or business rules.

A decision table is a good way to deal with combination of thing (e.g. inputs)

Also called ‘ Cause – effect ‘ table.

Decision will be taken by Boolean (true or false)

e.g. ->

Mr. X is having membership of amazon prime since last 2 years.

They are purchasing product now.

Mr. A is having a membership of amazon prime since last 5 years.

They are purchasing product now.

Business rules :

is member from > 3 year then annual discount will be 5 %

	Mr. X	Mr. A
having membership (>3) :	F	T

Discount (eligible) : Mr. A

4) State Transition Technique / State Transition Tech. :-

All the transaction are stored into finite machine and we are just evaluating each transaction state by state.

Example :-

To test the ATM machine transactions.

To test card then pin then valid or invalid transaction need to be tested.

Que – 13 Mention Bigbang testing is ?

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

Here all component are integrated at once, and then tested.

Advantage :-

- Easy for smaller system.

Disadvantage :-

- Fault identification is difficult.
- There should be chance to miss out to test any module or component for integration.
- Testing team will have less time for execution in the testing because of tested as a whole.
- All modules are tested at once, high risk for module isolation.

Que – 14 Difference between QA v/s QC v/c Testing

Ans –

S.N	**Quality Assurance**	**Quality control**	**Testing**
1	Activities which ensure the requirements 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 requirements.	Activities which ensure the identification of bugs/error/defects in the software.

2	Focus on processes and procedure rather than conducing actual testing on the system.	Focus on actual testing by executing software with identify bug/defect through implementation of procedures and process	Focus on actual testing.
3	Process oriented activities.	Product oriented activities.	Product oriented activities.
4	Preventive activities.	It is a corrective process.	It is a preventive process.
5	It is a subset of Software Test Life Cycle (STLC)	QC can be considered as the subset of Quality Assurance	Testing is subset of Quality Control

Que – 15 Different between Smoke and Sanity ?

Ans –

Smoke Testing	Sanity Testing
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 ‘’ stability “ of the system in order to the with more rigorous testing	The objective of this testing is to verify the “rationality” of the system in order 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 not documented or 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.

Que – 15 When should " Regression Testing " be performed ?

Ans – To test all the test cases whether they are positive or negative like there is some change in code for the previous build like insert, delete or updating any functionality or bug fixing should not affected the current functionalities.

- To much regression is not possible for manual testing so automation testing was invested for this.

Regression Testing Tools:-

1) Quick Test professional (QTP)
2) Selenium
3) Rational Function Tester (RFT)

Que – 16 Difference between Verification & Validation ?

Verification	Validation
The process of evaluating work – product of development phase to determine whether	The process of evaluating software during or at the end of the development process to

they meet the specified all that phase.	determine whether it satisfies specified business requirements.
To ensure that the product is being build according to the requirements and design specifications. In other words, to ensure that work product 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 fulfill its intended use when placed in its intended environment.
Are we building the product right….?	Are we building the right product…?
Plans, Requirement spaces, Design spaces, code, Test cases.	The actual product / software.
- Review - Walkthroughs - Inspections	- testing

Que – 17 What is Alpha testing ?

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

Alpha testing is not open to the market and public. It is always performed within the organization.

Alpha testing on "project"

It is performed in virtual environment.

It is combination of Black box testing and White box testing.

Que – 18 What is Beta testing ?

Ans – Beta testing is always performed by the customers at their own site.

It is only performed by independent testing team.

Beta testing is always open to the market and public.

Beta testing on "product"

It is performed in Real Time Enviornment.

It is only a kind of Black Box Testing.

Que – 19 What is Boundry Value testing ?

Ans – The method refines EP method, where we can just analyze the boundary get the valid range.

e.g :- 1 to 50

Lower boundary(1) :	a-1	a	a+1
(invalide)	0	1	2
Upper boundary(1) :	a-1	a	a+1
	49	50	51 (invalid)

Que – 20 what is Equivalence partitioning testing ?

Ans – Equivalent Partitioning :-

e.g. emp_no : ______58______ (1 to 100)

Range (E.P)	Representative (input)	Result
1 – 20	15	pass (in bound –valid)
21- 40	22	pass
41- 60	58	pass

61- 80	77	pass
81 – 100	91	pass
101 – 120	105	Fail (out bound-invalid)

We are checking a range of numbers by this method.

In this method, Divide the range by equivalent partition, then select one representative value from each partition to the test the whole partition as pass (in range) or fail (out of range).

Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.

Que – 21 what is determines the level of risk?

Ans – A risk – A factor that could result in future negative consequences, usually express as impact and likelihood (possibility).

To identify the risk is not enough, you need to identify the mitigation (reduce the harmness).

Que – 22 what component testing ?

Ans – Component testing also known Unit testing / Module testing / Program testing.

Unit Testing is level of the software testing process where individual units / component of the software / system are tested.

Unit tests are performed by developers In the form of debugging process.

Que – 23 Mention what are the categories of defects ?

Ans – Defect can be categorized into different types basing on the core issues they address.

Some defect address security or database issue while others may refer to functionality or UI issues.

1. Functionality Defects:- Defect directly related to functionality. Not working feature properly.
2. Performance Defects:- Software doesn’t meet the expected performance requirements.
3. User Interface Defects:- Difficult to operate for the users. Not user friendly.
4. Compability Defects:- Software does not work correctly on different hardware and software configuration.
5. Security Defects:- Application security defect generally involve improper handling of data sent from the user application. These defects are the most severe and given highest priority for a fix.
6. Documentation Defects:- Document Defects refer to flaw or imperfection in a document that can compromise its quality, functionality or legal validity.

Que – 23 Explain Types of Performance testing ?

Ans – To check the stability of the app by applying (designed no of users).

Stability + Response time + Applying load = Performance

Types of performance testing:-

1. Load Testing
2. Stress Testing

3. Scalability Testing
4. Volume Testing (Flood Testing)
5. Endurance Testing (Soak Testing)
6. Spike Testing

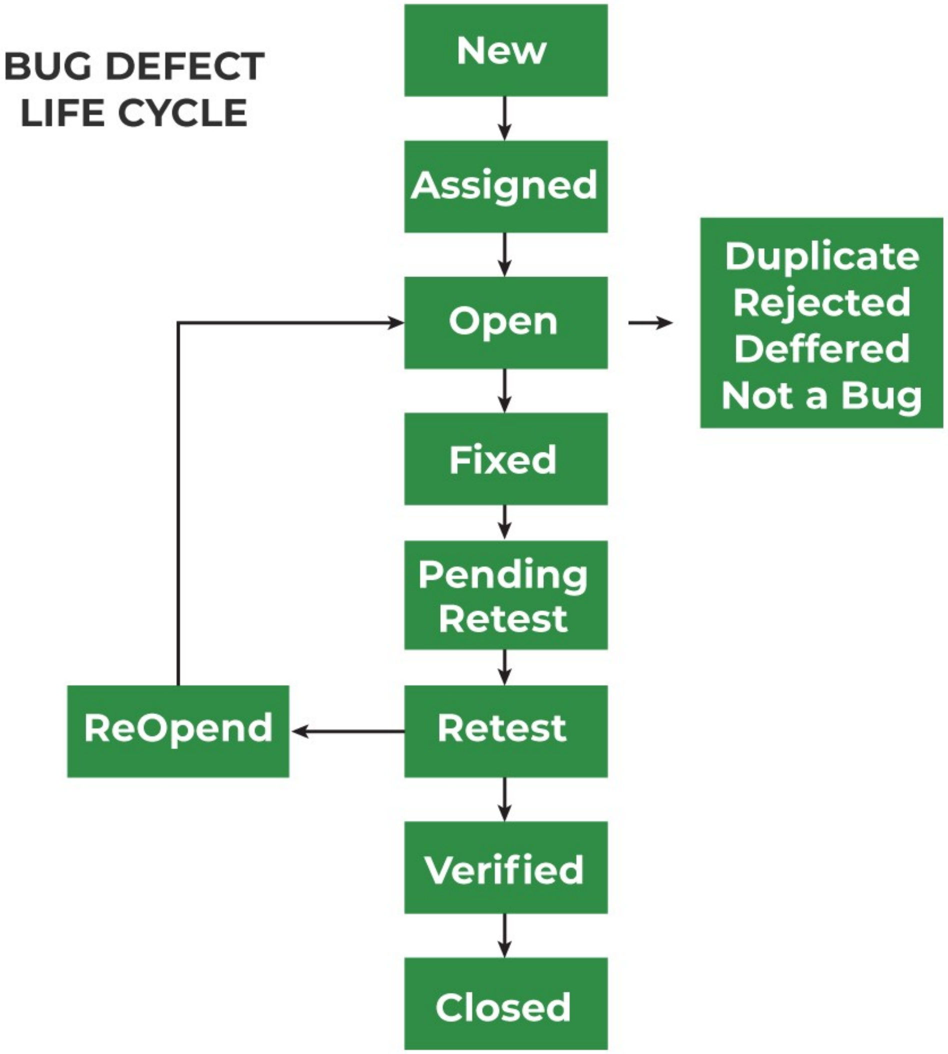
Que – 24 Difference between Priority and Severity ?

Ans –

Severity	Priority
Severity is a parameter to denote the impact of a particular defect on the software.	Priority is a parameter to decide the order in which defects should be fixed.
Severity means how sever the defect is affecting the functionality.	Priority means how fast the defect has to be fixed.
Severity is related to be quality standard.	Priority is related to scheduling to resolve the problem.
Severity is divided into 4 categories: 1. Critical 2. Major 3. Medium 4. Low	Priority is divided into 3 categories: 1. Low 2. Medium 3. High
Its value is objective.	Its value is subjective.
Its value doesn’t change time to time.	Its value changes from time to time.
It is associated with functionality or standards.	It is associated with scheduling.

Que – 25 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 mistakes and error, made by people, in either a program’s source code or its design.



Que – 26 Explain the different between Functional testing and Non functional testing.

Ans –

Functional Testing	Non Functional Testing
Functional testing performed before non functional testing.	Non function testing performed after the functional testing.
It is based on customer's requirements.	It focus on customer's expectation.
Functional Testing	Non-Functional Testing
It is easy to define functional requirements.	It is difficult to define the requirements for non –functional testing.
Helps to validate the behavior of the application.	Helps to validate the performance of the application.
Carried out to validate software actions.	It is done to validate the performance of the software.
Function testing is carried out using the functional specification.	This kind of testing is carried out by performance specifications
Example:- 1. Unit testing 2. Smoke testing 3. User Acceptance 4. Integration Testing 5. Regression testing	Example:- 1. Performance testing 2. volume testing 3. Scalability testing 4. Load testing 5. Stress testing

Que – 27 What is the purpose of exit criteria ?

Ans – When to stop testing?

- Run out of time
- Run out of budget
- The business tells you it went live last night
- Boss says no
- All the defect have been fixed ?
- When out criteria have been met ?

Que – 28 What is the different between test scenarios, test cases, and test script ?

Ans – Test Scenarios:-

- A scenario is any functionality that can be tested.
- Test scenario is “ what to be tested ”
- The scenarios are derived from use cases.
- The scenarios can be functional or non – functional to test.

Test Cases:-

- Test cases are the input & output provided to the developed software.
- Test cases is “ How to be tested”

Test Script:-

- A set of sequential instruction that detail how to execute a care business function.
- Types of Test Script

- Manual Script
- Automated Script

Que – 27 Explain what Test Plan is ? What is the information that should be covered.

Ans – Test Plan:-

- ➢ SDLC – SRS
- ➢ STLC – Test Plan , RTM

-A document describing the scope, approach (model), resources and schedule of intended test activities.

- Determining the scope and risks, and identifying the objectives of testing.

Test Plan and Strategy:-

According to test policy (rules for the organization):

-Functional Test plan

-System Integration Test Plan

-UAT Test Plan

Test Planning Factors:-

-The organization's test policy

-Scope of the testing

-Testing objectives

-Project risk

-Constraints

-Criticality

-Availability of resources

Test Planning Activities:-

-To determine the approach for the testing (model)

-To check the integration and coordination for the other module.

- To make some decision:

Who to test ?

What to test ?

When and when to stop ?

how the result will be evaluated.

-Prepare and monitor the RTM (Requirement Traceability Matrix)

Exit criteria (when to stop making Test Plan):-

-Run out time?

-Run out budget?

-The business tells you it went live last night!

-Boss say stop.

-All Defects have been fixed?

-When out exit criteria have been met?

Que – 28 What is Priority ?

Ans – If you are raising any bug for any application, how soon want the developer to fix that bug is called priority.

-Priority is considered as customer's point view. But priority can be set by the QA tester. Later on it can be changed by project manager.

-The importance given to the bug to fix it that is priority.

-The high priority indicates that the bug to fix it first.

Que – 29 What is Severity ?

Ans – The impact of Defect /bug on the customer business workflow is known as Severity..

-If that impact is more then, there is high severity.

-If that impact is less then, there is low severity.

Que – 30 Mention what are the categories of defects ?

Ans – Defect can be categorized into different types basing on the core issues they address.

Some defect address security or database issue while others may refer to functionality or UI issues.

1. Functionality Defects:- Defect directly related to functionality. Not working feature properly.
2. Performance Defects:- Software doesn't meet the expected performance requirements.
3. User Interface Defects:- Difficult to operate for the users. Not user friendly.
4. Compability Defects:- Software does not work correctly on different hardware and software configuration.
5. Security Defects:- Application security defect generally involve improper handling of data sent from the user application. These defects are the most severe and given highest priority for a fix.
6. Documentation Defects:- Document Defects refer to flaw or imperfection in a document that can compromise its quality, functionality or legal validity.

Que – 31 Advantage of Bugzila ?

Ans – Advantages of Bugzila :-

1. Effective Bug Tracking :

   Bugzila provides a centralized platform for tracking and managing software defects.

2. Customization and Flexibility :

   Bugzila is highly customizable, allowing organizations to tailor it to their specific needs.

3. Integrated Collaboration:

   Bugzila supports collaboration among team members by providing a platform from discussions and attachments related to each bug reports.

4. Automation and Workflow Management

   Bugzila offers automation feature that streamline bug tracking processes.

5. Comprehensive Reporting and analysis

   Bugzila includes robust reporting tools that enable users to generate a variety of reports, charts, and ghraphs related to bug statistics and project progress.

Que – 32 What are the different Methodologies in Agile Development Model?

Ans – Agile Manifesto / 4 Values of Agile Menifesto:

1. Individuals and Interaction over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan