SOFTWARE TESTING ASSIGNMENT MODULE 2

Q-1. What is Exploratory testing ?

Ans:-

Exploratory Testing is a type of software testing where Test cases are not created in advance but testers check system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is more on testing as a "thinking" activity.

Exploratory Testing is widely used in Agile models and is all about discovery, investigation, and learning. It emphasizes personal freedom and responsibility of the individual tester.

Q-2 WHAT IS TRACEABILITY MATRIX ?

Ans:-

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. It is used to track the requirements and to check the current project requirements are met.

Q– 3 What is Boundary Value Testing?

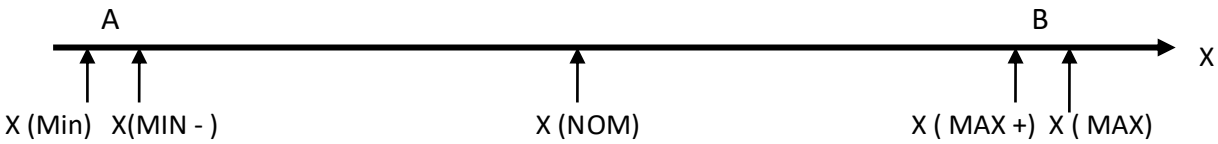
Ans:-

Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.

So these extreme ends like Start- End, Lower- Upper, Maximum-Minimum, Just Inside-Just Outside values are called boundary values and the testing is called "boundary testing".

The basic idea in normal boundary value testing is to select input variable values at their:

- Minimum
- Just above the minimum
- A nominal value
- Just below the maximum
- Maximum



Qua – 4 WHAT IS EQUIVALENCE PARTITIONING TESTING ?

Ans -

Equivalence Partitioning also called as equivalence class partitioning. It is abbreviated as ECP. It is a software testing technique that divides the input test data of the application under test into each partition at least once of equivalent data from which test cases can be derived.

An advantage of this approach is it reduces the time required for performing testing of a software due to less number of test cases.

Qua - 5 WHAT IS INTEGRATION TESTING ?

ANS:-

Integration testing is known as the second level of the software testing process, following unit testing. Integration testing involves checking individual components or units of a software project to expose defects and problems to verify that they work together as designed.

SOFTWARE TESTING ASSIGNMENT MODULE 2

Qua – 6 What determines the leval of Risk ?

Ans-

The level of risk are

1.Project risk - To identify the risk associated with your project
Example – A senior team member leaving the project shortly.
It is found at the time of developing mode.

2.Product Risk – To identify the risk associated with your product
Example – A flight reservation system did not installed the test environment.
It is fount after the development

Q – 7 What is alpha testing?

Ans -

Alpha testing is the initial phase of validating whether a new product will perform as expected. Alpha tests are carried out early in the development process by internal staff and are followed up with beta tests, in which a sampling of the intended audience actually tries the product out.

Qua – 8 What is Beta Testing?

Ans:-

Beta testing is an opportunity for real users to use a product in a production environment to uncover any bugs or issues before a general release.
Beta testing is the final round of testing before releasing a product to a wide audience. The objective is to uncover as many bugs or usability issues as possible in this controlled setting.

Beta testers are "real" users and conduct their testing in a production environment running on the same hardware, networks, etc., as the final release.

Qua-9 What is Component Testing?

Ans :-

Component testing is defined as a software testing type, in which the testing is performed on each individual component separately without integrating with other components. It's also referred to as Module Testing when it is viewed from an architecture perspective. Component Testing is also referred to as Unit Testing, Program Testing or Module Testing.

Qua – 11 What is Functional System testing ?

Ans :-

Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Qua – 12 What is Non-Functional Testing?

Ans:-

Non-Functional Testing is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional

testing. An excellent example of non-functional test would be to check how many people can simultaneously login into a software.

Qua-13 What is GUI Testing?

Ans:-

Graphic User Interface Testing (GUI) testing is the process of ensuring proper functionality of the graphical user interface (GUI) for a specific application. This involves making sure it behaves in accordance with its requirements and works as expected across the range of supported platforms and devices.

Qua – 14 What is the Adhoc testing ?

Ans: -

Ad hoc Testing is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

Qua – 15 What is load testing ?

Ans:-

Load testing is a type of performance testing that simulates a real-world load on any software, application, or website. Without it, your application could fail miserably in real-world conditions. That's why we build tools like Retrace to help you monitor application performance and fix bugs before your code ever gets to production. Load testing examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end-users. This tool is typically applied when a software development project nears completion.

Q. 16 What is stress Testing?

Ans:-

Stress Testing is a software testing technique that determines the robustness of software by testing beyond the limits of normal operation. Stress testing is particularly important for critical software but is used for all types of software. Stress testing emphasizes robustness, availability, and error handling under a heavy load rather than what is correct behavior under normal situations. Stress testing is defined as a type of software testing that verifies the stability and reliability of the system.

Q.17 What is white box testing and list the types of white box testing?

Ans :-

White Box Testing is a testing technique in which software's internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

White Box Testing Types:-

- **Statement Coverage**:- This technique requires every possible statement in the code to be tested at least once during the testing process of software engineering.
- **Branch Coverage –** This technique checks every possible path (if-else and other conditional loops) of a software application.
- **Condition coverage –** This is closely related to decision coverage but has better sesivity to contro flow.
  - Condition coverage reports the true or false outcomes of each conditions**.**
  - It covers the most path & code of the software and also optimise code so hidden error can be identify.

Q.18 What is black box testing? What are the different black box testing techniques?

Ans :-

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

Black Box Testing Techniques

Equivalence Partitioning

Testers can divide possible inputs into groups or "partitions", and test only one example input from each group. For example, if a system requires a user's birth date and provides the same response for all users under the age of 18, and a different response for users over 18, it is sufficient for testers to check one birth date in the "under 18" group and one date in the "over 18" group.

Boundary Value Analysis

Testers can identify that a system has a special response around a specific boundary value. For example, a specific field may accept only values between 0 and 99. Testers can focus on the boundary values (-1, 0, 99 and 100), to see if the system is accepting and rejecting inputs correctly

.

Decision Table Testing

Many systems provide outputs based on a set of conditions. Testers can then identify "rules" which are a combination of conditions, identify the outcome of each rule, and design a test case for each rule.

State Transition Testing

In some systems, significant responses are generated when the system transitions from one state to another. A common example is a login mechanism which allows users to authenticate, but after a specific number of login attempts, transition to a different state, locking the account. If testers identify a state transition mechanism, they can design test cases that probe the system when it transitions states. For example, for a system that locks the account after five failed login attempts, a test case can check what happens at the sixth login attempt.

Q.19 Mention what bigbang testing is?

Ans:-

Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

Q.21 What is the purpose of exit criteria?

Ans :-

Exit criteria in testing are often viewed as a single document commemorating the end of a life cycle phase. It can be defined as “The specific conditions or on-going activities that should be fulfilled before completing the software testing life cycle. STLC specifies which exit criteria is required at each testing phase”. The exit criteria can identify the intermediate deliverables and enable you to track them as independent events.

Q.22 When should "Regression Testing" be performed?

Ans:-

Regression testing should be performed are,

1. We do regression testing whenever the production code is modified.
2. When new functionalities are added to the application.
3. When there is a change requirement in the application.
   Ex. Remember password should be removed from the login page which is available earlier
4. When there is a defect
5. When there is a performance issue fixed
   Ex. Loading the home page takes 5 seconds reducing the load time to 2 second.
6. When there is a environment change
   Ex. Updating the database from my sql to oracle

Q.23 WHAT IS 7 KEY PRINCIPLES ? EXPLAIN IN DETAILS

Ans.

1. **Testing shows a presence of defects.**

Hence, testing principle states that – Testing talks about the presence of defects and don’t talk about the absence of defects. i.e. Software 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. If, you work extra hard, taking all precautions & make your software product 99% bug-free. And the software does not meet the needs & requirements of the clients.

2. **Exhaustive testing is not possible.**

Exhaustive testing is not possible because, we need the optimal amount of testing based on the risk assessment of the application.And the million dollar question is, how do you determine this risk? To answer this let's do an exercise In your opinion, Which operation is most likely to cause your Operating system to fail? I am sure most of you would have guessed, Opening 10 different application all at the same time. So if you were testing this Operating system, you would realize that defects are likely to be found in multi-tasking activity and need to be tested thoroughly which brings us to our next principle Defect Clustering.

3. Defect Clustering

Defect Clustering which states that a small number of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules. By experience, you can identify such risky modules. But this approach has its own problems.If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

4. Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

5. Absence of Error – fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user's needs & requirements.To solve this problem, the next principle of testing states that Early Testing

6. Early Testing

Early Testing – Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

7. Testing is context dependent

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed software's are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.

SOFTWARE TESTING ASSIGNMENT MODULE 2

Q. 24 Difference between QA vs QC vs Tester.

Ans:-

QUALITY ASSURANCE ENGINEER	QUALITY CONTROL	TESTER
1. QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	1. It includes activities that ensure the verification of a developed software with respect to documented(or not in somecases) requirements.	1.It includes activities that ensure the identification of bugs/error/defects in a software.
2. Focuses on processes and procedures rather than conducting actual testing on the system.	2. Focuses on actual testing by executing the software with an aim to identify bug/defect through implementation of procedures and process.	2. Focuses on actual testing.
3. Process-oriented activities.	3. Product-oriented activities.	3. Product-oriented activities.
4. Preventive activities.	4. It is a corrective process.	4. It is a preventive process.
5. It is a subset of Software Test Life Cycle (STLC).	5. QC can be considered as the subset of Quality Assurance	5.Testing is the subset of Quality Control.

Q.25 – DIFFERENCE BETWEEN SMOKE & SANITY TESTING.

SMOKE TESTING	SANITY TESTING
• The purpose of smoke testing is to verify the critical functionalities of a system,	• Sanity testing verifies the new functionality such as bug fixes
• The goal of Smoke testing is to verify stability	• The goal of Sanity testing is to verify rationality.
• Software Developers or Testers perform smoke testing.	• Only testers perform sanity testing.
• Smoke testing is a subset of acceptance testing.	• Sanity testing is a subset of regression testing.
• Smoke testing is documented or scripted,	• Sanity testing is not documented or scripted.
• In smoke testing, the entire system is verified from end to end.	• In sanity testing, on the other hand, only a particular component of the system gets verified.
• Smoke test is done to make sure that the critical functionalities of the program are working fine.	• Sanity testing is done to check that newly added functionalities, bugs, etc., have been fixed.
• The software build may be either stable or unstable during smoke testing.	• The software build is relatively stable at the time of sanity testing.
• Smoke testing is done on initial builds	• Sanity testing is done on relatively stable builds
• Smoke testing is done as a part of basic testing	• Sanity testing is done as part of regression testing.

• Smoke testing is usually done every time there is a new build release.	• Sanity testing is planned when there is not sufficient time for in-depth testing.
• Smoke testing is like a general health checkup.	• Sanity testing resembles a specialized health checkup.

Q.26 DIFFERENCE BETWEEN VERIFICATION AND VALIDATION.

Ans:-

VALIDATION	VERIFICATION
1. Evaluates the final product to check whether it meets the business needs.	1. Evaluates the intermediary products to check whether it meets the specific requirements of the particular phase.
2. It determines whether the software is fit for u satisfies the business needs.	2. Checks whether the product is built as per the specified requirement and design specification.
3. Checks "Are we building the right product"?	3. Checks "Are we building the product right"?
4. Is done with executing the software.	4. This is done without executing the software.
5 Includes all the dynamic testing techniques.	5. Involves all the static testing techniques.
6 Example includes all types of testing like smoke, regression, functional, systems and UAT.	6. Examples include reviews, inspection, and walkthrough.

Q.27 Explain types of Performance testing.

Ans:-

Types of Performance Testing for Software

1. Load Testing

Load testing measures system performance as the workload increases. That workload could mean concurrent users or transactions. The system is monitored to measure response time and system staying power as workload increases. That workload falls within the parameters of normal working conditions.

2. Stress Testing

Unlike load testing, stress testing — also known as fatigue testing — is meant to measure system performance outside of the parameters of normal working conditions. The software is given more users or transactions that can be handled. The goal of stress testing is to measure the software stability. At what point does software fail, and how does the software recover from failure?

3. Spike Testing

Spike testing is a type of stress testing that evaluates software performance when workloads are substantially increased quickly and repeatedly. The workload is beyond normal expectations for short amounts of time.

4. Endurance Testing

Endurance testing — also known as soak testing — is an evaluation of how software performs with a normal workload over an extended amount of time. The goal of endurance testing is to check for system problems such as memory leaks. (A memory leak occurs when a system fails to release discarded memory. The memory leak can impair system performance or cause it to fail.)

5. Scalability Testing

Scalability testing is used to determine if software is effectively handling increasing workloads. This can be determined by gradually adding to the user load or data volume while monitoring system

performance. Also, the workload may stay at the same level while resources such as CPUs and memory are changed.

6. Volume Testing

Volume testing determines how efficiently software performs with large projected amounts of data. It is also known as flood testing because the test floods the system with data.

Q.28 WHAT IS ERROR, DEFECT, BUG AND FAILURE ?

- **ERROR** – A mistake in coding is called error
- **DEFECT** – Error found by tester is called defect.
- **BUG** – Defect accepted by the development team then it is called bug.
- **FAILURE** – Build does not meet the requirements that it is called failure.

Q. 29 Difference between Priority and Severity

Ans:-

Parameters	Severity in Testing	Priority in Testing
Definition	Severity is a term that denotes how severely a defect can affect the functionality of the software.	Priority is a term that defines how fast we need to fix a defect.
Parameter	Severity is basically a parameter that denotes the total impact of a given defect on any software.	Priority is basically a parameter that decides the order in which we should fix the defects.
Relation	Severity relates to the standards of quality.	Priority relates to the scheduling of defects to resolve them in software.
Value	The value of severity is objective.	The value of priority is subjective.
Change of Value	The value of Severity changes continually from time to time.	The value of Priority changes from time to time.
Who Decides the Defect	The testing engineer basically decides a defect's severity level.	The product manager basically decides a defect's priority level.
Types	There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.	There are 3 types of Priorities: High, Medium, and Low.

Q.30 What is Bug Life Cycle?

Ans:-

The bug life cycle in testing refers to a cycle of defects in which it goes through different states throughout its life. The life cycle begins with a new defect discovered by a

tester while testing the application. It continues until the tester discovers a specific solution and closes the bug, so it does not reoccur.

Q.31 EXPLAIN DIFFERENCE BETWEEN FUNCTIONAL TESTING AND NON FUNCTIONAL TESTING.

FUNCTIONAL TESTING	NON FUNCTIONAL TESTING
• Functional testing is performed using the functional • specification provided by the client and verifies the • system against the functional requirements.	• Non-Functional testingchecksthe Performance, • reliability, scalability and other non-functional • aspects of the software system
• Easy to do manual testing	• Tough to do manual testing
• Manual testing or automation tools can be used for functional testing	• Using tools will be effective for this testing
• Functional testing is executed first	• Non functional testing should be performed after functional 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	• Nonfunctional testing describes how good the product works
• 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

Q.32 To create HLR & Testcase
(1) Instagram & Facebook only first page.
(2) Facebook Login page: http://www.facebook.com/

Ans:-

HLR

1(1). Instagram First Page	Click here
1(2). Facebook First Page	Click here
2 . Facebook Login Page : http://www.facebook.com/	Click here

SOFTWARE TESTING ASSIGNMENT MODULE 2

TEST CASE

1(1). Instagram First Page	**Click here**
1(2). Facebook First Page	**Click here**
2 . Facebook Login Page : http://www.facebook.com/	**Click here**

Q.33 DIFFERNCE BETWEEN STLC (SOFTWARE TESTING LIFE CYCLE) AND SDLC (SOFTWARE DEVELOPMENT LIFE CYCLE)

STLC	SDLC
• STLC is mainly related to software testing.	• SDLC is mainly related to software development.
• It focuses only on testing the software.	• Besides development other phases like testing is also included.
• STLC involves only five phases or steps.	• SDLC involves total six phases or steps.
• In STLC, less number of members (testers) are needed.	• In SDLC, more number of members (developers) are required for the whole process.
• In STLC, testing team(Test Lead or Test Architect) • makes the plans and designs.	• In SDLC, development team makes the plans and designs based on the requirements.
• Goal of STLC is to complete successful testing of software.	• Goal of SDLC is to complete successful development of software.
• It helps in making the software defects free.	• It helps in developing good quality software.
• STLC phases are performed after SDLC phases.	• SDLC phases are completed before the STLC phases.
• Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.	• Post deployment support , enhancement , and update are to be included if necessary.
• A tested software system is the end result of STLC.	• Creation of reusable software systems is the end result of SDLC.

Q.34 What is the difference between test scenarios, test cases, and test script?

Ans:-

TEST SCENARIO	TEST CASE	TEST SCRIPT
i) A Test scenario is any functionality that cab be tested. ii) Test scenario provides a small description, mostly one-line statements. iii) It is also called Test condition or Test possibility iv) It focus on "what to be tested". v) Test scenario is derived from Use case and SRS. vi) It requires fewer resources to write test scenario.	i) Test case is a detail document that describes step by step process to execute a test. ii) Test cases are more detailed with number of parameters iii) It focus on "How to be tested". iv) Test cases are derived from Test scenario. v) It requires more resources for documentation and execution.	i) A Test script is a set of instructions that is performed on a system under test to verify that the system performs as expected. ii) It is a line-by-line description that contain information about system functions that must be perform to verify an application or system under test. iii) It is used for automation testing that aims to validate the functionality of the software.

Q .35 Explain what TEST PLAN is ? what is the information should be covered.

Ans :-

A Test plan is a document that describes the strategy and objectives for a testing software product or system. It usually includes information such as the schedule, scope, resources, criteria and risks of the testing process. The Test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager. It guides the testing efforts and helps to find and resolve errors.

Creating a test plan is an important step in software development as it helps to ensure that the software products are thoroughly tested before they are released to the customers.

Test plan should cover the following information:

i) The roles and responsibilities of the test team and stakeholders.
ii) The general timelines and schedules for testing activities
iii) The levels and types of testing to be performed
iv) The test coverage, methods and responsibilities
v) The test environment and tools
vi) The test criteria and metrics
vii) The test deliverables and reports

Q.36 What is priority?

Ans : -

Priority is defined as the order in which the defects should be resolved. The priority status is usually set by the testing team while raising the defect against the dev team mentioning the timeframe to fix the defect. The Priority status is set based on end users requirement.

Q.37 What is Saverity?

Ans :-

Severity means how severe it is affecting the functionality. It is totally related to the quality standard or devotion to standard. The severity type is defined by the Software Tester based on the written test cases and functionality. Severity is associated with quality standards or devotion to standard. It is related to technical aspect of the product & decides based on how bad the bug is for the system.

Q.38 Bug categories are…

Ans:-

Bugs are classified into different categories depending on their nature, impact on the user experience and severity. These categories include:

a) critical bugs
b) Minor bugs
c) Major bugs
d) Trivial bugs

Q. 39 Advantage of Bugzila .

Ans:-

The Advantages of Bugzilla are:

- it is an open-source widely used bug tracker;
- it is easy in usage and its user interface is understandable for people without technical knowledge;
- it easily integrates with test management instruments;
- it integrates with an e-mailing system;
- it automates documentation.

Q.40 What are the different Methodologies in Agile Development Model?

Ans:-

There are more than a dozen agile techniques that are in use. Scrum, Extreme Programming (XP), lean product development, Kanban, Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), and the Crystal family of methodologies are the most popular approaches. We'll go into detail about each of these types of agile methodologies in the discussion. Scrum and XP are the two most popular agile methodologies. The core concepts, people, activities, and deliverables of these two

strategies will be thoroughly covered. To learn more in detail, please go through the various agile courses listed in the KnowledgeHut Agile Management training.

Types of Agile Methodology: 8 Important Categories

1. Scrum

Scrum is a well-known agile framework that is simple to learn and lightweight, but like all agile techniques, it is challenging to master completely. The "Scrum framework" contains a collection of processes, roles, events, artifacts, and rules that are intended to direct the team as they carry out the project.

- Scrum Pillars and Values

The theory behind Scrum is based on the three pillars

- Transparency,
- Inspection
- Adaptation.

In addition to the three pillars,

- Scrum also recognizes five fundamental values-
- Focus,
- Courage,
- Openness,
- Commitment, and
- Respect.

The major concepts and elements of the scrum are discussed in this section, as well as the responsibilities of a scrum team, the activities that take place during a sprint, and the sprint deliverables or artifacts.

2. Extreme Programming (XP)

Extreme Programming, sometimes known as "XP" or "eXtreme Programming," is an agile approach that is concentrated on software development. Scrum prioritizes work at the project management level and solicits feedback, whereas XP concentrates on best practices for software development.

XP Core Values

The core values of XP are

- Simplicity
- Communication
- Feedback
- Courage
- Respect

To organize their releases and iterations, XP teams employ "user stories," which are brief requirements. Developers write code in pairs throughout these iterations, which are typically two weeks long. "Architectural spikes" are iterations used to demonstrate a technology method. "Spikes" are periods of effort conducted to eliminate threats and issues. The release planning procedures incorporate the spikes.

3. Lean Product Development

Lean is not an agile methodology in the strictest sense, but as we'll see, the lean approach is quite like agile. Lean has its roots in the Toyota Production System, which was created to enhance Henry Ford's mass production method for making automobiles. Lean started as a manufacturing strategy, was subsequently extended to software development, and finally expanded to encompass many other types of knowledge work. When discussing lean in an agile setting, we are referring to a branch of lean known as "lean product development":

Lean product development focuses on creating new and improved goods, as opposed to the initial lean production techniques, which focused on manufacturing items. The following are some of the major tenets of lean product development:

Using visual management tools

Identifying customer-defined value

Building in learning and continuous improvement

Lean Core Concepts

Lean focuses on seven core concepts, as given below:

4. Feature-Driven Development (FDD)

FDD (Feature-Driven Development) is an easy-to-understand but effective method for creating products or solutions. A project team using the FDD methodology will first create a general model of the product, create a list of features, then schedule the work. To develop the features, the team then goes through the design and construction stages.

FDD suggests a collection of best practices that were inspired by software engineering. Among them are

5. Dynamic Systems Development Method (DSDM)

One of the first agile methodologies was called DSDM, and it had a very precise and prescriptive beginning. The project life cycle is thoroughly covered by us, which includes everything from the business case and feasibility to implementation in an agile project. The DSDM life cycle is depicted in the picture below.

There are eight guiding concepts for DSDM. Even though the Agile Manifesto was not yet established when these principles were developed, they are very similar to it.

The eight principles are:

Focus on the business need

Deliver on time

Collaborate

Never compromise quality

Build incrementally from firm foundations

Develop iteratively

Communicate continuously and clearly

Demonstrate control

By promoting early architectural considerations, agile appropriateness criteria, and agile contracts, DSDM has influenced the growth of agile.

6. Kanban

Kanban is a well-known Lean workflow management methodology for designing, monitoring, and enhancing information work delivery offerings. It enables you to visualize your job, increase productivity, and continuously get better. Taiichi Ohno, an engineer at Toyota, introduced "just in time" manufacturing to its supply chain in the late 1940s. Originating in the manufacturing industry Major players in the software sector rapidly saw how Kanban could improve the delivery of goods and services at the start of the twenty-first century.

7. Crystal

Crystal Crystal isn't just one technique; it's a family of tailored, situation-specific approaches that are color-coded. Crystal can handle a wide range of projects, from a small team developing a low-criticality system (Crystal Clear) to a large team developing a high-criticality system, thanks to the customization of each technique by criticality and team size (Crystal Magenta).

8. SAFe

The top business agility framework in the world is SAFe for Lean Enterprises. SAFe combines the strength of Lean, Agile, and DevOps into an all-

encompassing operating system that aids businesses in thriving in the digital age by producing cutting-edge goods and services more quickly, predictably, and of higher quality. SAFe's flexibility and scalability allow any firm to customize the framework to meet its business requirements. SAFe offers a complete range of solutions, from those requiring a few teams to those sophisticated systems requiring hundreds—and even thousands—of people to create and deliver, with four preconfigured configurations.

Q.41 Explain the difference between Authorization and Authentication in Web testing.What are the common problems faced in Web testing?

Ans:-

Authentication vs. Authorization

Authentication	Authorization
1. Authentication verifies who the user is.	1. Authorization determines what resources a user can access.
2. Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.	2. Authorization works through settings that are implemented and maintained by the organization.
3. Authentication is the first step of a good identity and access management process.	3. Authorization always takes place after authentication.
4. Authentication is visible to and partially changeable by the user.	4. Authorization isn't visible to or changeable by the user.
5. Example: By verifying their identity, employees can gain access to a human resources (HR) application that includes their personal pay information, vacation time, and 401K data.	5. Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization.

Common problems faced in Web testing :-

1. Ensuring cross browser compatibility

One of the major challenges in web application testing is achieving cross browser compatibility. Web applications need to function correctly across different web browsers such as Chrome, Firefox, Safari, and Microsoft Edge. When performing cross browser testing, QA testers may identify bugs related to layout inconsistencies, broken functionality, or JavaScript errors specific to certain browsers.

2. Dealing with dynamic content

Web applications often retrieve data from a server or database and display it dynamically on the user interface. In some cases, bugs may occur when the displayed data is inconsistent or not in sync with the actual data. This can happen due to caching issues, incorrect data

retrieval or manipulation, or improper handling of real-time updates. Such bugs can lead to user confusion, incorrect functionality, or incorrect decision-making based on the displayed data. Testing such dynamic elements can be challenging as they can change frequently.

3. Performance and scalability testing

Web applications need to be responsive and perform well under different user loads. QA testers can uncover bugs related to slow page loading times, high server response times, memory leaks, or inefficient database queries that affect the overall performance of the application. Performance and scalability testing can be tricky.

4. Security testing

Web applications are susceptible to various security risks. QA engineers play a crucial role in identifying potential security vulnerabilities such as SQL injection, cross-site scripting (XSS), insecure direct object references, or insufficient access controls. By finding and reporting these bugs, they help ensure the application's security and protect user data.

5. Test data management

Test data management involves providing the necessary data for testing scenarios. A bug in this area could occur if the test data provided is incomplete, inaccurate, or inconsistent. For example, missing or incorrect data values, inconsistent data formats, or inadequate coverage of test scenarios. This can lead to false-positive or false-negative test results, incomplete test coverage, and inaccurate assessment of the application's functionality and performance.

6. Communication and collaboration

Without proper communication and collaboration among the stakeholders, developers, and software testers, there is a higher chance of misinterpreting the requirements. This can lead to problems where the implemented functionality does not match the intended behavior or specifications.

Q.42 To create HLR & Testcase of web based (Whatsapp web, Instagram)

1.Whatsapp web: http://web.whatsapp.com/

2.Instagram web: http://www.instagram.com/accounts/login/?hl=en

Ans :-

HLR

1.WHATSAPP WEB : http://web.whatsapp.com/	**Click here**
2.INSTAGRAM WEB: http://www.instagram.com/accounts/login/?hl=en	**Click here**

Test case

1.WHATSAPP WEB : http://web.whatsapp.com/	**Click here**
2.INSTAGRAM WEB: http://www.instagram.com/accounts/login/?hl=en	**Click here**

Q.43 TO To create HLR & Testcase this link. https://artoftesting.com/

Ans :-

HLR

Link : https://artoftesting.com/	**Click here**

Test Case

Link : https://artoftesting.com/	**Click here**

Q.44 Write a scenario of only Whatsapp chat messages.

Ans :-

1. Whatsapp should connect with truecaller application, so if there any unknown contact want to message they can identify by there truecaller id.
2. Whatsapp should have FAVOURITE category which have contacts, with whom we chat repeatedely.
3. Whatsapp should have facility of separate massages of advertisements or brandings of any products from regular chat box.
4. Whatsapp should have reminder option for special occations or birthdays.
5. Whatsapp should have wallpaper's like or dislike option.
6. Whatsapp should have options other then 'ARCHIEVE CHAT' of chat hiding so anything personal chats are not displayed with the main chat.
7. Whatsapp should have SHARE MEDIA with the criteria of RESTRICTED or NOT RESTRICTED.
8. Whatsapp should follow all the GOVERNMENT REGULATIONS.
9. Whatsapp should have connection with INSTAGRAM OR FACEBOOK.
10. Whatsapp should have the options like GOOGLE TRANSLETER so if anybody sent messages with the language which is can't able to understand so it can help to explain the message.
11. Whatsapp Should be able to see the message of the sender with same time and date as he has sent on.
12. Whatsapp Should be able to use this function in saved conatcs and unsaved contacts.
13.. Whatsapp Should be able to shares emojis and gifs as well of the sender.
14.. Whatsapp Should be able to send special messages which change after sending in certain groups (e.g. Color change).
15. Whatsapp should check whether you can share any message to any sender.
16. Whatsapp should check if you can add more msgs in the same chat window of the sender with his msg.
17. Whatsapp check if this fuction does not work in broadcast msg
18. Whatsapp have check if any status can be send in reply private function.
19. Whatsapp have to check how old msgs can be send in this fuction.
20. Whatsapp should check if someone has uninstalled and installed again but still is able to see the msg he has sent in past.

Q.45 Write a Scenario of Pen.

Ans.

1. It should manufacture with fingergrip.
2. It should be with point with smooth writing.

3. The Point should not open easily because if its opening easily so ink can be spread easily.
4. The cap of every pen should have shirt holder clip on them.
5. The cap of pen have fit properly with pen if it is not fiting properly then pen's point can be harmful for hand or finger and it can do unwanted lines.
6. Tictac pen should have better quality of spring so button can work easily.
7. Tictac pen should have small lock for button for child's safety.
8. Gel pen should have proper wide point or nob so gel can easily write anything.
9. The brand name of pen should embosed or may properly print on the pen so it can easily see which brand's pen it is.
10. The pen should have ink eraser option also
11. The pen's ink should able to see user so if it is nearby to end, the user can refill the ink on time.
12. The pen's width should be ideal for every user's hand.
13. The pen's manufacturer company should test the pen on every type of paper, so they can know about pen's performance.
14. The Pen which is produce as a 'USE AND THROW it should have a reasonable price.
15. The pen should have any instructive icon or letter on pen for which colour's ink in the pen
16. The pen should have small digital display on which is working on button cell , it can show time or date.
17. The ink point pen which is used to write cursive writing it have proper ink tank so it can refill proper and not spread everywhere.
18. The ink point pen's company should sell refill ink properly for the same product so already written documents and after refilling the pen's writing match properly, if there is any different in the ink colour so the documents are not look good.

Q.45 Write a Scenario of Pen Stand.

Ans:-

1. The size of pen stand should be proper to put pen easily and should not easily fall down
2. The shape of pen stand should proper for holding more pieces of pen
3. The Body materials of pen stand should not easily crackable.
4. The colour of pen stand colour should decent, not too colourful for any corporate firm.
5. The pen stand should have digital watch.
6. The pen stand should have small calendar holding space
7. The Pen stand should have small categories partition for pen, pencil and markers
8. The pen stand should have branding option for cafes, restaurents and hotels.
9. The pen stand should have space for sticky notes
10. The pen stand should have well designed and with good quality.
11. The pen stand should have Proper visiblable height, Pen should recognised easily for which one to pick

Q.46 Write a scenario of Door

Ans :-

1. The door should with good quality material it manufacture with wood or glass, or any other material.
2. The Door should have different types like leading door, washroom door any other location door.

3. The door is built with glass; it should be a transparent or non-transparent door.
4. The door should have sign for push or pull.
5. There should different type of door like single door or a folded door.
6. The door signs for open on which side, whether inside or outside.
7. The size of the door should per the specification document
8. The door should made of the same material mentioned in the specification document.
9. The door's color should the same as the specification document.
10. The functionality of the door, whether that is a sliding or rotating door should sign on the door
11. The design of the door should as per the specification document
12. The position of door should proper in room.
13. The qulity accessories stoppers and other accessoies should present in the door
14. The locks should have a good quality for the door
15. The door should automatic close functionality in it and and it should represents the spring mechanism
16. The door should a stopper at the correct location.
17. The door shouldn't makes any sounds while opening or closing

Q.47 Write a scenario of ATM

Ans:-

1. The card reader of ATM should working correctly. A screen should ask you to insert the pin after inserting the valid card.
2. The cash dispenser should working as properly.
3. The ATM's receipt printer should working correctly. This means it sould print the data on the paper and the paper should comes out properly.
4. The Screen buttons should working correctly.
5. For touch screen: The display should operational and working as per the expectations.
6. The text on the screen button should visible clearly.
7. The ATM's number button on the Keypad should working properly.
8. The functionality of the cancel button on the keypad should working properly.
9. The keypad and numbers should be visible clearly.
10. The text color and font of the data on the screen should be able to read it clearly for users.
11. The language selection option should be available in the ATM and the messages or data should displayed in the selected language.
12. The receipt printing functionality after a valid transaction should working properly.
13. The system takes time to log out should be proper.
14. The timeout session functionality should available on all ATM.
15. The deposit slot functionality capability should displayed properly.
16. The different bank's cards should accessable.
17. After adding incorrect PIN sending the message service should be available.
18. When there is no cash in the it should notify on the screen of ATM.
19. There should be a service available for messages after a transaction.

Q.48 When to use Usabilty Testing?

Ans:-

Usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you've begun the strategy work around a brand new site or app. This will quickly identify areas for opportunity, and reduce the amount of

assumptions your design team will make with regard to what the user wants. Additionally, after the usability tests analysis, the team should have the ability to pinpoint the steps needed to achieve the project goals with as little disruption as possible. Don't assume that a system is completely broken when beginning a project. Most likely designers, developers, researchers, content strategists, etc. have already spent a lot of time building what you see before you. Rather than assuming that the efforts of previous teams were completely misguided, identify particular areas where design, testing, and validation can be conducted in order to enhance and correct the product. Ultimately, this will assist in limiting the scope of work.

Q.49 What is the procedure for GUI Testing ?

Ans:-

1) Verify the working of carousel arrows and way finders
2) Verify that the password field is accepting values only in a masked state
3) Verify that the 'save' button remains inactive until all required fields are entered
4) Verify that the user is allowed to navigate to the top of the page using 'Top' bar
5) Verify that proper message is displayed when the applied filters do not retrieve any results
6) Verify the navigation from links available in Headers and footers
7) Verify the alignment of radio buttons is accurate
8) Verify that multiple options in checkboxes can be selected at a time
9) Verify that the title of each section is in bold letters
10) Verify the color change of hyperlinks on clicking

Q.50 Write a scenario of Microwave oven?

Ans : -

1) The structure and dimensions of the microwave should conforms to the specified dimensions mentioned in the user manual.
2) The oven door should fits squarely and securely and opens and closes smoothly.
3) The door hinges should in proper condition and if the door's surface should not damaged.
4) There should no corrosion on the door, the door hinges, or the oven interior.
5) After Plugging the power terminal LED should glows And the oven should getting power supply properly.
6) The microwave should performs under different voltage conditions, it should able/unable to sustain fluctuating voltage.
7) By Switching ON the oven for a very long time (up to 12-24 hours continuously), thermostat should work good under such a situation.
8) The timer should works properly, if the microwave switches off automatically after the preset time.
9) Putting oven door open and it should cannot set a timer for cooking.
10) After you switch off the microwave before the timer, it should counted down to the present time.
11) The Cooking time for different food items should different in the microwave oven.
12) The bowls should made up from glass and plastics so it shouldn't heated when used in the oven.
13) The parts of frozen food remain frozen if insufficient time should allowed for the heating process.
14) The microwaves should not heat the oven walls and other non-metallic cooking utensils directly.

15) The microwave exposure causes serious health effects such as deep tissue burns and hyperthermia. This Warning should on the microwave oven.

16) The safety interlock should switches stop the generation of microwaves immediately after the door is opened.

Q.50 Write a scenario of Coffee vending machine?

Ans:-

1. The dimension of the coffee machine should as per the specification.
2. The outer body, as well as inner part's material, should as per the specification.
3. the machine's body color as well brand should correctly visible.
4. The input mechanism for coffee ingredients-milk, water, coffee beans/powder should accepting the ingredients frequently.
5. The quantity of hot water, milk, coffee powder per serving should written on machine or in the product user guide.
6. The power/voltage requirements of the machine should fulfil properly.
7. The machine should stop in that situation of power resumption, the remaining coffee should not get come out of the nozzle.
8. The coffee should not leak when not in operation.
9. The amount of coffee served in single-serving should as per specification.
10. The digital display should displays correct information.
11. The machine should be switched on and off using the power buttons.
12. The indicator lights should working when the machine is switched on-off.
13. The functioning of all the buttons should work properly when press.
14. The each button should have an image/text with it, indicating the task it performs.
15. The complete quantity of coffee should get poured in a single operation, no residual coffee should be present in the nozzle.
16. The coffee served has the same and correct temperature each time it should served by the machine.
17. The system should display an error when it runs out of ingredients.
18. Pressing the coffee button multiple times should leads to multiple serving of coffee.
19. The machine should work correctly in different climatic, moistures and temperature conditions.
20. Machine should not make too much sound when in operation.

Q.51 Write a scenario of Chair?

Ans:-

1. The chair should stable enough to take an average human load
2. The material used in making the chair-wood, plastic or any other mattle it should have quality.
3. The chair's leg should level to the floor
4. The different type of chairs for different usability Which purpose we have buy chair we should we should know that very well.
5. The chair should have back support proper.
6. The support for hands should have also.
7. The chair's material should not brackeble easily in plastic chairs.
8. The cushion chair should good have quality in the office chair.
9. We should know how to wash with water and is there is any effect from that.
10. The dimension of chair should as per the specifications
11. The height of the chair's seat from floor should ideal
12. The chair should not slippery on different kinds of surfaces like floor, glass, wood, Grass.
13. The branding should properly printed of company.

Q.52 To Create Scenario (Positive & Negative)

Ans:-

1.Gmail (Receiving Mail)

Positive :-

1. The new emails should received properly in the inbox
2. The newly received email should correctly displayed sender emailId or name, mail subject and mail body(trimmed to single line)
3. Clicking the newly received email, user should navigated to email content.
4. The email contents should correctly displayed with the desired source formatting
5. Any attachments are attached to the email and should downloadable.
6. The attachments should scanned for viruses before download.
7. All the emails marked as read should not highlighted.
8. All the emails read as well as unread should a mail read timeappended at the end on the email list displayed in the inbox section.
9. The unread email count should increase by one on receiving a new email.
10. The unread email count should decrease by one on reading an email (marking email as read).
11. The email recipients in cc should visible to all user.
12. The email should be received from non-gmail emailIds like -yahoo, hotmail etc.

NEGETIVE

1. email recipients in bcc should not visible to user.
2. All received emails should piled up in the 'Inbox' sectionand and should deleted in cyclic fashion based on the size availability.
3. The count of unread emails should displayed alongside 'Inbox' text inleft sidebar of GMail.

2. Online shopping to buy product (flipkart)

Ans:-

Positive

1. The product page, the user should select the desired attribute of the product e.g. size, color, etc.
2. The user should add to the cart one or more products.
3. The users should add products to the wishlist.
4. The user could see the previously added products on the cart page, after signing in to the application.
5. The user should successfully buy more than one products that were added to his/her cart.
6. The limit to the number of products a user can buy should working correctly. Also, an error message should display, prevent the user from buying more than the limit.
7. The delivery should be declined during checkout for the places where shipping is not available.
8. The Cash on Delivery option of payment should working fine.
9. The different prepaid methods of payments should working fine.
10. The product return functionality should works correctly.
11. The each of the payment options should selectable.
12. Check if the default credit/debit card gets automatically added.
13. the multiple cards should be saved as default or not.
14. The correct currency should reflect on the page.

15. The payment should not getting processed for null values in the cart.

NEGETIVE

1. The multiple payment options should not get select.
2. The payment should not getting proceeded with an expired/blocked card.
3. The page should not proceed to the payment page before all the mandatory information is filled.
4. The user should not add more than the available inventory of the product.
5. The payment should not getting processed for null values in the cart.
6. A pop-up should appears if the session has expired.
7. The user should get information about unsuccessful payments.
8. The double payment should not occurring in any case.

Q .53 Write a Scenario of Wrist Watch.

Ans:-

1. The watch needle should working fine all the needle like hours needle, Minutes needle and second needle.
2. The positioning of each needle should get change.
3. The user should able to change or set the time manually.
4. The date in the wristwatch should change automatically after 24 hrs.
5. The month should change automatically after 30 0r 31 days.
6. The year should change after 12 month.
7. The watch should shows the correct time on the basis of region.
8. The user should able to set the time or change the time on a watch.
9. The user should also able to change the day on watch if it is available.
10. The parts of wrist watch should properly fitted or not.
11. The Date, Time and other information in a wristwatch should properly visible to the user.
12. The watch should properly fits on the wrist.
13. The design of wrist watch should as per requirement.
14. The watch should waterproof.
15. The colour, width, dial, and length of wrist watch should as per CRS.
16. The materials used for the wrist watch body should as per requirement.
17. The material should have quality, it should manufacture wrist watch strap -Plastic, leather .etc
18. The wrist watch weight should as per requirement.
19. The Logo and name of the company should display properly on the watch.
20. The functionality of the button of the watch should working fine.

Q.54 Write a Scenario of Lift(Elevator)

ANS:-

1. The height, width, and life volume should as per the requirement.
2. The buttons for the closing lift, opening lift, fan, emergency, and all floor numbers should be there on the button panel
3. The presence of a display where the floor number appears should be available in the lift
4. The floor number should beannounced on each floor
5. The light and aroma, along with some instrumental music should be available in lift.
6. The maximum number of people and weight also emergency instructions should written on the wall.

7. The up and down buttons should outside the lift.
8. The present sensors should working properly.
9. The lift should move up and down as we instruct it.
10. The lift should capable of stopping on floors that are pressed on the button panel
11. The lift closes when the close button is placed or should close once after a few times as per the manual time
12. The fan should working by pressing the respective button
13. When the lift is going to close, someone has just put any object or body parts between the lift doors it should open automatically.
14. When the weight exceeds from standard, then it should give some alert message or some alert sound should ring
15. The light and fan should running at power failure
16. The speed of the lift shouldn't be fast.
17. lift doors shound not be open when the lift is moving down or upward
18. lift doors should close when no one is using it

Q.55 Write a Scenario of whatsapp Group (generate group)

Ans:

1. The user should create a Whatsapp group.
2. The user should set a name for the created group.
3. The user should add and save the group description.
4. The user should make multiple people as group Admin.
5. The only group admins should add people to the group.
6. The any option to mute group notifications for some time, the functionality should work properly
7. The option should have to add people to the group by sharing a link.
8. The admin should delete users from the group.
9. The admin should change settings like only admin can share information in this group, or everyone can share information in this group.
10. If a person is removed from the group, he/she should not be able to see any updates.
11. The user should exit the group by clicking on the exit button.
12. The admin should delete the group.
13. You should able to create a group without adding any member.
14. Every person should able to share information and every person could see the messages of group
15. people should share information in different media like photo, video, document, link, simple text.

Q.56 Write a Scenario of Whatsapp payment.

Ans:-

1. The payment gateway should connect with the bank systems.
2. The payment gateway should allow users to enter their payment details securely.
3. The users should initiate payments using valid credentials.
4. The payment gateway should process the payments accurately.
5. The payment gateway should respond to invalid credentials correctly.
6. The payment gateway should capture payment details accurately.
7. The payment gateway should return the correct response codes for successful and unsuccessful payments.
8. The ability of the payment gateway should handle multiple simultaneous payments.

9. The payment gateway should integrated properly with the merchant system.
10. financial data should not store after each payment processing session.