Module-2

* **What is 7 key principles? explain in details**
* **7 key principles**

1. **Testing shows presence of defects.**
2. **Exhaustive testing is impossible.**
3. **Early testing.**
4. **Defect clustering.**
5. **The pesticide paradox.**
6. **Testing is context dependent.**
7. **Absence of errors fallacy.**
8. **testing shows presence of defects**
9. **software testing reduces the presence of defects**
10. **even multiple testing can never ensure that software is 100% bug free.**
11. **Testing talks about the presence of defects and doesn’t talk about the absence of defects.**
12. **exhaustive testing is impossible**
13. **exhaustive testing is impossible means the software can never test at every test cases.**
14. **It can test only some test cases and assume that software is correct and it will produce the correct output in every test cases.**
15. **Early testing**

**a) The defect detected in early phases of SDLC will very less expensive.**

**b) For better performance of software. Start software testing will start at initial phase.**

1. **Defect clustering**

**a) In a project a small number of the module can contain most of the defects.**

**b) Pareto principle to software testing state that 80% of software defect comes from 20% of modules.**

1. **Pesticide paradox**

**a) Repeating the same test cases again and again will not find new bug.**

**b) It is necessary to review the test cases and add or update test cases to find new bug.**

**F. Testing is context dependent**

**a) testing approach depends on context of software developed.**

**b) different types of software need to perform different types of testing.**

**c) the testing of the e-commerce site is different from the testing of the android application.**

**G. absence of errors fallacy**

**a) If a built software is 99% bug free but it does not follow the user requirement then it is unusable.**

**b) It is not only necessary that software is 99% bug free but it is also mandatory to fulfill all the customer requirement.**

**• What is Error, Defect, Bug and failure**

1. **Error:** **we can’t compile or run a program due to coding mistake in a program, if a developer unable to successfully compile or run a program then they call it as an error.**
2. **Defect:** **the variation between the actual result and expected results is known as defect.**
3. **Bug:** **if tester find any mismatch in the application /system in testing phase then they call it as bug.**
4. **Failure: once the product is deployed & customers find any issues then they called the product as a failure product**.

* **Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| **QA (quality assurance)** | **QC (quality control)** | **Tester** |
| **Activities which ensure the implementation of process, procedures & standard in context to verification of developed software and intended requirement** | **Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements** | **Activities which ensure the identification of bugs/error/defects in the software.** |
| **QA is nothing but quality assurance which gives assurance towards quality** | **QC is nothing but quality control, which does not ensure quality, it only exposes lack of quality** | **Focus on actual testing** |
| **Process oriented activities** | **Product oriented activities** | **Product oriented activities** |
| **Preventive activities** | **QC is detection of defect and it is a corrective process.** | **It is a preventive process.** |
| **It is a subset of software test life cycle (STLC).** | **QC can be considered as the subset of quality assurance.** | **Testing is the subset of quality control.** |
| **QA belongs to verification** | **QC belongs to validation** |  |

* **What is exploratory testing?**
* **“Exploratory testing” is the practice of allowing testers to enter a product and find bugs and errors without the help of a script or test cases.**

**• What is traceability matrix?**

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

• **What is Boundary value testing?**

**Boundary value testing: - boundary value analysis is a methodology for designing test cases that concentrates software testing efforts on cases near the limit of valid ranges.**

**• What is Equivalence partitioning testing?**

**Equivalence partitioning: - aim is to treat groups of input as equivalent and to select one representative input to test them all.**

**• What is Integration testing?**

**Integration testing: - testing performed to exposed defects in the interfaces and in the interactions between integrated component or system.**

**• What determines the level of risk?**

* **Determining the level of risk usually involves trying to assess not only the likelihood of an identified risk from actually occurring, but also the potential magnitude the consequences this risk could have on an organization and its stakeholder, should it occur.**

**Level of risk**

* **High risk: - the impact of this risk would be very damaging and potentially non tolerable.**
* **Medium risk: - problems, challenges or glitches labeled as ‘medium risk’ may be tolerable but are certainly not desirable.**
* **Low risk: - ‘low’ risks can almost be classes as inconveniences or minor snags rather than actual threats to the project.**

**• What is Alpha testing?**

* **Alpha testing is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly.**

**• What is beta testing?**

* **Beta testing is performed by real user of the software application in a real environment. Beta testing is one of the types of user acceptance testing.**

• **What is component testing?**

* **Component testing: - the testing of individual software components. Sometimes known as unit testing, module testing or program testing.**

**• What is functional system testing?**

* **Functional system testing: - a requirement that specifies a function that a system or system component must perform.**

**• What is Non-Functional Testing?**

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

• **What is GUI Testing?**

* **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.**

**• What is Adhoc testing?**

* **Adhoc testing is an informal testing type with an aim to break the system. Testers randomly test the application without any test cases or any business requirement document.**

• **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.**

• **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.**

• **Mention what big bang testing is?**

* **In big bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.**

**• 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.**
* **Structured-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**

* **Statement coverage/ segment coverage/ line coverage: - the statement coverage covers only the true condition.**
* **Through statement coverage we can identify the statements executed and where the code is not executed because of blockage.**
* **Decision/branch coverage/ all edges: - it covers both the true and false conditions unlikely the statement coverage.**
* **A branch is the outcome of a decision, so branch coverage simply measures which decision outcomes have been tested.**
* **Condition coverage: -this is closely related to decision coverage but has better sensitivity to the control flow.**
* **Condition coverage measures the condition independently of each other.**
* **Condition coverage reports the true or false outcome of each condition.**

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

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

**Specification-based testing technique is also known as ‘black-box’ or input/output driven testing technique because they view the software as a black-box with inputs and outputs.**

**Techniques of black box testing**

1. **Equivalence partitioning: - aim is to treat groups of inputs as equivalent and to select one representative input to test them all.**
2. **Boundary value analysis: - boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.**
3. **Decision table: - the techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs.**
4. **State transaction testing: - a black box test design technique in which test cases are designed to execute valid and invalid state transitions. Also known as N-switch testing.**

**• Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| **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 the 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 or testers.** | **Sanity testing is usually performed by testers** |
| **Smoke testing is usually documented or scripted.** | **Sanity testing is a subset of acceptance testing.** |
| **Smoke testing is a subset of regression testing.** | **Sanity testing exercise only the particular component of the entire system.** |
| **Smoke testing is like general health check-up** | **Sanity testing is like specialized health** |

**• Difference between verification and Validation**

|  |  |  |
| --- | --- | --- |
| **Sr.no.** | **verification** | **validation** |
| **1.** | **Verification means are we building the software right?** | **Validation means are we building the right product?** |
| **2.** | **Evaluating products of a development phase** | **Evaluating products at the closing of the development phase** |
| **3.** | **The objective is making sure the product is as per the requirement and design specification** | **The objective is making sure that the product meets user’s requirements** |
| **4.** | **Activities included: reviews, meetings and inspections** | **Activities include: black box testing, white box testing and grey box testing.** |
| **5.** | **Items evaluated: plans, requirement, specification, code, and test cases.** | **Items evaluated: actual product or software under test.** |
| **6.** | **Verification is done before validation** | **Validation is done after verification.** |

**• Explain the difference between Functional testing and Non-Functional testing**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Functional testing** | **Non-functional testing** |
| **1.** | **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** |
| **2** | **Functional testing executed first** | **Non functional testing should be performed after functional testing** |
| **3.** | **Manual testing or automation tools can be used for functional testing.** | **Using tools will be effective for this testing** |
| **4.** | **Business requirements are the inputs to functional testing** | **Performance parameter like speed, scalability, are inputs to non-functional testing.** |
| **5.** | **Functional testing describes what the product does.** | **Non-functional testing describes how good the product works.** |
| **6.** | **Easy to do manual testing** | **Tough to do manual testing** |
| **7.** | **Types: unit testing**  **Smoke testing**  **Sanity testing**  **Integration testing**  **White/black box testing**  **User acceptance testing** | **Types: - performance testing**  **Load, stress, volume testing**  **Security testing**  **Installation testing**  **Penetration testing**  **Compatibility testing**  **Migration testing** |

**• Explain types of Performance testing**

**Types of performance testing**

1. **Load testing: - confirms that the system can handle the required number of users and still operate at a high level of performance.**
2. **Stress testing: - intentionally tries to break the software by simulating a number of users that greatly exceeds expectations.**
3. **Volume testing: - checks that the software can handle and process a large amount of data at once without breaking, slowing down or losing any information.**
4. **Spike testing: - this testing evaluates the ability of the application to handle sudden volume increases.**
5. **Endurance testing: - endurance testing evaluates the performance of the system under load over time.**
6. **Scalability testing: - this testing is used to determine your application’s ability to handle increasing amounts of load and processing.**

**• Mention what are the categories of defects?**

**Types of defects**

1. **Data quality/database defects: - deals with improper handling of data in the database.**

**Example: -**

* + - **values not deleted/inserted into the database properly.**
    - **Improper/wrong/null values inserted in place of the actual value.**

1. **Critical functionality defects: - the occurrence of these bugs hampers the crucial functionality of the application.**

**Example: -**

* **exceptions**

1. **functionality defects: - these defects affect the functionality of the application.**

**Example: -**

* + - **all java errors.**
    - **Buttons like save, delete, cancel, not performing their intended functions.**
    - **a missing functionality(or) a feature not functioning the way it is intended to continuous execution of loops.**

1. **Security defects: - application security defects generally involve improper handling of data sent from the user to the application.**

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

**Example: -**

* **Authentication: - accepting an invalid username/password.**
* **Authorization: -accessibility to pages though permission not given.**

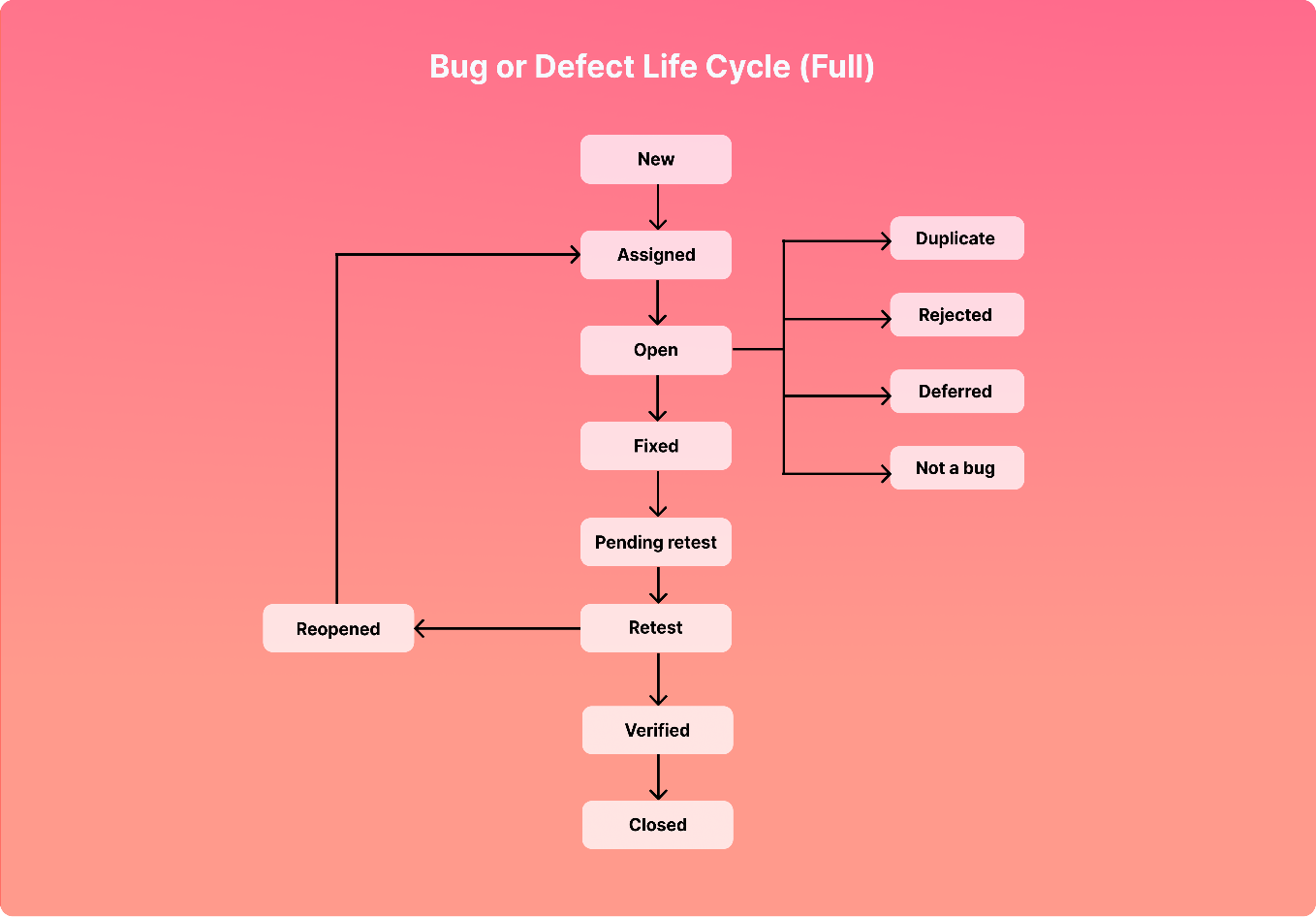
1. **User interface defects: - as the name suggests, the bugs deal with problems related to UI are usually considered less severe**

**Example: -**

* **improper error/warning/UI message**
* **Spelling mistakes**
* **Alignment problems**

**• What is Bug Life Cycle?**

* **The bug life cycle in testing refers to a cycle of defects in which it goes through different states throughout its life.**



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

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **SDLC** | **STLC** |
| **1.** | **SDLC generally related to software development.** | **STLC is generally related to software testing.** |
| **2.** | **Including the development, it focuses on other phases like testing are also included.** | **STLC focuses only on testing the software.** |
| **3.** | **SDLC consist of a total of six steps or phases.** | **STLC consist of only five phases or steps.** |
| **4.** | **In this more members(developers) are required for completing the whole process.** | **In this, less number of members(testers) are needed for completing the whole process.** |
| **5.** | **The development team makes the whole plans and designs.** | **The testing team makes the whole plans and designs.** |
| **6.** | **The objective of SDLC is to complete the successful development of software.** | **The objective of STLC is to complete the successful testing software.** |
| **7.** | **SDLC helps in developing good quality software.** | **STLC helps in making the software defects/error-free.** |
| **8.** | **These phases are completed before the STLC phases.** | **These phases are performed after the SDLC phases.** |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO.** | **TEST SCENARIO** | **TEST CASES** | **TEST SCRIPT** |
| **1.** | **It is a process to test an application with all possible ways.** | **it is a step by step by procedure that is used to test an application.** | **it is a set of instructions to test an application automatically.** |
| **2.** | **Test scenario are an input for the creation of test cases.** | **the term test case is used in the manual testing environment.** | **The term test script is used in automation testing environment.** |
| **3.** | **It reduces the complexity.** | **it is done manually** | **It is done by scripting format.** |
| **4.** | **Test scenario can be a single or a group of test cases.** | **It is development in the form of templates** | **It is development in the form of scripting.** |
| **5.** | **By writing scenarios it will be easy to understand the functionality of an application.** | **Test case templet includes test case ID, test data, test procedure, actual result, expected result etc.** | **In test script we can use different commands to develop script.** |
| **6.** | **These are one-line statements to explain what we are going to test** | **It is a base form to test an application in sequence** | **Once we develop, the script will run it multiple times until the requirement is changed.** |

**• What is the purpose of exit criteria?**

* **The set of generic and specific conditions, agreed, upon with the stakeholders, for permitting a process to be officially completed.**

**• When should "Regression Testing" be performed?**

* **Ideally, regression testing should be performed whenever your codebase has been modified or altered in any way as well as to verify previously discovered issues marked as fixed.**

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

* **A document describing the scope approach, resources, and schedule of intended test activities**

**Follow the seven steps below to create a test plan: -**

1. **Analyze the product**
2. **Design the test strategy**
3. **Define the test objectives**
4. **Define test criteria**
5. **Resource planning**
6. **Plan test environment**
7. **Schedule and estimation**
8. **Determine test deliverables**

**• What is priority?**

**Priority is relative and business-focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is based on the customer requirements.**

**Example: - if the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.**

**• What is severity?**

* **Severity is absolute and customer-focused. It is the extent to which the defect can affect the software. In otherwards it defines the impact that a given defect has on the system.**

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

**• Bug categories are…**

**Bug categories: -**

**1.Data quality/database defects: - deals with improper handling of data in the database.**

**Example: -**

* + - **values not deleted/inserted into the database properly.**
    - **Improper/wrong/null values inserted in place of the actual value.**

**2. Critical functionality defects: - the occurrence of these bugs hampers the crucial functionality of the application.**

**Example: -**

* **exceptions**

1. **functionality defects: - these defects affect the functionality of the application.**

**Example: -**

* + - **all java errors.**
    - **Buttons like save, delete, cancel, not performing their intended functions.**
    - **a missing functionality(or) a feature not functioning the way it is intended to continuous execution of loops.**

1. **Security defects: - application security defects generally involve improper handling of data sent from the user to the application.**

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

**Example: -**

* **Authentication: - accepting an invalid username/password.**
* **Authorization: -accessibility to pages though permission not given.**

1. **User interface defects: - as the name suggests, the bugs deal with problems related to UI are usually considered less severe**

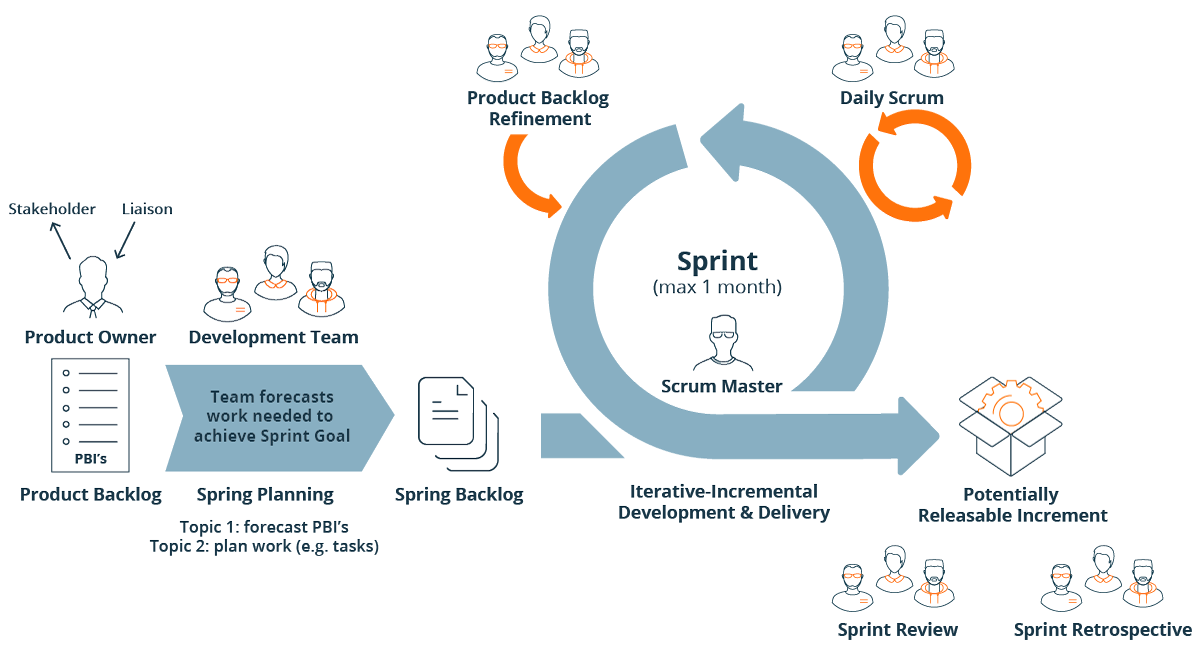
**• Difference between priority and severity**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **severity** | **priority** |
| **1.** | **Defect severity is specified as the degree of impact that a defect has on the operation of the product.** | **Defect priority has specified the order in which the developer should fix a defect.** |
| **2.** | **Severity means the seriousness of the defect in the product functionality.** | **Priority means how soon the bug should be fixed** |
| **3** | **The test engineer determines the severity level of the defect.** | **Priority of defects is decided in discussion with the manager/client.** |
| **4.** | **It is driven by functionality.** | **It is driven by business value.** |
| **5.** | **Severity status is established on the technical aspect of the product.** | **Priority status is established on customer requirements.** |

**• What are the different Methodologies in Agile Development Model?**

**1. Scrum**

**scrum is an agile development method which concentrates particularly on how to manage tasks within a team based development environment.**



**Scrum master: - master is responsible for setting up the team, sprint meeting and removes obstacles to progress**

**Product owner: - the product owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration.**

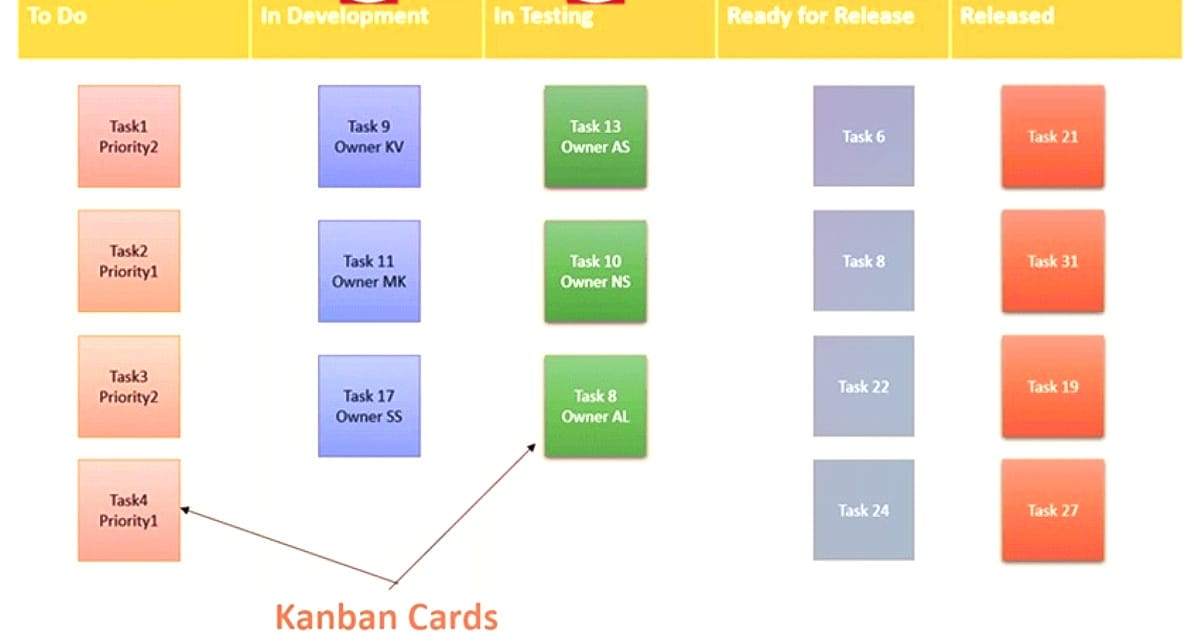
**Scrum team: - team manages its own work and organizes the work to complete the sprint or cycle.**

**2. 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.**

****

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

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **authentication** | **authorization** |
| **1** | **Authentication verifies who the user is.** | **Authorization determines what resources a user can access** |
| **2** | **Authentication works through password, one-time pins, biometric information and other information provide or entered by the user** | **Authorization works through setting that are implemented and maintained by the organization.** |
| **3** | **Authentication is the first step of a good identity and access management process** | **Authorization always takes place after authentication** |
| **4** | **Authentication is visible to and partially changeable by the user** | **Authorization isn’t visible to or changeable by the user** |
| **5** | **Example: - by verifying their identity, employees can gain access to a human resources (HR) application that includes their personal pay information, vacation time** | **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.** |

**One of the main challenges of web application testing is ensuring that your web app works well across different browsers, devices, and operating systems. Different browsers may have different rendering engines, standard support, features and extensions that can affect how your web app looks and behaves.**

**• When to used Usability Testing?**

* **The parameters tested during usability testing are efficiency, effectiveness and satisfaction. If should be performed before any new design is made. This test should be iterated unless all the necessary changes have been made.**

**• What is the procedure for 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.**
* **To create HLR & Test Case of Web Based (WhatsApp web , Instagram) 1. WhatsApp Web :** [**https://web.whatsapp.com**](https://web.whatsapp.com)

**1.FOLDER: - WHATSAAP WEB BASED HLR AND TEST CASE**

**2. FOLDER: - INSTAGRAM WEB BASED HLR AND TEST CASE**

• **To create HLR and Test Case on this Link.** [**https://artoftesting.com/**](https://artoftesting.com/)

1. **FOLDER: - ART OF TESTING HLR AND TEST CASE**

• **To create HLR & Test Case of 1) (Instagram, Facebook) only first page**

**1.FOLDER: - FACEBOOK HLR AND TEST CASE**

**2. FOLDER: - INSTAGRAM MOBILE BASED HLR AND TEST CASE**

**• Write a Scenario of Pen**

* **FILE: - TEST SCENARIO >SHEET: - 1**

**• Write a Scenario of Pen Stand**

* **FILE: - TEST SCENARIO >SHEET: - 10**

**• Write a Scenario of Door**

* **FILE: TEST SCENARIO > SHEET: - 3**

**•Write a Scenario of ATM**

* **FILE: TEST SCENARIO > SHEET: - 8**

**• Write a scenario of Microwave Owen**

* **FILE: TEST SCENARIO > SHEET: - 7**

**• Write a scenario of Coffee vending Machine**

* **FILE: TEST SCENARIO > SHEET: - 9**

**• Write a scenario of chair**

* **FILE: TEST SCENARIO > SHEET: - 2**

**• Write a Scenario of Wrist Watch**

* **FILE: TEST SCENARIO > SHEET: - 6**

**• Write a Scenario of Lift (Elevator)**

* **FILE: TEST SCENARIO > SHEET: - 5**

**• Write a scenario of only WhatsApp chat messages**

* **FILE: TEST SCENARIO > SHEET: - 11 > TABLE: - 2**

**• Write a Scenario of WhatsApp Group (generate group)**

* **FILE: TEST SCENARIO > SHEET: - 11 > TABLE: - 4**

**• Write a Scenario of WhatsApp payment**

* **FILE: TEST SCENARIO > SHEET: - 11 > TABLE: - 6**

**• To Create Scenario (Positive & Negative)**

**FILE: GMAIL RECEIVING MAIL SCENARIOS**

**2. Online shopping to buy product (flipkart)**

**FILE: FLIPKART BUY PRODUCT SCENARIOS**