**DAY 1 - Assignment 1 MRIGANKA PATRA**

**Tasks:**

1. Define the following concepts:

* Functional Testing

Functional testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of functional testing is to ensure that the application behaves as expected and provides the correct output when given specific inputs.

* SDLC (Software Development Life Cycle)

SDLC is a structured process followed by software development teams to design, develop, test, deploy, and maintain software. It ensures software quality and project management discipline.

* Waterfall Model

The Waterfall model is a linear, sequential SDLC approach. Each phase (Requirements → Design → Implementation → Testing → Deployment → Maintenance) must be completed before the next begins. It is best suited for well-defined, unchanging requirements.

* Agile Methodology

Agile is an iterative, flexible SDLC approach where software is developed in small, incremental cycles called sprints. It emphasizes collaboration, customer feedback, and responsiveness to change.

* Scrum vs. Kanban

|  |  |  |
| --- | --- | --- |
| **Feature** | **Scrum** | **Kanban** |
| Definition | A structured Agile framework that uses time-boxed iterations called sprints to manage work. | A flexible Agile methodology that focuses on continuous delivery and visual management of work. |
| Framework | Defined roles (Scrum Master, Product Owner, Development Team) and specific events (sprint planning, review, and retrospective). | No predefined roles; focuses on visualizing work using a Kanban board. |
| Iteration | Work is divided into fixed-length iterations (sprints), typically lasting 2-4 weeks. | Work flows continuously without fixed iterations; tasks are pulled as capacity allows. |
| Change Management | Changes to scope are discouraged during a sprint; adaptations occur at sprint reviews. | Changes can be made at any time, allowing for quick adjustments based on priorities. |
| Metrics | Uses velocity (amount of work completed in a sprint) and burndown charts to track progress. | Focuses on cycle time (time taken to complete a task) and lead time (time from request to delivery). |

2. Identify and describe each of the Agile Sprint activities:

* Sprint Planning
  + Sprint Planning is the first event in the sprint cycle where the team defines what will be accomplished during the upcoming sprint.
  + Purpose:
    - To establish a clear goal for the sprint.
    - To select and prioritize items from the product backlog that will be worked on during the sprint.
  + Participants: The entire Scrum team, including the Scrum Master, Product Owner, and Development Team.
  + Process:
    - The Product Owner presents the prioritized backlog items.
    - The team discusses the items, clarifies requirements, and estimates the effort required.
    - The team commits to a set of items they believe can be completed by the end of the sprint.
* Daily Stand-up
  + The Daily Stand-up (or Daily Scrum) is a short, time-boxed meeting held every day during the sprint.
  + Purpose:
    - To synchronize the team’s activities and plan for the next 24 hours.
    - To identify any obstacles or impediments that may hinder progress.
  + Participants: All members of the Development Team, with the Scrum Master facilitating the meeting.
  + Process:
    - Each team member answers three questions:
      * What did I accomplish yesterday?
      * What will I work on today?
      * Are there any impediments blocking my progress?
    - The meeting typically lasts 15 minutes and is usually conducted standing to encourage brevity.
* Sprint Review
  + The Sprint Review is held at the end of the sprint to inspect the increment and adapt the product backlog if needed.
  + Purpose:
    - To demonstrate the work completed during the sprint to stakeholders.
    - To gather feedback from stakeholders and make necessary adjustments to the product backlog.
  + Participants: The Scrum Team, stakeholders, and anyone interested in the project.
  + Process:
    - The Development Team presents the completed work (increment) to stakeholders.
    - Feedback is collected and discussed, which may lead to updates in the product backlog.
    - The team may also discuss what went well and what could be improved.
* Sprint Retrospective
  + The Sprint Retrospective is a meeting held after the Sprint Review and before the next Sprint Planning.
  + Purpose:
    - To reflect on the past sprint and identify areas for improvement.
    - To create a plan for implementing improvements in the next sprint.
  + Participants: The Scrum Team, including the Scrum Master, Product Owner, and Development Team.
  + Process:
    - The team discusses what went well, what didn’t go well, and what could be improved.
    - Action items are created based on the discussion to enhance team performance in the next sprint.
    - The retrospective is a safe space for open communication and continuous improvement.

3. Choose a user story (e.g. 'User logs into the system and views their profile').

* Write at least 5 functional test cases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Description | Expected Value | Actual Value | Result |
| T01 | Verify that a user can log into the system with valid credentials | User is successfully logged in to and redirected to their profile | User is able to access their profile | Pass |
| T02 | Verify that the system displays an error message when the user enters an invalid username | User should not be able to access his profile and the system is to give error message “Invalid Credentials” | An error message is displayed indicating that the username is incorrect. | Pass |
| T03 | Verify that the system displays an error message when the user enters an incorrect password. | User should not be able to access his profile and the system is to give error message “Invalid Credentials” | An error message is displayed indicating that the username is incorrect. | Pass |
| T04 | Verify that the password recovery link is functional. | User is redirected to the password recovery page. | The password recovery page is not available | Fail |
| T05 | Verify that a confirmation email is sent after requesting a password reset. | A confirmation email is received at the entered email address. | A conformation mail is received | Pass |

* Add 2 "out-of-the-box" negative or edge test cases that go beyond the expected flow.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Test Description | Expected Value | Actual Value | Result |
| T01 | Verify the restriction on the number of OTP verifications during registration. | After the allowed number of attempts (e.g., 3 attempts), the system should display an error message indicating that the user has exceeded the maximum number of OTP verification attempts. | The user was able to request and verify OTPs without any restrictions beyond 3 attempts. The system allowed the user to continue verifying OTPs indefinitely without displaying any error messages or restrictions. | Fail |
| T02 | Verify how the system handles extremely long input values for username and password. | The system should not allow login and should display an appropriate error message indicating that the input exceeds the allowed length | An error message is displayed indicating that the input values exceed the allowed length. | Fail |

4. Create a chart or table listing different testing types (Unit Integration. System. Regression. Acceptance, etc.) with a one-line usage description for each.

|  |  |
| --- | --- |
| **Test Type** | **Description** |
| Unit Testing | Tests individual components or functions of the code to ensure they work as intended. |
| Integration Testing | Evaluates the interaction between integrated components or systems to identify interface defects. |
| Functional Testing | Verifies that the software functions according to the specified requirements and use cases. |
| System Testing | Tests the complete and integrated software system to evaluate its compliance with specified requirements. |
| Acceptance Testing | Determines whether the system meets the acceptance criteria and is ready for deployment. |
| Regression Testing | Confirms that recent changes or enhancements have not adversely affected existing functionalities. |
| Performance Testing | Assesses the speed, scalability, and stability of the application under varying loads. |
| Load Testing | Simulates expected user loads to evaluate how the system behaves under normal and peak conditions. |
| Stress Testing | Tests the system's robustness by pushing it beyond normal operational limits to identify breaking points. |
| Usability Testing | Evaluates the user interface and user experience to ensure the application is user-friendly. |
| Security Testing | Identifies vulnerabilities, threats, and risks in the software to ensure data protection and security. |
| Smoke Testing | Conducts a preliminary check to determine if the major functionalities of the application work correctly. |
| Sanity Testing | Verifies specific functionalities after changes to ensure they are working as expected. |
| Alpha Testing | Conducted by internal teams to identify bugs before releasing the product to external testers. |
| Beta Testing | Performed by a limited number of end-users in a real environment to gather feedback before final release. |
| Exploratory Testing | Involves simultaneous learning, test design, and execution, allowing testers to explore the application freely. |
| Mutation Testing | Evaluates the effectiveness of tests by intentionally introducing faults (mutations) and checking if tests can detect them. |
| Compatibility Testing | Ensures the software works across different devices, operating systems, browsers, and network environments. |
| Ad-hoc Testing | Informal testing without a formal test plan, focusing on finding defects through random checking. |
| Install/Uninstall Testing | Validates the installation and uninstallation processes to ensure they work correctly and do not leave residual files. |
| Retesting | Verifies that specific defects have been fixed after being reported and addressed in the application. |
| Static Testing | Involves reviewing the code, requirements, or documentation without executing the code to find errors. |
| Dynamic Testing | Tests the software by executing the code to validate its behaviour and functionality during runtime. |

5. Summarize the benefits of Agile testing over traditional testing models.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Agile Testing** | **Traditional Testing** |
| Flexibility | Highly flexible; can adapt quickly to changing requirements. | Rigid; changes are often difficult and costly to implement. |
| Feedback Loop | Continuous feedback from stakeholders throughout the process. | Feedback is typically received late in the development cycle. |
| Collaboration | Emphasizes collaboration among cross-functional teams. | Often siloed; limited interaction between developers and testers. |
| Time to Market | Faster delivery through shorter development cycles (sprints). | Longer release cycles due to sequential phases. |
| Defect Detection | Early detection of defects through concurrent testing. | Late detection, often resulting in higher costs to fix issues. |
| User Focus | Prioritizes user experience and feedback in every iteration. | User feedback is often collected after the product is completed. |
| Quality Assurance | Quality is built into the process with ongoing testing. | Quality assurance is a separate phase after development. |
| Team Empowerment | Teams are empowered to take ownership of quality. | Teams may have less autonomy; testing is often seen as a task. |
| Risk Management | Reduces risk through incremental releases and continuous involvement of stakeholders. | Higher risk due to late testing and potential misalignment with user needs. |
| Automation | Encourages the use of automated testing to enhance efficiency. | Automation is often implemented later in the process. |