

## Module-2

### Q.1 What is Exploratory Testing?

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 bugs
- Is structured and rigorous
- Is cognitively (thinking) structured as compared to procedural structure of scripted testing. This structure comes from Charter, time boxing etc.
- 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.

### Q.2 What is traceability matrix?

Traceability testing is a type of software testing that ensures there is a relationship between the requirements and the corresponding test cases. It verifies that each requirement is tested by at least one test case and that each test case is derived from a requirement

- The purpose is to ensure that all requirements are covered by the testing process and to provide a way to trace defects back to their source requirements.
- This helps in validating that the software meets the specified requirements and facilitates better management of the testing process.
- To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.
- A software process should help you keeping the virtual table up to date.

### Q.3 What is Boundary value testing?

Boundary value testing is a software testing technique that focuses on testing the boundary or edge conditions of input ranges.

- It involves selecting test cases at the boundaries of input domains , including the minimum,maximum and just beyond those limits.
- The idea is to identify any errors or issues that may arise at these critical points where the behavior of the software might change .
- By testing boundary values , testers can often uncover defects that might go unnoticed with typical input values.
- This technique helps improve the robustness and reliability of the software.

#### **Q.4 What is Equivalence partitioning testing?**

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.
- **If we want to test the following if statement :**
- “If value is between 1 and 100 (inclusive) (e.g value  $\geq 1$  and value  $\leq 100$ ) Then...”

We could put a range of numbers as shown in the below figure.

- EP says that by testing just one value we have tested the partition (typically a mid-point value is used).
- It assumes that: If one value finds a bug, the others probably will too If one doesn't find a bug, the others probably won't either.

#### **Q.5 What is Integration testing?**

Integration testing- testing performed to expose defects in the interface and in the interactive 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.  
There are 2 levels of integration testing.
  - Component integration testing
  - System integration testing

## **Q.6 What determines the level of risk?**

“ A factor that could result in future negative consequence; usually expressed as impact and likely hard.”

### **Types of Risk**

- Project Risk
- Product Risk

### **Examples :**

Example of project risk is senior team member leaving the project abruptly.

Every risk assigned a likelihood i.e. chance of it occurring , typically on a scale of 1 to 10. Also the impact of that risk is identified on a scale of 1-10.

- But just identifying the risk is not enough you needs to identifying mitigation.
- In this case mitigation could be knowledge transfer to other team members & having a buffer tester in place.
- Example of product risk would be flight reservation system not installing in test.
- Mitigation in this case would be conducting a smoke or sanity testing . accordingly you will make changes in your scope items to include sanity testing.

## **Q.7 What is Alpha testing?**

- 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 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 form of acceptance testing.
- Alpha testing is definitely performed and carried out at the developing a organization location with the involvement of developers.
- It comes under the category of both white box testing and black box testing.

### **Q.8 What is beta testing ?**

- 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 always performed outside the organization.
- 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 location and site using customer data.
- It is only a kind of black box testing.
- Beta testing can be considerd “ pre- release “ testing.
- Pilot testing is testing to product on real world as well as collect data on the use of product in the class room.

### **Q. 9 What is component testing?**

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 component.

- Unit testing is a level of the software testing process where individual unit/ component 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.

## **Q.10 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 Test Approach

- 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 specific

## **Q.11 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

- 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.
- 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. The goal of performance testing is to reduce response time to an acceptable level
- Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

### **Non – functional testing example :**

**Web Based Testing :** Identify the software processes that directly influence the overall performance of the system.

● In website number of user/customer will increase , how the website will handled to every customer/user.

● **Desktop Based Testing :** Numerous other such GUI test cases, the desktop application tester must view Guarantee that error messages are instructive and helpful for the client Memory, and different other issues

● **Mobile Based Testing :** In mobile , automatically will switch off without any reason. To stop the application which is not in our hand.

● **Game Based Testing :** Confirms workability and stability of the software. Validate whether the user interface of the app is as per the screen size of the device and ensure high quality

### **Q.12 What is GUI Testing?**

Graphical User Interface (GUI) testing is the process of testing the system's GUI of the 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** 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.

## Q.13 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 it 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.
- Using experience to postulate errors.
- Use Error Guessing to Complement Test Design Techniques. 25

#### **Q.14 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.

**This testing usually identifies** – The maximum operating capacity of an application

- Determine whether current infrastructure is sufficient to run the application
- Sustainability of application with respect to peak user load
- Number of concurrent users that an application can support, and scalability to allow more users to access it.
- It is a type of non-functional testing.
- Load testing is commonly used for the Client/Server, Web based applications – both Intranet and Internet.

## **Q.15 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.

## **Q.16 What is white box testing and list the types of white box testing ?**

**White Box Testing:** 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.

White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

**list the types of white box testing :-**

- **Branch Condition testing** Branch Condition Testing requires that the True and False of each Boolean operand is tested (Boolean Operands in this example: If A > 30 and B >= 5)

- **Branch Condition Combination testing** Branch Condition Combination Coverage would require all combinations of Boolean operands to be evaluated
- **Modified Condition Decision testing** Modified Condition Decision Coverage requires test cases to show that each Boolean operand can independently affect the outcome of the decision
- **Dataflow testing** Data flow testing aims to execute sub-paths from points where each variable in a component is defined to points where it is referenced.
- **Linear Code Sequence And Jump (LCSAJ) testing** LCSAJ testing requires a model of the source code which identifies control flow jumps (where control flow does not pass to a sequential statement).

## **Q.17 What is black box testing? What are the different black box testing techniques?**

**Black-box testing:** 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 inputs and outputs.

### **Techniques of Black Box Testing**

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

## **Q.18 Mention what are the categories of defects?**

Defect is the variance from a desired product attribute (it can be a wrong, missing or extra data).

### **Types of Defects:-**

Data Quality/Database Defects: Deals with improper handling of data in the database. **Examples:** Values not deleted/inserted into the database properly

- Improper/wrong/null values inserted in place of the actual values
- Critical Functionality Defects: The occurrence of these bugs hampers the crucial functionality of the application.

Examples: - Exceptions

- Functionality Defects: These defects affect the functionality of the application.

Examples: All JavaScript

- errors Buttons like Save, Delete, Cancel not performing their intended functions
- A missing functionality (or) a feature not functioning the way it is intended to
- Continuous execution of loops

### **Security Defects:**

Application security defects generally involve improper handling of data sent from the user to the application.

These defects are the most severe and given highest priority for a fix.

Examples:

**Authentication:** Accepting an invalid username/password Authorization: Accessibility to pages though permission not given

- User Interface Defects: As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

**Examples:**

- I mproper error/warning/UI messages
- Spelling mistakes
- Alignment problems

## **Q.19 Mention what bigbang testing is?**

- In 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.

## **Q.20 What is the purpose of exit criteria?**

**Exit Criteria:** Successful Testing of Integrated Application.

Executed Test Cases are documented

All High prioritized bugs fixed and closed

Technical documents to be submitted followed by release Notes.

- Limitations

Any condition not specified in integration tests, apart from the confirmation of the execution of the design items is usually not tested.

**Exit criteria :** Exit criteria defines the items that must be completed before testing can be concluded.

In an ideal world , you will not enter the next stage until the exit criteria for the previous stage is met. But practically this is not always possible.

## **Q.21 When should "Regression Testing" be performed?**

- Quick Test Professional (QTP)
- Rational Functional Tester (RFT)
- Selenium

**When use the testing tools?**

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

- In such cases, Manual execution of test cases increases test execution time as well as costs. Automation of regression test cases is the smart choice in such cases.
- Extent of automation depends on the number of test cases that remain reusable for successive regression cycles.

## **Q.22 What is 7 key principles? Explain in detail?**

### **General Testing Principles**

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 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 but, 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.
- However Testing cannot prove that there are no defects present

#### **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 risks and priorities to focus testing efforts.
- For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.

- This is very unlikely that the project timescales would allow for this number of tests.
- So, assessing and managing risk is one of the most important activities and reason for testing in any project.
- We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions).
- That is we must Prioritise our testing effort using a Risk Based Approach.

### **3. 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 written!

### **4. Defect Clustering**

- A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
- Defects are not evenly spread in a system
- They are ‘clustered’ In other words, most defects found during testing are usually confined to a small number of modules
- Similarly, most operational failures of a system are usually confined to a small number of modules An important consideration in test prioritisation!

### **5. 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.
- Testing identifies 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

## 6. Testing is Context Dependent Testing

- Testing is Context Dependent Testing is basically 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.
- Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing
  
- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software
- 1 to 3 failures per KLOC typical for industrial software
- 1 failure per KLOC for NASA Shuttle code!
- Also different industries impose different testing standards

## 7. 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 users' needs and expectations

### **Q.23 Difference between QA v/s QC v/s Tester .**

<b>Sr.no.</b>	<b>Quality assurance</b>	<b>Quality control</b>
<b>1</b>	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.
<b>2</b>	Focuses 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.
<b>3</b>	Process oriented activities.	Product oriented activities.
<b>4</b>	Preventive activities.	It is a corrective process.
<b>5</b>	It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.

### **Q.24 Difference between Smoke and Sanity?**

<b>Sr.no.</b>	<b>Smoke testing</b>	<b>Sanity testing</b>
<b>1</b>	Smoke Testing is performed to ascertain that the critical functionality of the programme is working fine.	Sanity testing is done to check the new

		functionality bugs have been fixed.
<b>2</b>	The objective of this testing is to verify "stability" of the system in order to proceed to more rigorous testing	The objective of the testing is to verify the "rationality" of the system in order to proceed to more rigorous testing
<b>3</b>	This testing is performed by the developers or tester.	Sanity testing is usually performed by testers
<b>4</b>	Smoke testing is usually documented or scripted	Sanity testing is usually not documented and is scripted.
<b>5</b>	Smoke testing is a subset of Regression testing	Sanity testing is a subset of Acceptance testing
<b>6</b>	Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system
<b>7</b>	Smoke testing is like General Health Checkup.	Sanity Testing is like specialized health check up.

## Q.25 Difference between verification and Validation

criteria	Verification	validation
Definition	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.	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	To ensure that the product actually meets the user's

	design specifications. In other words, to ensure that work products meet their specified requirements.	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

## Q.26 Explain types of Performance testing.

### Types of Performance Testing

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

#### 1. Load testing

Load testing is to test the system behavior under normal work load condition , and it is just testing or simulating with the actual work load.

- Load testing identifies the bottlenecks breaking the system under various work loads and checks how the system reacts when the load is gradually increased.

- Load testing does not break the system.

## **2. 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.

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.

## **3. Endurance testing**

Endurance testing is a type of performance testing that evaluates the systems ability to handle a sustained workload over an extended period. It helps identify potential issues related to resource exhaustion , memory leaks , and performance degradation under continuous usage.

## **4. Spike testing**

Spike testing is a type of performance testing that involves subjecting a system to sudden and extreme increase in load or traffic to assess its stability and response under such conditions. This helps evaluate how the system handles sudden bursts of activity and if it can recover gracefully without crashing or experiencing performance degradation.

## **5. Volume testing**

volume testing is a type of performance testing that involves testing a system with a large amount of data to evaluate its performance scalability , resource utilization and response time. It helps ensure that the system can handle the expected volume of data without experiencing performance issues such as slowdowns or crashes.

## **6. Scalability testing**

Determines maximum user load the software application can handle.

## **Q.27 What is Error, Defect, Bug and failure?**

“ 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.”

## **Q.28 Explain the difference between Functional testing and NonFunctional testing**

<b>Sr.no.</b>	<b>Functional testing</b>	<b>Non functional testing</b>
<b>1</b>	Functional testing is performed using the 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
<b>2.</b>	Functional testing is executed first	Non functional testing should be performed after functional testing
<b>3.</b>	Manual testing or automation tools can be used for functional testing	Using tools will be effective for this testing
<b>4.</b>	Business requirements are the inputs to functional testing	Performance parameters like speed , scalability are inputs to non-functional testing.
<b>5.</b>	Functional testing describes what the product does	Nonfunctional testing describes how good the product works
<b>6.</b>	Easy to do manual testing	Tough to do manual testing

<b>7.</b>	<p>Types of Functional testing are</p> <ul style="list-style-type: none"> <li>· Unit Testing</li> <li>· Smoke Testing</li> <li>· Sanity Testing</li> <li>· Integration Testing</li> <li>· White box testing</li> <li>· Black Box testing</li> <li>· User Acceptance testing</li> <li>· Regression Testing</li> </ul>	<p>Types of Nonfunctional testing are</p> <ul style="list-style-type: none"> <li>· Performance Testing</li> <li>· Load Testing</li> <li>· Volume Testing</li> <li>· Stress Testing</li> <li>· Security Testing</li> <li>· Installation Testing</li> <li>· Penetration Testing</li> <li>· Compatibility Testing</li> <li>· Migration Testing</li> </ul>
-----------	--	--

### **Q.29 What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle) ?**

<b>Sr.no.</b>	<b>STLC</b>	<b>SDLC</b>
<b>1.</b>	Requirement Phase Testing also known as Requirement Analysis in which test team studies the requirements from a testing point of view to identify testable requirements and the QA team may interact with various stakeholders to understand requirements in detail.	SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are a number of different development models.
<b>2.</b>	<p><b>STLC phases :-</b></p> <ol style="list-style-type: none"> <li>1. Requirement Analysis</li> <li>2. Test Planning</li> <li>3. Test case development</li> </ol>	<p><b>SDLC phases:-</b></p> <ol style="list-style-type: none"> <li>1. Requirements Collection/Gathering</li> <li>2. Analysis</li> </ol>

	4. Test Environment setup 5. Test Execution 6. Test Cycle closure	3. Design 4. Implementation 5. Testing 6. Maintenance
3.	Activities in Requirement Phase Testing:- 1.Preparation of test plan/strategy 2.document for various types of testing 3.Test tool selection 4.Test effort estimation Resource 5. planning and determining roles and responsibilities. 6.Training requirement	Maintenance is the process of changing a system after it has been deployed:- 1.Corrective maintenance 2. Adaptive maintenance 3. Perfective maintenance

**Q.30 What is the difference between test scenarios, test cases, and test script?**

Sr.no.	<b>Test scenarios</b>	<b>Test cases</b>	<b>Test script</b>
1.	A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.	Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.	A set of sequential instruction that detail how to execute a core business function
2.	Test Scenario is 'What to be tested'	Test Case is 'How to be tested'	One script is written to explain how to simulate each business scenario
3.	Test scenario is nothing but test procedure.	Test case consist of set of input values, execution	Written to a level of detail for which someone else

		precondition, expected Results and executed post- condition developed to cover certain test Condition.	(other than the script writer) would be able to easily execute
<b>4.</b>	The scenarios are derived from use cases	Test cases are derived (or written) from test scenario. T	There are various means for executing test scripts. 1. Manual Testing 2. Automation Testing
<b>5.</b>	Test Scenario represents a series of actions that are associated together.	Test Case represents a single (low level) action by the user	Written to a level of detail for which someone else (other than the script writer) would be able to easily execute
<b>6.</b>	Scenario is thread of operations	Scenario is thread of operations	Should demonstrate how the system can support the HCA warehouse business processes

**Q.31 Explain what Test Plan is? What is the information that should be covered.**

Test Planning in STLC is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project.

The Test Plan gets prepared and finalized in the same phase.

Activities in Requirement Phase testing:-

1. Preparation of test plan/strategy
2. document for various types of testing
3. Test tool selection
4. Test effort estimation Resource planning and determining roles and responsibilities.
5. Training requirement

#### **Deliverables of Requirement Phase Testing**

- Test plan /strategy document.
- Effort estimation document.

### **Q.32 What are the different Methodologies in Agile Development Model?**

- The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.
  - Agile is a philosophy, i.e., a set of values and principles to make a decision for developing software.
  - Agile is based on the iterative-incremental model. In an incremental model, we create the system in increments, where each increment is developed and tested individually.

### **Q.33 when to used usability testing?**

Usability testing is typically used during the design and development process of product or service to evaluate how easy and intuitive it is for users to interact with .its beneficial to conduct usability testing.

- 1. During initial design phases:** To gather feedback on early prototypes or wireframes before investing significant resources in to development.
- 2. During development:-** to identify and address usability issues as they arise ensuring a smoother user experience.
- 3. Before lunch :** to ensure that the final product meets user needs and expectations , reducing the risk of negative user experiences and costly redesigns post - launch.
- 4. For comparative analysis:** - to compare the usability of different versions or designs to inform decision making.

### **Q.34 What is the procedure for 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**

- 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.