Assignment [Module 1]

Question 1:

What is Exploratory Testing?

Answer:

Exploratory testing is a software testing technique that does not use any specific test design, plan or approach.

It is a software testing technique in which the testers explore and identify different means of evaluating and improving the quality of the software.

Exploratory testing is an unscripted approach to software testing, where the tester is free to select any possible methodology to test the software. Exploratory testing is a common practice by software developers that utilize their personal skills and abilities to test the software they developed and/or coded.

Question 2:

What is traceability matrix?

Answer

Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability.

Traceability can be horizontal through all the test documentation for a given test level (e.g. system testing, from test conditions through test cases to test scripts) or it can be vertical through the layers of development documentation (e.g. from requirements to components).

Question 3:

What is Boundary value testing?

Answer

Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges Boundary value analysis is a method which refines equivalence partitioning. Boundary value analysis generates test cases that highlight errors better than equivalence partitioning.

Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design

Question 4:

What is Equivalence partitioning testing?

Answer:

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.

Equivalence partitioning is the process of defining the optimum number of tests by:

Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function,

Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.

If we want to test the following IF statement: “If value is between 1 and 100 (inclusive) (e.g value >=1 and value <=100) Then...” We could put a range of numbers as shown in the below figure.

Question 5:

What is Integration testing?

Answer:

Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

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

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

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems.

Integration testing is done by a specific integration tester or test team. Components may be code modules, operating systems, hardware and even complete systems.

Question 6:

What determines the level of risk?

Answer

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

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

When testing does find defects, the Quality of the software system increases when those defects are fixed.

The Quality of systems can be improved through Lessons learned from previous projects Analysis of root causes of defects found in other projects can lead to Process Improvement.

Process Improvement can prevent those defects reoccurring Which in turn, can improve the Quality of future systems Testing should be integrated as one of the Quality assurance activities.

Question 7:

What is Alpha testing?

Answer:

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 the form of Acceptance Testing.

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

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

Question 8:

What is beta testing?

Answer:

It is always performed by the customers at their own site.

It is not performed by Independent Testing Team.

Beta Testing is always open to the market and public.

It is 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 (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.

It is only a kind of Black Box Testing.

Question 9:

What is component testing?

Answer:

Component Testing – The testing of individual software components.

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

The purpose is to validate that each unit of the software performs as designed.

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

Sometimes known as Unit Testing, Module Testing or Program Testing.

Question 10:

What is functional system testing?

Answer:

It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification.

In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value.

Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting.

It is done to verify the functionality of the application.

Question 11:

What is Non-Functional Testing?

Answer:

Non-functional testing is a type of software testing to test non-functional parameters such as reliability, load test, performance and accountability of the software.

The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters.

The parameters of non-functional testing are never tested before the functional testing.

Question 12:

What is GUI Testing?

Answer:

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.

Question 13:

What is Adhoc testing?

Answer :

Adhoc testing is a type of software testing that is performed without a predetermined test plan or script. This method involves exploring or testing features spontaneously as the tester perceives potential issues or areas that require further testing.

Here are some of the reasons why ad hoc testing is performed:

1. To find unexpected bugs and errors.
2. To explore the software and learn more about its functionality.
3. To get feedback from user.
4. To improve the test coverage.
5. To be more efficient and flexible.

Question 14:

What is load testing?

Answer:

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 :

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

Question 15:

What is stress Testing?

Answer:

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.

Question 16:

What is white box testing and list the types of white box testing?

Answer:

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.

Types of white box testing:

1. Unit testing
2. Static analysis
3. Dynamic analysis
4. Statement coverage
5. Branch testing
6. Path testing
7. Loop testing

Question 17:

What is black box testing? What are the different black box testing techniques?

Answer:

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.

Types of black box testing techniques:

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing
5. Use-case testing

Question 18:

Mention what are the categories of defects?

Answer:

Classification of defects:

1. Error of commission:

Commission means instruction or some kind of command given. The error of commission means the error made in command or instruction.

1. Errors of omission:

As name is already describing error of omission is something which happens accidently.

1. Error of clarity:

This error happens due to misunderstanding between the developer and client. It travels most of the time from the requirements to the software.

1. Error of speed or capacity:

The name of the error itself enough I think to tell about it this error. Your software is working fine but not working in the required time this is the error of speed. When it comes to capacity it can be relevant to memory.

Question 19:

Mention what bigbang testing is?

Answer:

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

Big Bang testing has the advantage that everything is finished before integration testing starts.

Question 20:

What is the purpose of exit criteria?

Answer:

The purpose of exit criteria is to have a **“control mechanism”** to verify that a requirement has been met and the service, product, or process can move to the next step.

Question 21:

When should "Regression Testing" be performed?

Answer:

Whenever a new feature is developed, when an existing feature is improved or if any UI updates are made, ideally, there is a need to perform software regression and functional testing. It will ensure that the changes do not impact the software’s existing functionality.

In addition, the verification method should be performed whether there is a small or significant change in an application that might affect any change to its old functionality.

Functional regression testing teams must verify that the new code does not conflict with older code, and also confirms that the code that has not been changed is still working as expected. Therefore, this type of **visual regression testing** ensures that the previous functionality of the application works effectively and new changes have not introduced any new bugs within the application.

Question 22:

What is 7 key principles? Explain in detail?

Answer:

1. key principle:

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.

2) Exhaustive testing is not possible

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.

This is very unlikely that the project timescales would allow for this number of tests.

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.

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’.

most defects found during testing are usually confined to a small number of modules.

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.

6) 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.

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.

Question 23

Difference between QA v/s QC v/s Teste• Difference between QA v/s QC v/s Teste

Answer:

he terms QA, QC, and Tester often overlap in the software industry, but they have distinct roles:

* **Quality Assurance (QA)** is a process-oriented approach that focuses on preventing defects in the development process. QA activities include process definition and implementation, training, audits, and process improvements.
* **Quality Control (QC)** is product-oriented and involves the actual testing of the software to identify defects. QC activities include executing test cases, identifying bugs, and verifying that the product meets the specified requirements.
* A **Tester** is typically involved in the QC process and is responsible for creating and executing test cases, reporting bugs, and ensuring that the software functions as expected.

In essence, QA is about ensuring quality processes, QC is about ensuring a quality product, and Testers are the professionals who carry out QC activities.

Question 24

Difference between Smoke and Sanity?

Answer:

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

Question 25:

Difference between verification and Validation

Answer:

**Verification testing is a process used in software development to ensure two things. One is that a product or system meets its specified requirements. Another is that the product or system adheres to the design and development standards.**

**The verification process is typically performed throughout the software development cycle. It involves various testing techniques and tools. These tests ensure that each component of the software functions correctly and meets the expected requirements.**

**Validation testing involves various testing techniques and tools designed to simulate real-world scenarios. Then, you compare the results to the requirements and specifications. It will ensure that the software meets all necessary criteria.**

**Any defects or issues identified during validation testing are addressed. The software also gets retested until it meets all requirements.**

Question 26:

Explain types of Performance testing.

Answer:

Types of performance testing

Load testing:

Load testing is a type of testing which involves evaluating the performance of the system under the expected workload. A typical load test includes determining the response time, throughput, error rate, etc during the course of the load test.

Stress testing:

Stress testing is a type of performance testing where we evaluate the application’s performance at a load much higher than the expected load. Another aspect of the stress testing is to determine the break-point of the application, the point at which the application fails to respond in the correct manner.

Endurance testing:

Endurance testing is also known as ‘Soak Testing’. It is done to determine if the system can sustain the continuous expected load for a long duration. Issues like memory leakage are found with endurance testing.

Spike testing:

In spike testing, we analyze the behavior of the system on suddenly increasing the number of users. It also involves checking if the application is able to recover after the sudden burst of users.

Volume testing:

The volume testing is performed by feeding the application with a high volume of data. The application can be tested with a large amount of data inserted in the database or by providing a large file to the application for processing. Using volume testing, we can identify the bottleneck in the application with a high volume of data.

Question 27:

What is Error, Defect, Bug and failure?

Answer:

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

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

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

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

Failure: The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

Question 28:

Difference between Priority and Severity

Answer:

Difference between priority and severity

The term severity defines, to what degree the system is impacted. Whereas, priority is all about scheduling or urgency.

Usually, it is the test engineer who determines the severity. While the product owners decide the priorities of defects.

It is very unlikely that the severity might change. Whereas, the priorities change from time to time.

The severity is usually determined in terms of a technical point of view. Whereas, priority depends upon the user experience.

The severity affects the technical working of the system. Whereas the latter affects business.

Question 29:

What is Bug Life Cycle?

Answer:

A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program’s source code or its design.

The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as Defect Life Cycle.

When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and closed.

The process by which the defect moves through the life cycle is depicted next slide.

Question 30:

Explain the difference between Functional testing and Non Functional testing

Answer:

|  |  |
| --- | --- |
| 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 testing checks the performance, reliability, scalability and other non-functional aspects of the software system |
| Functional testing is executed first | Non functional testing should be performed after functional testing |
| Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| Business requirements are the inputs to functional testing | Performance parameters like speed, scalability are inputs for non-functional testing |
| Functional testing describes what the product does | Non functional testing describes how good the product works |
| Easy to do manual testing | Tough to do manual testing |
| Types of functional testing are:   * Unit testing * Smoke testing * Sanity testing * Integration testing * White box testing * Black box testing * User acceptance testing * Regression testing | Types of non functional testing are:   * Performance testing * Load testing * Volume testing * Stress testing * Security testing * Installation testing * Penetration testing * Compatibility testing * Migration |

Question 31:

To create HLR & TestCase of 1)(Instagram , Facebook) only first page

2) Facebook Login Page : <https://www.facebook.com>

Question 32:

What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

Answer:

|  |  |
| --- | --- |
| SDLC(Software development life cycle) | STLC(Software testing life cycle) |
| SDLC is mainly related to software development | STLC is mainly related to software testing |
| Besides development other phases like testing is also included | It focuses only on testing the software |
| SDLC involves total six phases or steps | STLC involves only five phases or steps |
| In SDLC, more number of members (developers) are required for whole process | In STLC, less number of members(testers) are needed |
| In SDLC, development team makes the plans and designs based on the requirements | In STLC, testing team(Test Load or Test Architect) makes the plans and designs |
| Goal of SDLC is to complete successful development of the software | Goal of STLC is to complete successful testing of the software |
| SDLC phases are completed before the STLC phases | STLC phases are performed after SDLC phases |
| Creation of reusable software is the end result of SDLC | Tested software system is the end result of STLC |

Question 33:

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

Answer:

|  |  |  |
| --- | --- | --- |
| Test Scenario | Test Case | Test Script |
| Is any functionality that can be tested | Is a set of actions executed to verify particular feature or functionality | Is a set of instructions to test an app automatically |
| Is delivered from test artifacts like business requirements specification(BRS) and software requirement specification(SRS) | Is mostly derived from test scenario | Is mostly derived from test cases |
| Helps test the end to end functionality in an agile way | Helps in exhaustive testing of an app | Helps to test specific things repeatedly |
| Is more focused on what to test | Is focused on what to test and how to test | It focused on expected result |
| Allows quickly assessing the testing scope | Allows deleting errors and defects | Allows carrying out an automatic execution o test cases |

Question 34:

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

Answer: A test plan in detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources( human resources, software, and hardware), test estimation and test deliverables.

The test plan is a base of every software’s testing. It is the most crucial activity which ensures availability of all the lists of planned activities I an appropriate sequence.

The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager. The plan is prepared by the test load60%), test manager(20%), and by the test engineer(20%).

Question 35:

What is priority?

Answer:

Priority as the name suggests is about prioritizing a bug according to its severity. Priority signifies the importance or urgency of fixing a defect. It is associated with scheduling, to resolve a bug. The priority status of a defect is initially set by a tester to a developer to indicate the importance of fixing that defect. If the priority status is high then that bug will be fixed first followed by lower prior ones.

1. High Priority

High Priority defects are business-critical and have to be fixed with immediate effect. Here, Issue 1 "website fails to accept credit cards" is a defect that has a high impact on the revenue of the organization and so should be fixed immediately. So Issue 1 can be set as a High priority one.

1. Medium Priority

Medium-priority defects are those which have a moderate level of impact on business and have to be fixed in the current release itself. With respect to the above eg: Issue 2 "Order confirmation emails are not getting delivered to the consumer" should be reported as a Medium priority issue.

1. Low Priority

Low-priority defects are those which do not have any impact on the business and those which are cosmetic in nature. This can be pushed to the next release also. So, Issue 3 "Misalignment of submit button in a particular version of browser" should be reported as a Low priority one.

Question 36:

What is severity?

Answer:

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

Critical: The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.

Major (High): The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.

Moderate (Medium): The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.

Minor (Low): The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

Cosmetic: The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

Question 37:

Bug categories are…

Answer:

1. Functional bug:

This is a broad type of error that happens whenever software doesn’t behave as intended. For example, if the end user clicks the “Save” button, but their entered data isn’t saved, this is a functional error. After some investigation, a software tester may identify a more specific culprit behind the error and reclassify it as a different type of bug.

1. Syntax Error:

A syntax error occurs in the source code of a program and prevents the program from being properly compiled. This type of error is very common and typically occurs when there are one or more missing or incorrect characters in the code. For example, a single missing bracket could cause a syntax error.

1. Logic errors:

A logic error represents a mistake in the software flow and causes the software to behave incorrectly. This type of error can cause the program to produce an incorrect output, or even hang or crash. Unlike syntax errors, logic errors will not prevent a program from compiling.

1. Calculation Bugs:

Anytime software returns an incorrect value — whether it’s one the end user sees or one that’s passed to another program — that’s a calculation error.

1. Unit level bugs:

David LaVine, founder of [RocLogic Marketing](https://roclogicmarketing.com/" \t "_blank) and a former engineer, says unit-level software bugs are the most common. They’re also typically the easiest to fix.

1. System level integration bugs:

This type of bug occurs when two or more pieces of software from separate subsystems interact erroneously. Often the two sets of code are written by different developers. LaVine explains that even when there’s a solid set of requirements for developers to follow, there’s usually some level of interpretation required or details that get overlooked, causing the interaction between two pieces of software to fail.

1. Out of bounds bugs:

LaVine notes that these types of software bugs show up when the end user interacts with the software in ways that weren’t expected. This often occurs when the user sets a parameter outside the limits of intended use, such as entering a significantly larger or smaller number than coded for or inputting an unexpected data type, like text where a number should be.

Question 38:

Advantage of Bugzila.

Answer:

Advantages of Bugzilla:

* It is open source, widely used but tracker
* It is easy in usage and its user interface understandable for people without technical knowledge
* It easily integrates with the test management instruments
* It integrates with an E-mail system
* It automate documentation

Question 39:

Difference between priority and severity

Answer:

|  |  |
| --- | --- |
| Severity | Priority |
| Severity determines the defect’s effect on the application | Priority determines the defect urgency of repair |
| How bad the defect is | How soon we need to fix |
| Severity is given by QA Testers | Priority is given by test lead or project manager |
| Levels:   * Critical: the software will not run * High: unexpected errors * Medium: a feature is malfunctioning * A cosmetic issue | Levels:   * P1: fix before next build to test * P2: Fix before final release * P3: we probably will not get to these, but we want to track them anyway to resolve the priority-severity divide |

Question 40:

What are the different Methodologies in Agile Development Model?

Answer:

1. Agile scrum methodology:

Scrum is a lightweight framework of Agile Project Management, it can be adopted to conduct iterative and all types of incremental projects.

Due to its specific characteristics like simplicity, sustained productivity, and strength for blending several underlying approaches adopted by other agile methods, Scrum has obtained popularity over the years.

1. Lean

It is the iterative, agile methodology that directs the team on addressing customer values by compelling value stream mapping, although, it is a deeply adaptable, emerging methodology with the absence of solid guidelines, laws, or methods.

1. Kanban

Kanban is an eminently visual workflow management approach, famous amidst Lean teams, that can be employed for visualizing and thoroughly maintaining the making of products, it focuses on continual delivery of the product, but is not making stress to the entire software development life cycle.

1. Extreme Programming (XP):

Generally being used with Scrum, it can focus on how Agile can increase customer satisfaction, instead of delivering at the entirety, the customer seeks for the near future, it provides them what they demand at present.

1. Crystal:

In actual, the Crystal Methodology is a class of small agile methodologies that incorporate ***Crystal Clear, Crystal Yellow, Crystal Red and many more*.** Every methodology has its own specific framework. It is introduced by [Alistair Cock burn](http://alistair.cockburn.us/) while participated in writing the Agile manifesto for software development.

1. Dynamic System Development Method(DSDM):

DSDM gives a board work-frame that is outlined;

* To plan, handle, execute, and scale the complete process of software development,
* *It directs on the business-driven approach and*
* *It doesn’t negotiate on quality and timely delivery.*

1. Feature Driven Development(FDD):

Being an iterative and accumulative development process, FDD is a lightweight, agile methodology to develop software. It mixes various industry-established best practices into united as a whole. These practices are encouraged in terms of a feature-first perspective while aiming at creating values for the client.

Question 41:

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

Answer:

Difference between authorization and authentication:

|  |  |
| --- | --- |
| Authentication | Authorization |
| Authentication is the process of identifying a user to provide access to a system. | Authorization is the process of giving permission to access the resources. |
| It requires the logic details of the user, such as user name, password, etc | It requires user’s privilege or security level |
| Example: Entering login details is necessary for the employees to authenticate themselves to access the organizational emails or software | Example: After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles |
| Authentication credentials can be partially changed by the user as per the requirement. | Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner/manager of the system, and he can only change it. |

Common problems faced in Web Testing:

1. Integration:

* How would you rate any application?
* You would love to have the application’s functionalities, reliability and performance to be working amazingly and those things make the application rocking. I bet.
* We cannot compromise on any of the above three aspects of the web application. So, it requires integration testing as its result would define the overall applications’ performance at the user’s end.
* I am sure, no businesses can afford losing even a single customer.

1. Interoperability:

* Wow! I need to elaborate this.

While developing any application, we don’t know which device, operating system & browsers our customers will be using while browsing our web application. We call it responsive web application development.

* But, in software testing industry, they need to be tested for various functionalities to work on any environment, device and browser.
* Responsive web application is in #Trend these days!

1. Security:

* We all are aware of the cyber threats that can take place at any moment. Regular security testing is required to check for any cyber threat. Also, data integrity tests need to be performed if there are any chances of data loss. Testers take due time to perform security tests as they carefully deal with the unsecured data transfer.
* How about the risks based on Intranet-based applications and Internet-based applications?
* Intranet-based applications are bound to pre-defined usage of software and hardware required and they can be tested easily. But, when it comes to Internet-based applications, security measures are defined much more clearly by the skilled testers and QA team.

1. Performance:

* Application’s Page/Loading Speed matters a lot. And seriously, we make sure our customer’s application loads much faster as delay in a single second can divert the user and we don’t want that, period.
* Hardware testing, misunderstanding the required application’s features can surely lead to a breach of application performance. Integration and interoperability testing directly influence performance testing.

1. Usability:

* We all love scalable and interactive web applications. Whether into Windows-based or Mac-based devices, the testers definitely perform the usability testing of the web applications on the real browsers of the real devices.
* An important one, right?

1. Incoming and outgoing of the apps:

* Shopaholics can enter into an e-commerce website via signing in or directly on a product page. Also, in an online ticket booking application, a user may directly choose a movie/event to buy a ticket or maybe, he/she directly lands on the payment page via 3rd party websites.
* When your application has various entry and exit points, the application testers take deep care to test those points well.

1. Lower internet speed bandwidth:

* Low Internet speed is hated by you, me and everyone. So, we want applications to work on those devices as well where people don’t have high Internet Bandwidth.
* While considering testing Internet-based applications, with low Internet speed, we can imagine that various web components can take more time to load and that can affect other components as well. So, this is quite important to test.