Software Testing Assignment

Module 2

1. What is Exploratory Testing?

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

Exploratory testing is a concurrent process whare

- Test design, execution and logging happen simultaneously
- Testing is often not recorded
- Makes use of experience, heuristics and test patterns
- Testing is based on a test charter that may include

2. What is traceability matrix?

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

To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

<u>Pros of Traceability Matrix</u>

- Make obvious to the client that the software is being developed as per the requirements.
- Easy to identify the missing functionalities.

Cons of Traceability Matrix

- Poor or unknown test coverage, more defects found in production
- Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays.

3. What is boundary value testing?

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.
- At those points when input values change from valid to invalid errors are most likely to occur.

4. What is Equivalence partitioning testing?

The numbers fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition (EP) or Equivalence Class.

- If one value finds a bug, the others probably will too
- If one doesn't find a bug, the others probably won't either
- Plus there are 2 Invalid partitions
  - Valid partitioning
  - Invalid partitioning

5) What is Integration testing?

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

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.

- There are 2 levels of Integration Testing
  - Component Integration Testing
  - System Integration Testing

6) What determines the level of risk?

‘A factor that could 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
- Process Improvement can prevent those defects reoccurring
- Process Improvement can prevent those defects reoccurring
- Which in turn, can improve the Quality of future systems
- <u>Risks are of two types:-</u>
  - Project risks
  - Product risk

7) What is Alpha testing?

- Alpha testing is not open the market and public.
- Application and project
- Performed in virtual environment.
- It is the form of acceptance testing.
- White box testing and black box testing.

8) What is beta testing? (field)

- Beta testing is open the market and public.
- Performed in real time environment.
- It is the form of acceptance testing.
- It is only for black box testing.
- Beta testing “pre- release”
- Pilot Testing is testing to product on real world as well as collect data on the use of product in the classroom.
- It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area.

9) What is component testing?

Component testing The testing of individual software components.

- A unit is the smallest testable part of an application like functions/procedures, classes, interfaces.
- Unit testing is performed by using the White Box Testing method.
- Below we look at some of what extreme programming brings to the world of unit testing:
  - Tests are written before the code
  - Rely heavily on testing frameworks
  - All classes in the applications are tested
  - Quick and easy integration is made possible

10) What is functional system testing?

A requirement that specifies a function that a system or system component must perform

- A Requirement may exist as a text document and/or a model
- 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.

- **Functional testing mainly involves black box testing** and it is not concerned about the source code of the application.

11) What is Non-functional testing?

Non-Functional Testing: Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability

- May be performed at all Test levels (not just Non Functional Systems Testing)

- It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.

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

- Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling

12) What is GUI testing?

Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test.

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

- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.

- Check you can execute the intended functionality of the application using the GUI

- Check for Clear demarcation of different sections on screen

- Check the positioning of GUI elements for different screen resolution.

13) What is Adhoc testing?

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

- This is why an error guessing approach, used after more formal techniques have been applied to some extent, can be very effective.

- Using experience to postulate errors.

- Use Error Guessing to Complement Test Design Techniques.

- Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.

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

14) What is load testing?

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.

- Load testing gives confidence in the system & its reliability and performance.

- Load Testing helps identify the bottlenecks in the system under heavy user stress scenarios before they happen in a production environment.
- The maximum operating capacity of an application

15) What is stress testing?

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.

- Under Stress Testing, AUT is be stressed for a short period of time to know its withstanding capacity.

- Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.

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

White Box Testing: Testing based on an analysis of the internal structure of the component or system.

Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

- For example, a structural technique may be concerned with exercising loops in the software.

- Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.

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

- White box testing is the detailed investigation of internal logic and structure of the code.

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

❖ Black box techniques:-

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition testing
- Use-case testing
- Other black box testing

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

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

- The testers have no knowledge of how the system or component is structured inside the box.
- The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
- The tester is oblivious to the system architecture and does not have access to the source code.

- Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

18) What is 7 key principles? Explain in detail?

- Testing shows presence of Defects
- Exhaustive Testing is Impossible!
- Early Testing
- Defect Clustering
- The Pesticide Paradox
- Testing is Context Dependent
- Absence of Errors Fallacy

1. Testing shows presence of Defects

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

- However Testing **cannot prove** that there are **no** defects present

2.Exhaustive testing is impossible!

- Testing everything including all combinations of inputs priorities to focus testing efforts.

- That is we must prioritise our testing effort using a Risk based Approch.

3.Early Testing

- These activities should be focused on defined objectives outlined in the test strategy

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

4.Defect Clustering

- Defects are not evenly spread in a system
- They are ‘clustered’
- An important consideration in test prioritization!

- In other words, most defect found during testing are usually confined to a small number of modules.

5) Pesticide Peradox

- testing identifies bugs, and programmers respond to fix them
- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.

6) Testing is context Dependent

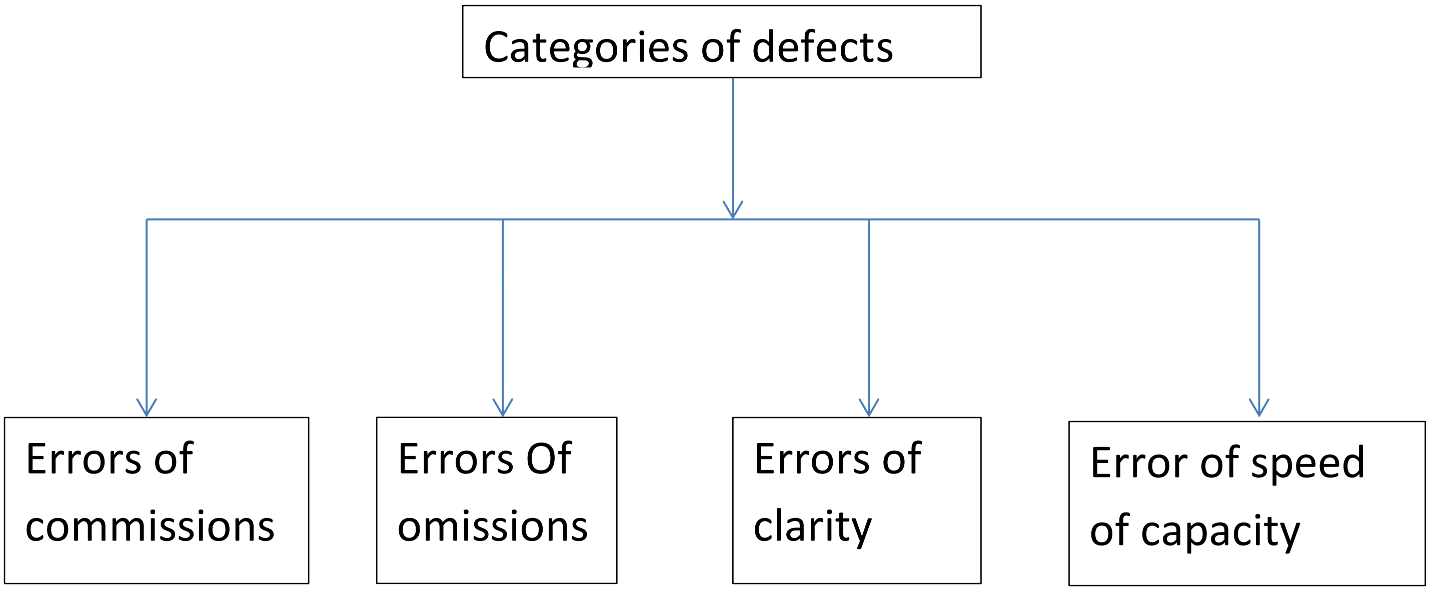
- Testing is basically context dependent
- Testing is done differently in different contexts
- Different kinds of sites are tested differently.

7) Absence of Errors Fallacy

- It doesn't make it a good system
- If we build a system and, in doing so, find and fix defect.
- Even after defects have been resolved it may still be unusable and/or does not fulfill the users' needs and expectations

19) Mention what are the categories of defects?

- Categories of defects are Errors of commissions, Errors Of omissions, Errors of clarity, and Error of speed and capacity.



- <u>Errors of commissions:-</u>

- Commission means instruction or some kind of command given.

- Now the error in commission means the error in made in command or instruction.

- For example, suppose I wrote a loop which I was trying to run 10 times but I command it to run more than 10 times by mistake this is the error of commission.

- <u>Errors Of Omissions:-</u>
- As name is already describing error of omission is something which happens accidentally.
- Omission word means something left out or executed.
- <u>Error of Clarity:-</u>
- The most common error in the natural languages.
- This error happens due to miss understanding between the developer and client.
- It travels most of the time from the requirements to the software.
- <u>Error of speed or capacity:-</u>
- The name of the error is 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.

20) What is the purpose of exit criteria?

Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

- ➢ Thoroughness measures, such as coverage of requirements or of code or risk coverage
- ➢ Estimates of defect density or reliability measures. (e.g. how many defects open by category)
- ➢ Residual Risks, such as defects not fixed or lack of test coverage in certain areas
- ➢ Schedules - such as those based on time to market

- Successful Testing of Integrated Application
- Executed Test Cases are documented
- All High prioritized bugs fixed and closed
- Technical documents to be submitted followed by release Notes

21) Mention what bigbang testing is?

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.

- The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.

- **Advantages:**

- Convenient for small systems.

- **Disadvantages:**

- Fault Localization is difficult.

- Given the sheer number of interfaces that need to be tested in this approach, some interfaces links to be tested could be missed easily.

- Since the integration testing can commence only after “all” the modules are designed, testing team will have less time for execution in the testing phase.

22) When should “regression testing” be performed?

When the system is stable and the system or the environment changes

- when testing bug-fix releases as part of the maintenance phase
- It should be applied at all Test Levels
- It should be considered complete when agreed completion criteria for regression testing have been met
- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation

23) Difference between QA/QC tester

Quality Assurance

- It is a subset of Software Test Life Cycle (STLC).
- Preventive activities.
- Process oriented activities.
- Focuses on processes and procedures rather than conducting actual testing on the system

Quality control

- It is a corrective process.
- It is a corrective process.

- Product oriented activities.

- Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.

Tester

- Testing is the subset of Quality Control.

- It is a preventive process.

- Product oriented activities.

- Focuses on actual testing.

24) Difference between smoke and sanity?

Smoke

- Smoke Testing is performed to ascertain program is working fine
- This testing is performed by the developers or testers
- Smoke testing is usually documented or scripted
- Smoke testing is like General Health Check Up

Sanity

- Sanity Testing is done to check the new functionality / bugs have been fixed
- Sanity testing is usually performed by testers
- Sanity testing is usually not documented and or scripted is unscripted
- Sanity testing is a subset of Acceptance testing

25) Difference between verification and Validation

Verification

- The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.

- To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.

- Are we building the product right?

Validation

- The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.

- The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.

- Are we building the right product?

26) Explain types of performance testing.

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

Load testing:-

- It’s 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.

- Some extremely popular sites have suffered serious downtimes when they get massive traffic volumes. E-commerce websites invest heavily in advertising campaigns, but not in Load Testing to ensure optimal system performance, when that marketing brings in traffic.

❖ Stress testing:-

- 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.

- Under Stress Testing, AUT is be stressed for a short period of time to know its withstanding capacity.

- Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.

❖ Endurance testing:-

- Endurance testing is non-functional type of software testing where A Software is tested with high load extended over a significant amount of time to evaluate the behavior of software application under sustained use.

- The main purpose of endurance testing is to ensure that the application is capable enough to handle extended Load without any deterioration of response time.

- This makes endurance testing differ from Load Testing, which usually ends in a couple of hours or so.

❖ Spike testing:-

- Spike testing is a type of software testing in which a software application is tested with extreme increments and decrements in traffic load.

- The main purpose of spike testing is to evaluate the behaviour of the software application under sudden increment or decrement in user load and determine recovery time after a spike of user load.

- Spike Testing is performed to estimate the weaknesses of software applications.

❖ Volume testing:-

- Volume testing is a type of Software Testing, where the software is subjected to a huge volume of data. It is also referred to as flood testing.

- Volume testing is done to analyze the system performance by increasing the volume of data in the database.

- With the help of Volume testing, the impact on response time and system behavior can be studied when exposed to a high volume of data.

❖**Scalability testing:-**

- Scalability Testing is a non-functional test methodology in which an application's performance is measured in terms of its ability to scale up or scale down the number of user requests or other such performance measure attributes.

- Scalability testing can be performed at a hardware, software, or database level.

- Parameters used for this testing differ from one application to the other, for a web page, it could be the number of users, CPU usage, and network usage, while for a web server it would be the number of requests processed.

27) What is Error, Defect, bug and failure?

"A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure"

Error:- A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

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

Bug:- 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.

28) Difference between priority and severity

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.

- 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.

29) What is bug life cycle?

"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'.

- As you can see from above diagram, a defect's state can be divided into Open or Closed.
- When a bug reaches one of the Closed or Terminal states, its lifecycle ends. Each state has one or more valid states to move to.

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

30) Explain the difference between functional testing and Non-functional testing

Functional testing:-

- Testing based on an analysis of the specification of the functionality of a component or system.

- 'Specification' – E.g. Requirements specification, Use Cases, Functional specification or maybe undocumented.

- Functional testing verifies that each functional of the software application operates in conformance with the requirement specification.

- This testing mainly involves black box testing and it is not concerned about the source code of the application.

Non-functional testing:-

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

- May be performed at all Test levels (not just Non Functional Systems Testing)

- Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling

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

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

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

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

❖STLC (Software Testing Life Cycle) :-

- ➢ In essence, STLC is related to testing. Meaning that it is a software testing process that entails several phases

- ➢ Since it is a part of SDLC and only involves testing, it is considered a child or successor.

- ➢ The Focus on test development and helps to make the testing process more sophisticated consistent and useful

- ➢ Phases of STLC are carried out after the phases of SDLC

- ➢ The QA team analyze requirement documents such as functional and non-functional requirements, and then prepare a system. Test plan as a part of the requirement analysis phases of the STLC.

❖ SDLC(Software development life cycle):-

- Software development and includes all phases of software development, including testing.

- As a whole, it covers the entire life cycle of the software and can be considered the predecessor.

- SDLC aims to manage the entire process of software development from start to finish and to deliver a quality product that meets customer needs.

- Phases of the SDLC are completed before those of the STLC.

- Business Analysts and Product Analysts collect requirements and prepare a Development Plan during the Requirements collection phase of the SDLC.

- Throughout the SDLC process, the intent is to overcome any hurdle on the way to successful software development

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

❖ Test Scenarios:-

- Is any functionality that can be tested.

- Is derived from test artifacts like business Requirement Specification (BRS) and Software Requirement Specification (SRS).

- Helps test the end-end functionality in an Agile way.

- Is more focused on what to test.

- Takes less time and fewer resources to create.

- Includes an end-to-end functionality to be tested.

- The main task is to check the full functionality of a software application.

- Allows quickly assessing the testing scope.

❖ Test case:-

- Is a set of actions executed to verify particular features or functionality

- Is mostly derived from test scenarios.

- Helps in exhaustive testing of an app.

- Is focused on what to test and how to test.

- Requires more resources and time.

- Includes test steps, data, expected result for testing.

- The main task is to verify compliance with the applicable standards, guidelines, and customer requirement.

- Allows detecting errors and defects.

❖ <u>Test script:-</u>

- Is a set of instruction to test an app automatically
- Is mostly derived from test cases.
- Helps to test specific things repeatedly.
- Is focused on the expected result.
- Requires less time for testing but more resources for scripts creating and updating.
- Includes different commands to develop a script.
- The main task is to verify that nothing is skipped, and the results are true as the desired testing plan.
- Allows carrying out an automatic execution of test cases.

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

- A test plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product.

- Test Plan helps us determine the effort needed to validate the quality of the application under test.

- 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.

- As per ISTQB definition: “Test Plan is a document describing the scope, approach, resources, and schedule of intended test activities.”

Covered Information:-

- **A first** and last name
- A home or other physical address, including street name and name of city or town
- **An email** address or other online contact information, such as an instant messaging user identifier or a screen name
- A telephone number
- **A Social** Security number
- A driver’s license or other state- issued identification number

34) What is priority?

- If something is a priority, it is the most important thing you have to do or deal with, or must be done or dealt with before everything else you have to do.

- Being a parent is her first priority.

- The government's priority is to build more power plants.

- Getting your priorities in order is a good way to not waste energy on meaningless pursuits.

35) What is severity?

- 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.

- 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.

- The severity is high but priority is low.

36) Bug categories are…

- Thare are some category
- Security
- Database
- Functionality (Critical/General), UI

37) Advantage of Bugzila.

- Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.

- Bugzilla is a defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other test case management tools like Quality Center, Testlink etc.

- This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the datamanagement chain.

- Key features of Bugzilla includes
- Advanced search capabilities
- E-mail Notifications
- Modify/file Bugs by e-mail
- Time tracking
- Strong security
- Customization
- Localization

38) Difference between priority and severity

Priority testing:-

- Priority is a term that defines how fast we need to fix a defect.
- Priority is basically a parameter that decides the order in which we should fix the defects.
- Priority relates to the scheduling of defects to resolve them in software.
- Priority relates to the scheduling of defects to resolve them in software.
- The value of priority is subjective.
- The value of Priority changes from time to time.

Severity testing:-

- Severity is a term that denotes how severely a defect can affect the functionality of the software.

- Severity is basically a parameter that denotes the total impact of a given defect on any software.

- Severity relates to the standards of quality.

- The value of severity is objective.

- The value of Severity changes continually from time to time.

39) What are the different Methodologies in Agile Development Model?

1.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 specific characteristics like simplicity, sustained productivity and strength for blending several

Underlying approaches adapted by other agile methods, Scrum has obtained popularity over the years.

2.Kanban:

- Kanban is an eminently visual workflow management approach, 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.

- Similar to scrum, kanban is the process developed for supporting collaborative teamwork more effectively.

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

Authentication	**Authorization**
Authentication verifies who the user is.	Authorization determines what resources a user can access.
Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.	Authorization works through settings that are implemented and maintained by the organization.
Authentication is the first step of a good identity and access management process.	Authorization always takes place after authentication.

Authentication is visible to and partially changeable by the user.	Authorization isn’t visible to or changeable by the user.
Example: By verifying their identity, employees can gain access to an HR application that includes their personal pay information, vacation time, and 401K data.	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.

41) When to use Usability Testing?

- Usability testing is a method of testing the functionality of a website, app or other digital product by observing real users as they attempt to complete tasks on it. The users are usually observed by researchers working for a business.

- Usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you have begun the strategy work around a brand new site or app.

42) What is the procedure for GUI Testing?

- Check all the GUI elements for size, position, width, length and acceptance of Characters or numbers. For instance, you must be able to provide inputs to the input fields.

- Check you can execute the intended functionality of the application using the GUI.

- Check Error Messages are displayed correctly
- Check for Clear demarcation of different sections on screen
- Check Font used in application is readable
- Check the alignment of the text is proper
- Check the Color of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Check the positioning of GUI elements for different screen resolution

43) Write agile manifesto principles?

- **Customer satisfaction through early and continuous software delivery** –Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.

- **Accommodate changing requirements throughout the development process** – The ability to avoid delays when a requirement or feature request changes.

- **Frequent delivery of working software** – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.

- **Collaboration between the business stakeholders and developers throughout the project** – Better decisions are made when the business and technical team are aligned.

- **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.

- **Enable face-to-face interactions** – Communication is more successful when development teams are co-located.

- **Working software is the primary measure of progress -**Delivering functional software to the customer is the ultimate factor that measures progress.

- **Agile processes to support a consistent development pace** – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.

- **Attention to technical detail and design enhances agility** – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.

- **Simplicity** – Develop just enough to get the job done for right now.

- **Self-organizing teams encourage great architectures, requirements, and designs** – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.

- **Regular reflections on how to become more effective** – Self Improvement, process improvement, advancing skills, and techniques help team members work more efficiently.