Software Testing Assignment –

Module_2 Manual Testing

Prepared By: Saurabh Jadav

Course: Software Testing

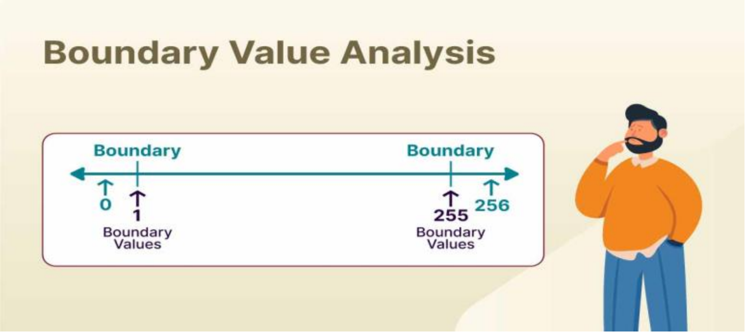
1.What is Exploratory Testing?

- Exploratory testing is a new way of thinking. You are exploring the app. This is not structured way of testing.
- Exploratory testing is an unscripted, hands-on approach to testing software, where a tester:
  - Learns about the software's features and functionality.
  - Designs and executes tests simultaneously.
  - Explores the software to identify defects and issues.
  - Uses their skills, experience, and creativity to test the software.

2.What is traceability matrix?

- Traceability matrix is a document that demonstrates the relationship between requirements and other artifacts. It's used to prove that requirements have been fulfilled. And it typically documents requirements, tests, test results, and issues.
- A **software process** should help you keeping the virtual table up-to-date.
- To find out the cause of defect.
- To ensure the complete coverage of testing.

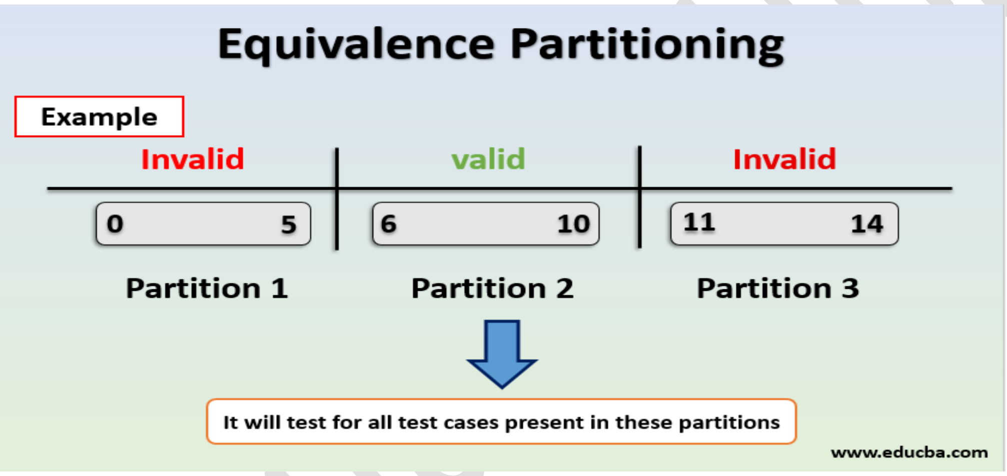
3.What is Boundary value testing?



- Boundary-value analysis is a software testing technique in which tests are designed to include representatives of boundary values in a range. This is the part of Black Box Testing Methods.
- In simple terms boundary value Analysis is like testing the edge cases of our software where most of the time it will get broke.

4.What is Equivalence partitioning testing?

- This is the method of black box testing that is use to drive the long-range inputs into equivalence partition and pick one representative from the partition for checking the hole range.
- Ex. -Input field for:
  - Partition 1: 0-5
  - Partition 2: 6-10
  - Partition 3: 11-14

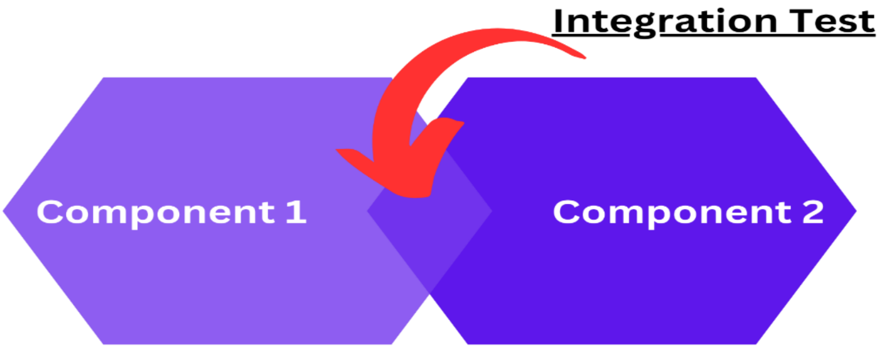

Equivalence Partitioning

It will test for all test cases present in these partitions

www.educba.com

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 is the process of testing the interface between two software units or modules.

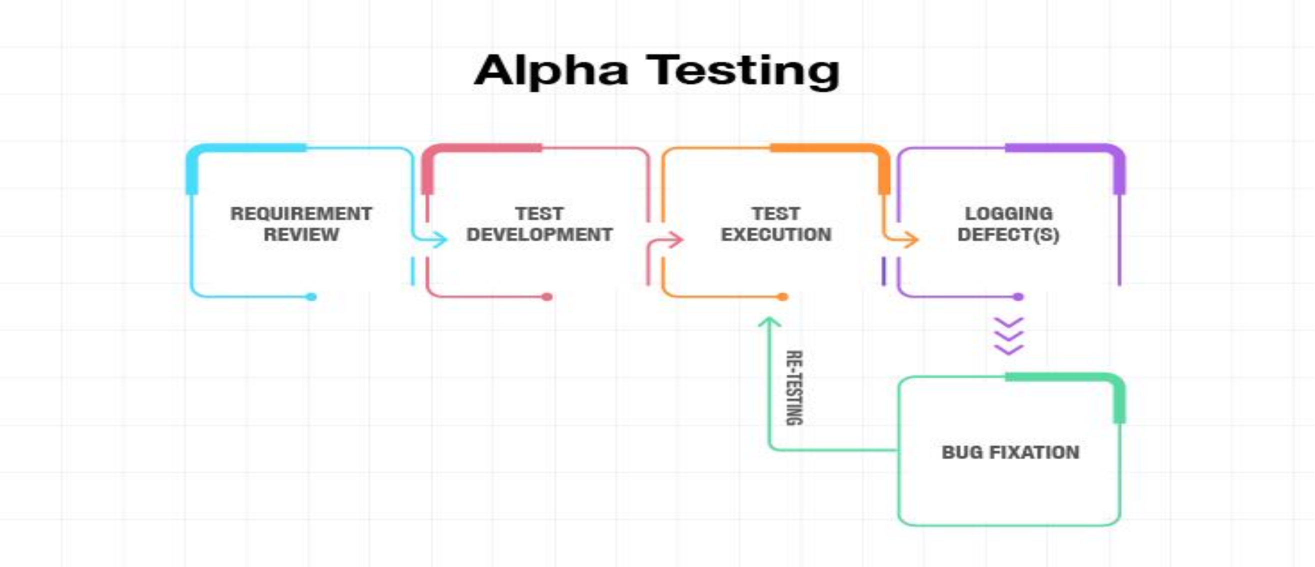


6.What determines the level of risk?

- In the real software testing scenario, the risk level is determined by two dimensions: probability and impact.
- **Probability:** It measures the likelihood of an event occurring, typically expressed as a percentage or qualitative scale.
- **Impact:** Risk, by default, brings a negative impact to any project.

7.What is Alpha testing?

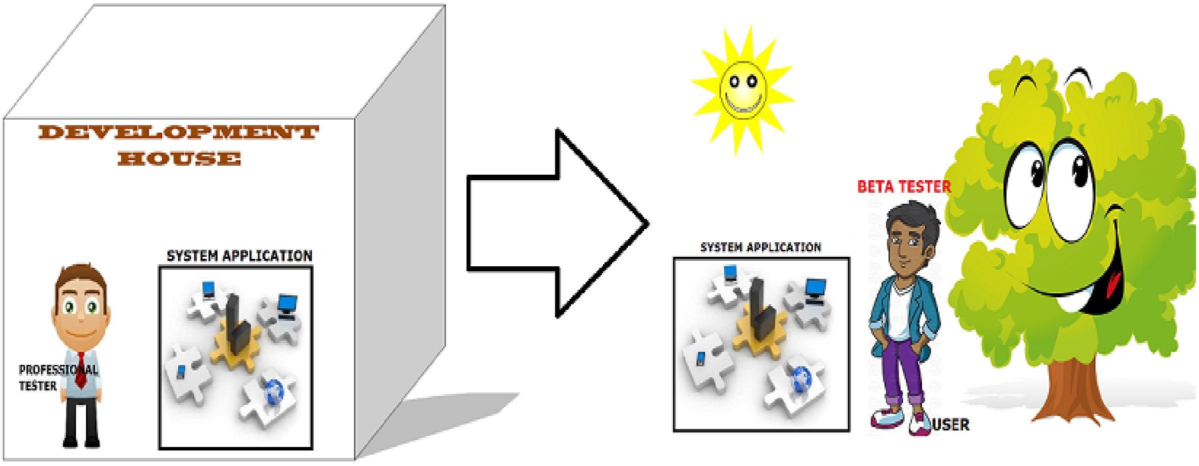
- Alpha testing is a critical phase of software testing conducted by the development or QA team before beta testing. The purpose of alpha testing is to identify and resolve critical bugs and issues in the software before it is released to the public.



- It is always performed by the developers at the s/w development site.
- Sometimes it is also performed by Independent Testing Team.
- Alpha Testing is not open to the market and public.
- It is always performed in Virtual Environment, Within the organization.
- It is the form of Acceptance Testing.
- It comes under the category of both White and Black Box Testing.

8.What is beta testing?

- Beta testing is the last phase of the testing process before a product is released. The purpose of this phase is to evaluate the level of customer satisfaction with the product.



BETA TESTING OF THE PRODUCT
IN REAL WORLD ENVIRONMENT

- 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.

9.What is component testing?

- Component testing, also known as unit testing or module testing, is a level of software testing that focuses on verifying the individual components or units of a system.

10.What is functional system testing?

- Functional system testing meets the functional requirements and works as expected. It involves testing the system's functionality, user interactions, and business processes from start to finish.
- To test the attributes of the system directly affected to the functionalities.

11.What is Non-Functional Testing?

- Non-functional testing is a type of software testing that evaluates the software's characteristics, properties, and behaviors that are not related to its functional requirements.
- To test the attributes of the system indirectly affected to the functionalities.

12.What is GUI Testing?

- GUI testing is a software testing technique that checks the Graphical User Interface of the software application to ensure the application's functionality and features meet the business requirements. Some of the common things that are checked in GUI testing include like Size, position and width of the images, Error messages, Sections on screen, Font and color of the text, Alignment of the text and images etc.
- 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 and properly aligned.
- Check the positioning of GUI elements for different screen resolution.

13.What is Ad-hoc testing?

- Ad-hoc testing is informal testing technique type. Main aim of this testing is finding the defect with random checking and break the system.
- It does not follow any test design techniques to create test cases.
- This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Main aim of this testing is to find defects by random checking.
- Adhoc testing can be achieved with the testing technique called Error Guessing.
- Types of Adhoc Testing:

  1) Buddy Testing (one developer, One tester)

  2) Pair Testing (Two persons pared as testers)

  3) Monkey Testing (randomly jump to the app. area to break)

14.What is load testing?

- This a type of performance testing. that determines how well a system, software product or application performs under real-life load conditions.
- Stability + response time + applying load (app will withstand with designed no. of users).

15.What is stress Testing?

- Stress testing is a type of software testing that helps to ensure the stability and reliability of a system. Stress testing is performed until the system fails or crashes, which helps to identify the weak points and vulnerabilities of the system.
- Stress Testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing.

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

- White box testing is approach in which testing is based on an analysis of the internal structure. White box testing analyzes the internal code, internal design and working functionality. White box testing is also known as transparent testing an open box testing.
- There are three methods of white box testing:
  1. **Statement coverage:** Statement coverage also known as line coverage and segment coverage. This method covers only true condition
  2. **Decision/Branch coverage:** This method covers the both conditions true and false.
  3. **Condition coverage:** Test all conditions (true/false) in each decision point.

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

- Black box testing, to check the application without having any knowledge of the internal part / source code. In black-box testing the tester is concentrating on what the software does, not how it does it.
- There are four methods of black box testing.

1. **Equivalence partitioning (EP):** It is a method to drive the long-range inputs into equivalence partition and pick one representative from the partition for checking the hole range.
2. **Boundary value analysis:** To check the boundary values with valid range or invalid range.
3. **Decision tables:** Boolean expressions: true or false

   Decision table shows the relationship between inputs and possible outputs are mapped together.

4. **State transition/transaction testing:** To test the system by all given transaction stored in finite state machine. Any system where you get different outputs for the same input.

18.Mention what are the categories of defects?

- Defects can be categorized into different types basing on the core issues they address.
- Some defects address security or database issues while others may refer to functionality or UI issues.
- Here some categories of defects listed below.
- **Functionality Defects:** Defects directly related to functionalities. Not working features properly.
  e.g Calculator has no '=' button for the calculation.

- **Performance Defects:** Software doesn't meet the expected performance requirements.
  e.g Website's loading time to open.

- **User Interface Defects:** Difficult to operate for the users. Not user friendly.
  e.g Login page has no cancel button, Alignment problem.

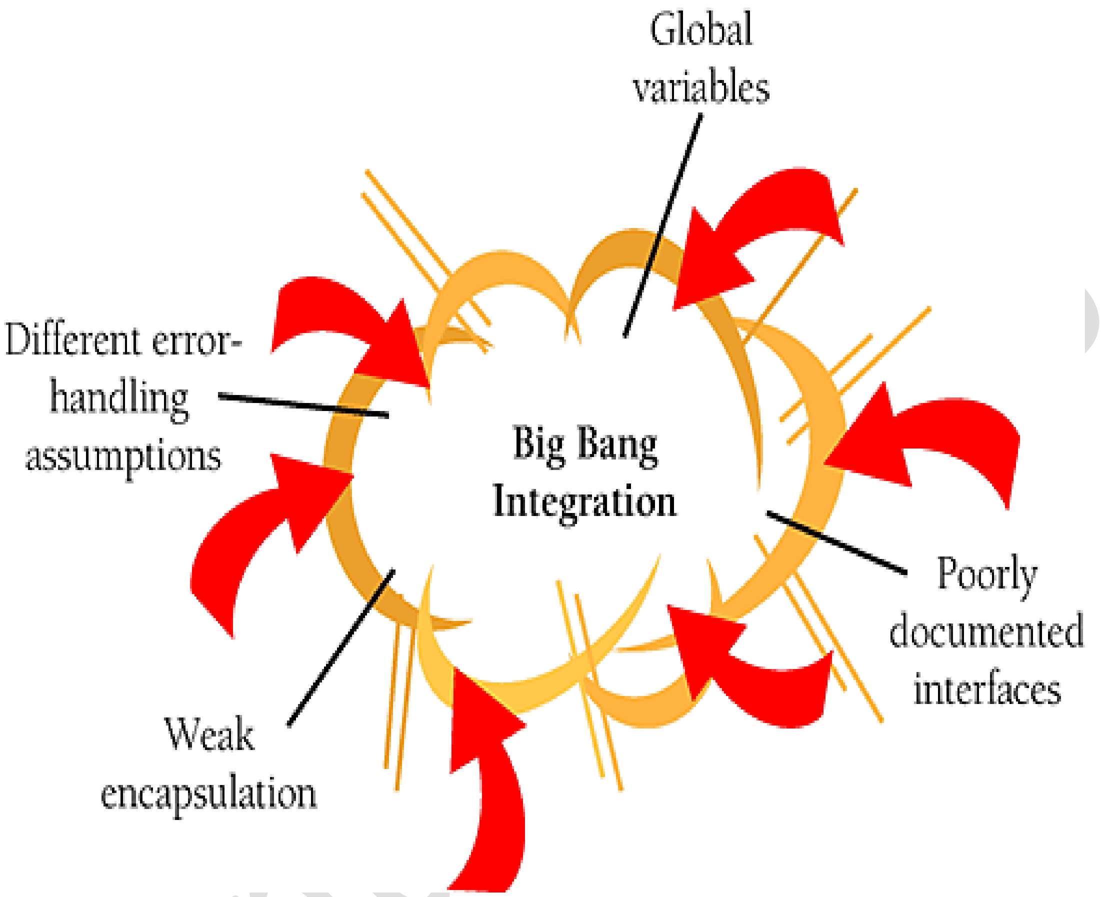
- **Compatibility Defects:** Software does not work correctly on different hardware and software configuration.
  e.g Application not running on Android or Windows platform.
  Application interface shows differently in different browsers.

- **Security Defects:** Software doesn't protect the user's data from malicious attack.
  e.g Password entered in visible form.
  Authentication: Accepting an invalid username/password
  Authorization: Accessibility to pages though permission not given

- **Documentation Defects:** Document is incorrect or inaccurate to use the features of the app.
  e.g TC had a wrong entry.
- **Database Defects:** When someone enter the data it enter wrong way into the Database.

19.Mention what big-bang testing is?

- That is a part of Integration Testing. Big Bang testing is a testing approach where all individual components or modules of a software application are tested together in a single.


Global variables
Different error-handling assumptions
Big Bang Integration
Poorly documented interfaces
Weak encapsulation

- Big Bang testing has the advantage that everything is finished before integration testing starts.
- In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
- It works as a simulation of the complete software. All the modules are combined together and tested simultaneously.
- The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.

20.What is the purpose of exit criteria?

- The main purpose of exit criteria is all condition must be full fill before leaving the project. Ensure that the project has met its objectives and requirements.
- 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.When should "Regression Testing" be performed?

- Regression testing should be performed in the following scenarios:
  - When new functionality is added to the system.
  - When some defect is identified in the software and the code is debugged to fix it.
  - When the code is modified to optimize its working.
  - Whenever the production code is modified.
  - When a website's new feature is added.
  - When a bug is fixed by developers.
  - When there is an update in the database from one software to another.
  - Defect Fixing.
  - Performance issue fix.
  - When testing bug-fix releases as part of the maintenance phase
  - When the system is stable and the system or the environment Changes.

22.What is 7 key principles? Explain in detail.

- The 7 Key Principles of Software Testing are fundamental guidelines that ensure effective testing and quality assurance. These principles are applied in the software testing industry. Here's a detailed explanation of each principle.

1. **Testing Shows the Presence of Defects:**

- Testing can show that defects are present, but cannot prove that there are no defects.
- We can just reduce the probabilities of the defect.
- E.g. 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 there is no defect.

2. **Exhaustive Testing is Impossible:**

- Testing everything including all combinations of inputs and preconditions is not possible.
- Exhaustive testing is complex because of requiring too much time & resources as well as it can be very expensive.
- E.g. If a company has 10000 employees and you want to test each emp_no. It is not possible to test every emp_no because it is very time consuming.

3. **Early Testing:**

- The testing process as early as possible in the Software Development Life Cycle (SDLC) to identify and address defects sooner.
- Early testing saves time and money.
- E.g. If bug found early then it less cost effective, if it found later on like production or maintenance phase, it should be more expensive.

4. **Defects Clustering:**

- A majority of defects are often found in a small portion of the software.
- Defects are need to be clustered because one small bug can spread in the system.
- E.g. When I fix a defect, I change some code. Whenever I create or code, there’s a probability that I will introduce a new defect. When I fix these new defects, I change more code, which creates a higher probability of more new defects.

5. **The Pesticide paradox:**

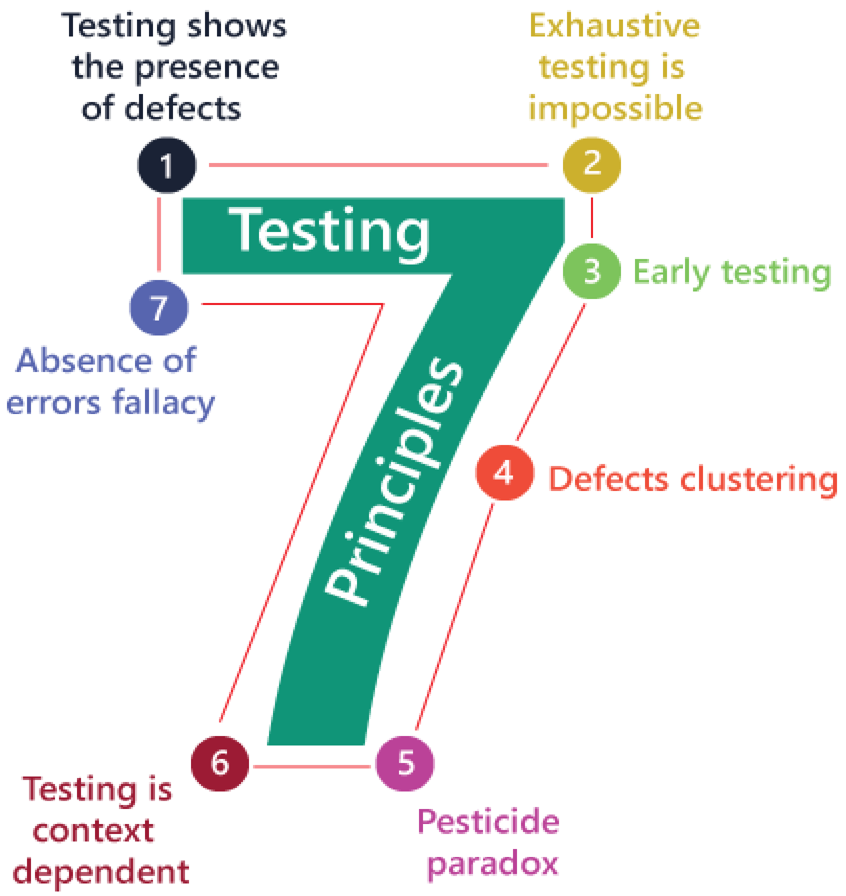
- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
- Rethink and write more and better test cases, especially when it is a crucial part of the application.
- E.g. Consider a resource scheduling software that schedules resources for tasks based on their work timings, time zones, and holidays. Testers wrote ten test cases related to scheduling, and after four rounds of testing, all the test cases passed. Does this mean the module is defect-free? Probably not, since it took four cycles to clear ten bugs.

6. **Testing is Context-Dependent**:

- Testing is basically context dependent.
- Testing is done differently in different contexts.
- Different kinds of sites are tested differently
- E.g. · Safety – critical software is tested differently from an e-commerce site.
- Game application is tested differently from Flight reservation application.

7. **Absence of errors is a fallacy**:

- Even if no defects are found, it doesn't mean the software is error-free. Defect free software is not as important as meets user requirement important.
- If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.
- E.g. Linux always had very few bugs, while Microsoft Windows was notorious for its bugs. However, most people used Microsoft Windows as their operating system because they found it easier to use and solved their problems better.


Testing shows the presence of defects
1
Exhaustive testing is impossible
2
Testing
3
Early testing
7
Absence of errors fallacy
Principles
4
Defects clustering
6
5
Testing is context dependent
Pesticide paradox

23.Difference between QA v/s QC v/s Tester

Quality Assurance	**Quality Control**	**Tester**
Focused on preventing defect and ensuring quality throughout the software development process.	Focuses on detecting and correcting defects in the software product.	Focuses on actual testing.
Quality Assurance (QA) is a process-oriented approach to ensuring that a product or service meets a set of quality standards	Quality Control (QC) is a product-oriented approach that focuses on identifying and fixing defects in the final product.	It includes activities that ensure the identification of bugs/error/defects in a software.
Involves testing, inspection, and validation to ensure the software meets the required standards.	Involves planning, designing, and implementing processes and procedures to ensure quality.	Product-oriented activities.
QA is process oriented	QC is product oriented.	Product-oriented activities.
QA is a managerial tool.	QC is a corrective tool.	It is corrective tool.
Preventive process	Corrective process	Preventive process
Subset of STLC (Software Testing Life Cycle)	Subset of QA	Subset of QC

24.Difference between Smoke and Sanity?

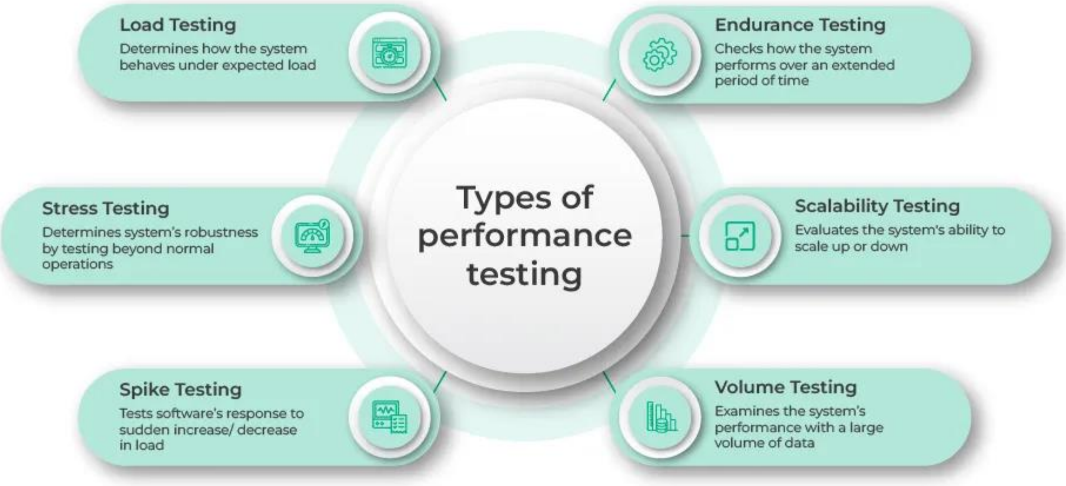
Smoke testing	Sanity testing
To check the stability of the system.	To check the rationality of the system.
Smoke testing is scripted, documented, well-planned according to STLC	Sanity testing is unscripted, not documented, not well-planned.
This is the subset of acceptance testing	This is the subset of the regression testing.
That is performed by the developer & tester.	That is performed by tester only.
Ex. -To check the login functionality works, and the application doesn't crash or produce errors.	Ex. -To check the login functionality to work as expected with a focus on the logical flow and expected behavior.

25.Difference between verification and Validation

Verification	Validation
Verification does not involve code execution	Validation involves code execution.
The verifying process includes checking documents, design, code, and program	Validation is a dynamic mechanism of testing and validating the actual product
It finds bugs early in the development cycle	It can find bug that the verification process can’t find the bug
It comes before validation	It comes after verification

26.Explain types of Performance testing.

- Performance testing is a testing measure that evaluates the speed, responsiveness and stability of a software program under a workload.
- The goal of performance testing is to identify and nullify the performance bottlenecks in software applications, helping to ensure software quality.
- Performance testing aims to examine the following factors of software performance: speed, robustness, reliability, application size, scalability, stability, and response times.


Load Testing
Determines how the system behaves under expected load
Endurance Testing
Checks how the system performs over an extended period of time
Stress Testing
Determines system's robustness by testing beyond normal operations
Types of performance testing
Scalability Testing
Evaluates the system's ability to scale up or down
Spike Testing
Tests software's response to sudden increase/ decrease in load
Volume Testing
Examines the system's performance with a large volume of data

- **Load:** 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.
  - E.g. For a newly developed application with an anticipated load of around 1000 concurrent users. We will create a load test script and configure it with 1000 virtual users and run it for say 1-hour duration. After the load test completion, we can analyze the test result to determine how the application will behave at the expected peak load.

- **Stress:** 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**.**

  "To determine or validate an application's behavior when it is pushed beyond normal or peak load conditions."

  - Stress Testing is done to make sure that the system would not crash under crunch situations.
  - E.g. For an application with an anticipated load of 1000 users we will run the test with 1200 users and check if the application is robust enough to not crash.

- **Endurance:** 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.

  - E.g. For an application like Income tax filing, the application is used continuously for a very long duration by different users. In this type of application, memory management is very critical. For an application like these, we can run the test for 24 hours to 2 days duration and monitor the memory utilization during the whole test execution.

- **Scalability:** Scalability testing determines the application performance by gradually increasing the user load and monitoring the system behavior. The increasing number of users on a stable application helps analyze its effectiveness.

  - E.g. The application should sustain a throughput of 500 requests per second with an average response time under 3 seconds.

- **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.
    - E.g. For an e-commerce application running an advertisement campaign, the number of users can increase suddenly in a very short duration.
- **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.
    - Volume testing is also known as flood testing, where you populate a database with a huge number of data and then observe and track the system's behavior to analyze application performance.
    - E.g. For a newly developed e-commerce application, we can perform volume testing by inserting millions of rows in the database and then carry out the performance test execution.

27.What is Error, Defect, Bug and failure?

Error: A mistake is code/incorrect value is called Error.

Defect: When the tester finds the error, it is called a defect.

Bug: When defect is accepted by developer or developer team, defect is called a bug.

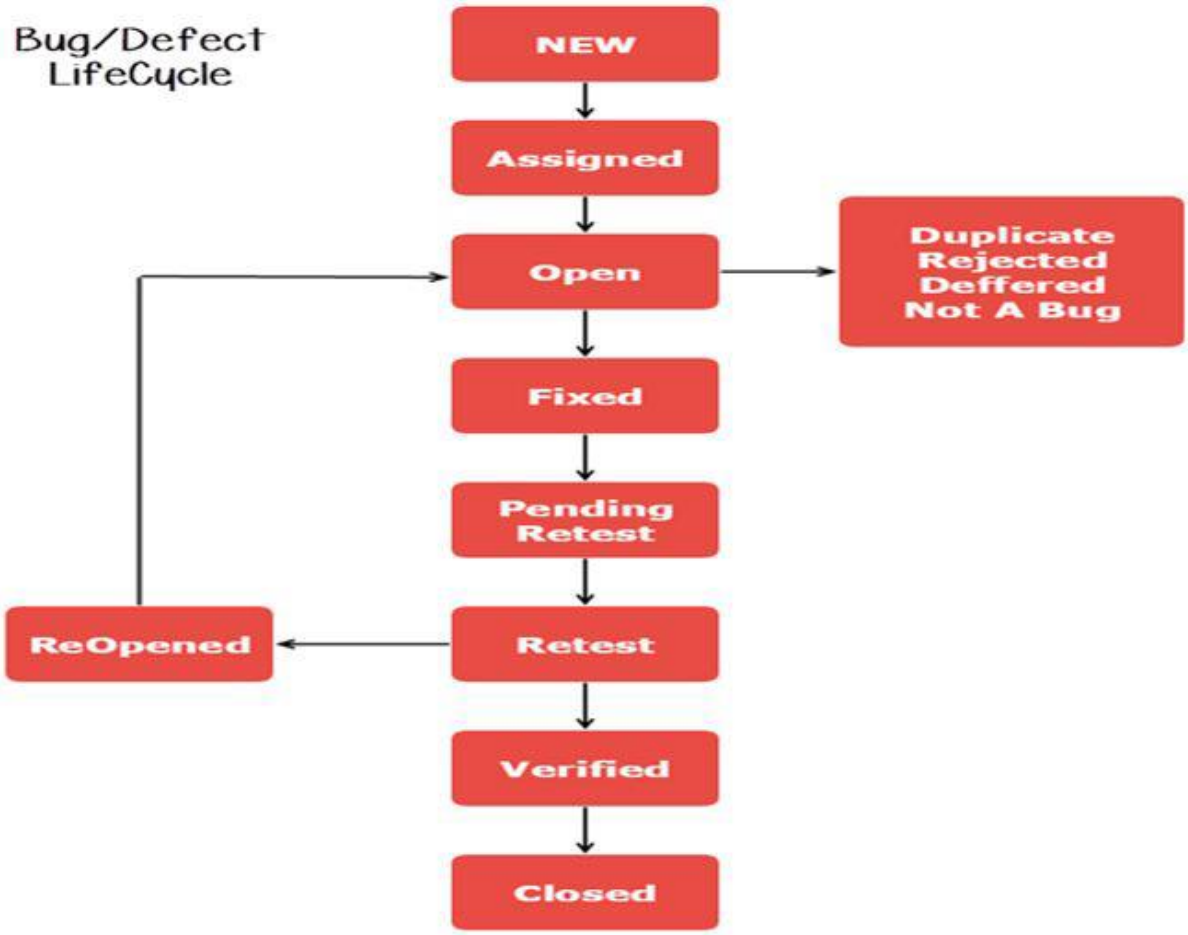
Failure: Software Build failure refers to a situation where a software system or application does not perform as expected, does not meet user requirement.

28.Difference between Priority and Severity?

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.

29.What is Bug Life Cycle?

- The duration or timespan between the first time defect found and the time when defect closed, rejected, differed or postponed is called defect/bug life cycle.



- **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
- **Open**: The developer starts analysing and works on the defect fix
- **Fixed**: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
- **Pending retest**: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing

remains pending from the testers end, the status assigned is "pending retest."

- **Retest**: TE retests that code to check, a bug is fixed or not.
- **Verified**: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."
- **Reopen**: If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.
- **Closed**: If the bug is no longer exists then tester assigns the status "Closed."
- **Duplicate :** Defect repeating twice.
- **Rejected:** DEV feels defect is not genuine.
- **Differed:** Present bug not having prime priority. Defect can be fixed in next release.
- **Not a bug:** Defect does not affect the functionality of an app.

30.Explain the difference between Functional testing and Non-Functional testing?

Functional testing	**Non-Functional testing**
Ensure the software performs its intended functions correctly.	Evaluate the software's performance, security, usability, and other non-functional aspects.
Easy to do manual & automation with Functional Testing.	Tough to do manual testing with Non-functional testing.
First to execute Functional.	Non-functional should be executed after Functional testing.
Functional testing describes what the product does.	Nonfunctional testing describes how good the product works.

31.To create HLR & Test-Case of (Instagram, Facebook) only first page.

HLR & Test-Case of Facebook →	Click here
HLR & Test-Case of Instagram →	Click here

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

- The STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle) are both crucial processes in software development, but they serve different purposes and focus on different aspects of the software development and testing process. Here are the key differences between STLC and SDLC:

SDLC	STLC
Focused on software development	Focused on software testing
Helps to develop good quality software.	Helps to make software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Coders create a well-organized development plan.	QA team defines the test plan.
Developers create the actual software.	Tester designs test cases, set up the environment & work out the RTM.

- The SDLC provides a linear process to accurately build the correct system, whereas the STLC allow you to robustly test what has been built.

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

Test Scenario	Test Case	Test Script

Is any functionality that can be tested.	Is a set of action executed to verify particular feature or functionality.	Is a set of instructions to test an app automatically.
Helps to test end to end functionality in an Agile Way.	Helps in executing testing an app.	Helps to test specific things repeatedly.
Includes an end-to-end functionality to be tested.	Includes test step, data, expected result for testing.	Includes different commands to develop a script.
Allow quickly assessing the testing scope.	Allow detecting error and defects.	Allowing carrying out an automatic execution of test case.

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

- A Test Plan is a detailed document outlining the approach, scope, and timeline for testing a software application or system. It serves as a guide for the testing process, ensuring that the testing is thorough, efficient, and effective.
- A comprehensive Test Plan should cover the following information:
- **Test Objectives:** Clearly state the goals and objectives of testing.
- **Scope:** Define what is included and excluded from testing.
- **Test Environment:** Describe the hardware, software, and network settings for testing.
- **Test Approach:** Outline the testing methodology, techniques, and tools.
- **Test Cases:** List the specific test scenarios and expected results.
- **Test Data:** Identify the data required for testing and how it will be obtained.
- **Test Schedule:** Provide a timeline for testing, including milestones and deadlines.
- **Resources:** Identify the personnel, equipment, and budget required for testing.

- **Risks and Assumptions:** Document potential risks and assumptions made during testing.
- **Test Deliverables:** Define the outputs and artifacts expected from testing, such as test reports and defect logs.
- **Test Criteria:** Establish the criteria for determining when testing is complete and successful.

35. What is Priority?

- Priority refers to the urgency with which a defect needs to be fixed. It measures how quickly the issue should be resolved based on the business impact or the project timeline. The defect priority of a defect can also be classified into different levels such as high, medium, or low.

36. What is Severity?

- Severity refers to the impact of a defect on the functionality or the end-user experience. It measures how severe the issue is and how critical it is to fix it. The severity of a defect can be classified into levels such as critical, high, medium, low or cosmetics.

37. Bug categories are….?

- Defects can be categorized into different types basing on the core issues they address.
- Some defects address security or database issues while others may refer to functionality or UI issues.
- Here some categories of defects listed below.
- **Functionality Defects:** Defects directly related to functionalities. Not working features properly.
  e.g Calculator has no '=' button for the calculation.

- **Performance Defects:** Software doesn't meet the expected performance requirements.
  e.g Website's loading time to open.

- **User Interface Defects:** Difficult to operate for the users. Not user friendly.

e.g Login page has no cancel button, Alignment problem.

- **Compatibility Defects:** Software does not work correctly on different hardware and software configuration.
  e.g Application not running on Android or Windows platform.
  Application interface shows differently in different browsers.

- **Security Defects:** Software doesn't protect the user's data from malicious attack.
  e.g Password entered in visible form.
  Authentication: Accepting an invalid username/password
  Authorization: Accessibility to pages though permission not given

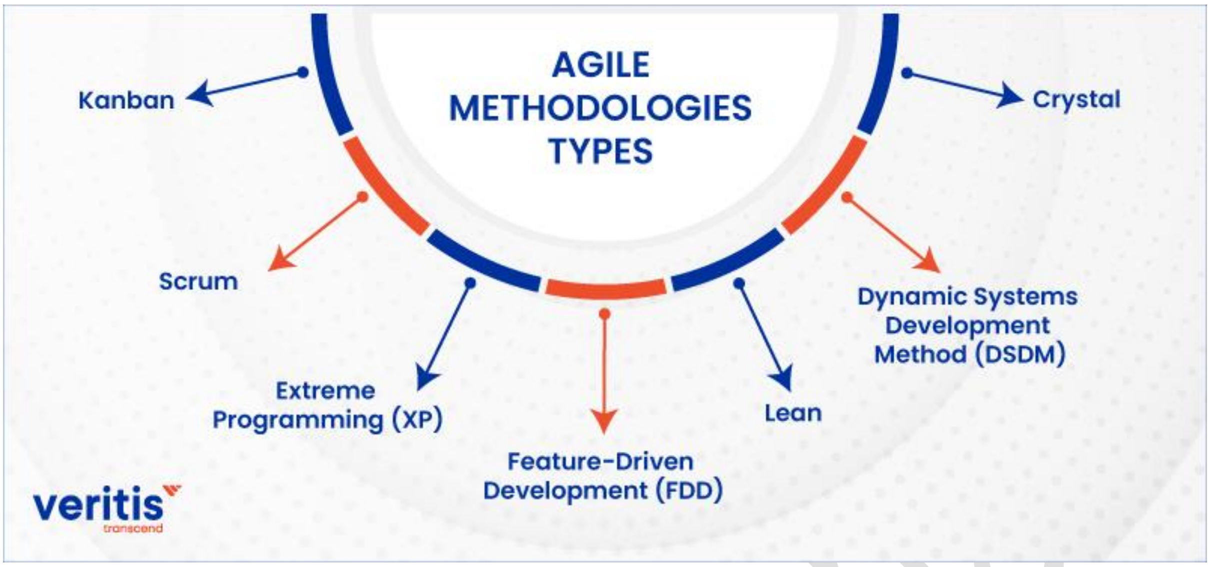
- **Documentation Defects:** Document is incorrect or inaccurate to use the features of the app.
  e.g TC had a wrong entry.
- **Database Defects:** When someone enter the data it enter wrong way into the Database.

38. Advantages of Bugzilla?

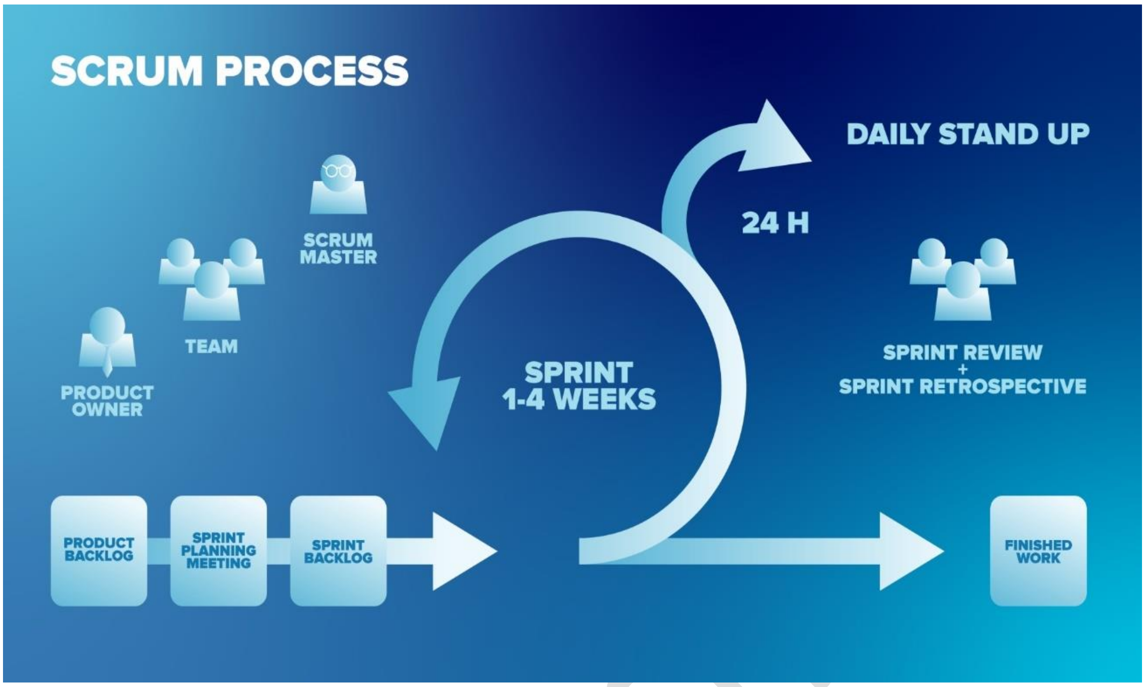
- Bugzilla is an open source bug-tracking system.
- **Deadlines:** To fix the bugs, deadlines can be established.
- **Types:** It reports in a variety of formats and types.
- **Request System:** You can use the 'request system' provided by Bugzilla to ask other users to evaluate codes, provide information and other things.
- **Flexible:** Bugzilla is quite flexible, so you can modify it to fit your unique process and requirements.
- **Bug tracking tool:** Bugzilla is extremely good at monitoring and handling bugs and issues.

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

- Agile is a philosophy, set of values and principles to make a decision for developing a software.

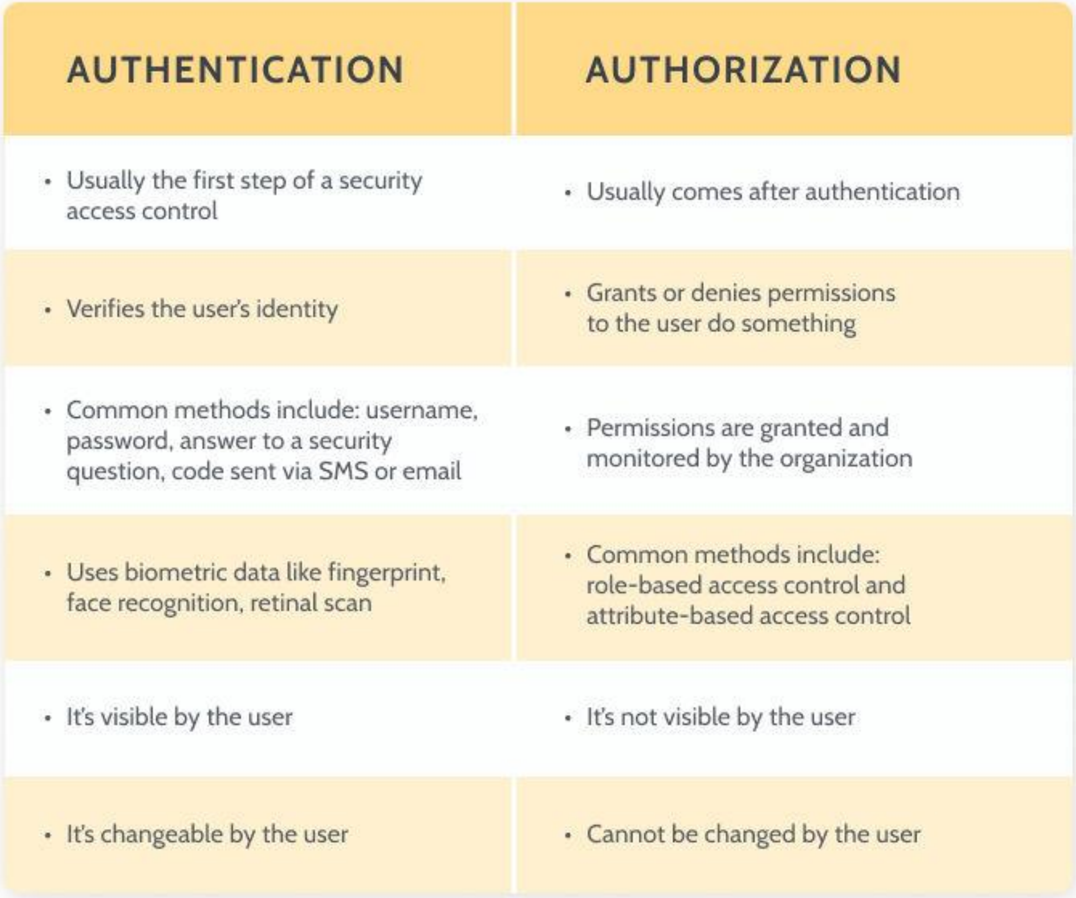

AGILE METHODOLOGIES TYPES
Kanban
Crystal
Scrum
Dynamic Systems Development Method (DSDM)
Extreme Programming (XP)
Lean
Feature-Driven Development (FDD)
veritis transcend

- Here, we will discuss widely used 2 methodologies of agile.
- **Scrum:-**
  - Scrum is a framework through which we build software product by following Agile principles.
  - SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team-based development environment.
  - Scrum includes group of people called a scrum team. Normally contains 5 to 9 members.
  - Scrum team can involve the people like product owner, scrum master, DEV team, QA team etc.
  - There are three roles in it, and their responsibilities are:
  - **Scrum Master:** The scrum can set up the master team, arrange the meeting and remove obstacles for the process
  - **Product owner:** The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.
  - **Scrum Team:** The team manages its work and organizes the work to complete the sprint or cycle.



- **Kanban:**
- Kanban is a very popular framework for development in the agile software development methodology.
- It provides a transparent way of visualizing the tasks and work capacity of a team.
- It mainly uses physical and digital boards to allow the team members to visualize the current state of the project they are working on.
- Kanban originated in Toyota in the 1940s.
- Kanban’s meaning in Japanese is “billboards.”
- The Kanban board has columns and story cards.
- The columns are nothing, but workflow states and cards are nothing but a demonstration of the actual task a team member is performing.
- Kanban should be used when you want to visualize your work, and you want to see the progress of your tasks visually.

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



AUTHENTICATION	AUTHORIZATION
• Usually the first step of a security access control	• Usually comes after authentication
• Verifies the user's identity	• Grants or denies permissions to the user do something
• Common methods include: username, password, answer to a security question, code sent via SMS or email	• Permissions are granted and monitored by the organization
• Uses biometric data like fingerprint, face recognition, retinal scan	• Common methods include: role-based access control and attribute-based access control
• It's visible by the user	• It's not visible by the user
• It's changeable by the user	• Cannot be changed by the user

- ➢ Web testing have common problem like:
- ➢ **Testing web applications can present a number of challenges, including:**
- ➢ **Performance:** Web applications may crash or slow down when there is a sudden increase in internet traffic. Performance testing ensures that the application's speed is not affected by increased activity.
- ➢ **Security:** Security testing verifies that an application's design or configuration is correct and that it is not vulnerable to security threats.
- ➢ **Cross-browser compatibility:** Different browsers and versions can cause inconsistencies in how web content is rendered or functionality issues. Cross-browser compatibility testing ensures a consistent experience across different browsers and device viewports.

- **Usability:** Developers may face issues with scalability and interactivity when testing usability. To test usability, developers should use a representative group of users to test the application across different browsers and hardware.
- **UI testing:** Testing all the features and components of a web application one by one can be challenging, especially if the application has complex features or a complicated infrastructure.
- **Regression testing:** Creating and maintaining regression test cases can be time-consuming and resource intensive. The software may need to be tested after every change.
- **System integration:** Inconsistent environment, infrastructure, interaction model, performance, and reliability issues can be challenging.

41. When to used Usability Testing?

- Usability testing is a technique that can be used throughout the product development cycle to help designers understand their users' needs and behaviors. It can be conducted at many different stages, including:
- **Usability testing ideas:** Once you've got an idea, conduct usability testing before putting any design resources to work. Identify specific areas where testing and validation can enhance your concept. After you get the results from your initial test, share them with your team. Then, continue testing users as you build a prototype.
- **Usability testing prototypes:** Once your concept is fleshed out, you can build a prototype. Prototype-based usability testing lets you test different designs against each other. Invite people who match your ideal users to try out the prototype. You'll see firsthand whether users are getting it. This will get your prototype in the best shape before development.

- **Usability testing before launch:** Running usability tests is a quick and easy way to tease out any unknown bugs or inconveniences users experience with your website or app. To assess, run remote unmoderated tests to see if contributors struggle with specific elements or flows. If so, redesign the user interface of the website or app based on the feedback and then evaluate again with a small set of users.
- **Usability testing after launch:** Your work isn't done despite testing throughout the design and development cycle. You'll want to continue usability testing after a product or feature goes live to optimize consistently. This is how organizations create unique experiences that stand the test of time as user behavior evolves.

42. What is the procedure for GUI Testing?

- 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.
- Font size, style, & readable font, image resolution & clarity, size,
  Error message window, color combination.
- Approach of GUI Testing

  MANUAL BASED TESTING
  RECORD AND REPLAY
  MODEL BASED TESTING
- The procedure for GUI testing involves the following steps:

- **Identify components**: Identify the GUI components and what needs to be tested.
- **Check visuals**: Verify the visual aspects of the application or website.
- **Write test cases**: Create test cases to verify the working and style of the GUI components.
- **Automate and test**: Automate repetitive test cases and manually test the ones that can't be automated.
- **Report and retest**: Report the defects and retest the application.

43.To create HLR & Test-Case of Web-Based (WhatsApp-web, Instagram-web)

1. HLR & Test-Case of WhatsApp-web	→	Click here
2. HLR & Test-Case of Instagram-web	→	Click here
3. HLR & Test-Case of Art of Testing	→	Click here

44.Write a scenario of only WhatsApp chat messages?	Click Here
45.Write a Scenario of Pen?	
46.Write a Scenario of Pen Stand?	
47.Write a Scenario of Door?	
48.Write a Scenario of ATM?	
49.Write a scenario of Microwave Owen?	
50.Write a scenario of Coffee vending Machine?	
51.Write a scenario of chair?	
52.To Create Scenario (Positive & Negative)? For Gmail (Receiving Gmail)	
53. For Online shopping to buy product (flip-kart)	
53.Write a Scenario of Wristwatch?	
54.Write a Scenario of Lift (Elevator)?	
55.Write a Scenario of WhatsApp Group (generate group)?	