Agile Testing:

Agile testing is a software testing practice that follows the principles of agile software development, emphasizing continuous testing, close collaboration, and frequent, small releases. It is a flexible and iterative approach where testing begins early and runs concurrently with development, making quality the responsibility of the entire team.

Agile testing helps improve software quality by breaking down complex testing into smaller parts which are more manageable. It allows teams to automate tests more frequently, so they can catch issues early and fix them quickly.

The main focus for Agile testers is customer satisfaction, verifying that the product meets the needs and expectations of the users.

- Agile testing is an informal process that is specified as a dynamic type of testing.
- It is performed regularly throughout every iteration of the Software Development Lifecycle (SDLC).
- Customer satisfaction is the primary concern for agile test engineers at some stage in the agile testing process.

Agile Testing Principles

Agile testing combines traditional testing with development to provide continuous feedback, faster fixes, and better alignment with customer needs.

The main principles of Agile testing focus on:

Shortening feedback iteration: In Agile Testing, the testing team gets to know the product development and its quality for each and every iteration. Thus continuous feedback minimizes the feedback response time, and the fixing cost is also reduced.

Testing is performed alongside Agile testing is not a different phase. It is performed alongside the development phase. It ensures that the features implemented during that iteration are actually done. Testing is not kept pending for a later phase.

Involvement of all members: Agile testing involves each and every member of the development team and the testing team. It includes various developers and experts.

Documentation is weightless: In place of global test documentation, agile testers use reusable checklists to suggest tests and focus on the essence of the test rather than the incidental details. Lightweight documentation tools are used.

Clean code: The defects that are detected are fixed within the same iteration. This ensures clean code at any stage of development.

Constant response: Agile testing helps to deliver responses or feedback on an ongoing basis. Thus, the product can meet the business needs.

Customer satisfaction: In agile testing, customers are exposed to the product throughout the development process. Throughout the development process, the customer can modify the requirements, and update the requirements and the tests can also be changed as per the changed requirements.

Test-driven: In agile testing, the testing needs to be conducted alongside the development process to shorten the development time. But testing is implemented after the implementation or when the software is developed in the traditional process.

Agile Testing Methodologies:

Agile testing methodologies focus on flexibility, collaboration, and continuous improvement. Here are some key Agile testing methods explained in simple terms:

1. Test-Driven Development (TDD): In TDD, tests are written before writing the actual code. This approach involves three steps: writing a unit test, coding to pass the test, and then refactoring the code. It verify that the code is always tested and improved in small, manageable steps.

2. Behavior Driven Development (BDD): BDD is all about understanding and testing how users interact with the application. It focuses on creating features based on user behavior and encourages collaboration between developers, testers, and customers to ensure the software meets user expectations.

3. Exploratory Testing: Here, testers are free to explore the software as they see fit, without following predefined test scripts. This method helps uncover unknown risks and bugs by allowing testers to test the software in creative and flexible ways.

4. Acceptance Test-Driven Development (ATDD): ATDD involves the customer, developers, and testers working together to define the requirements and potential challenges before coding begins. This collaborative effort reduces the chance of errors and helps build software that meets customer needs from the start.

5. Extreme Programming (XP): XP is focused on delivering high-quality software that meets customer needs. It involves practices that emphasize customer involvement, simplicity, and frequent releases to ensure the final product aligns with customer expectations.

6. Session-Based Testing: This method involves structured, time-limited testing sessions. Testers focus on different aspects of the software within a set time frame (usually 45 to 90 minutes), during which they document their findings in a charter document. This ensures focused and efficient testing.

7. Dynamic Software Development Method (DSDM): DSDM is an Agile framework for delivering projects. It provides a set of principles for developers, users, and testers to collaborate and build systems that meet the needs of the business.

8. Crystal Methodologies: The Crystal methodology focuses on the people involved in a project rather than on processes or tools. It adapts to the size and criticality of the project, emphasizing communication, flexibility, and simplicity to deliver results effectively.

Benefits of Agile Testing

Below are some of the benefits of agile testing:

Saves time: Implementing agile testing helps to make cost estimates more transparent and thus helps to save time and money.

Reduces documentation: It requires less documentation to execute agile testing.

Enhances software productivity: Agile testing helps to reduce errors, improve product quality, and enhance software productivity.

Higher efficiency: In agile software testing the work is divided into small parts thus developer can focus more easily and complete one part first and then move on to the next part. This approach helps to identify minor inconsistencies and higher efficiency.

Improve product quality: In agile testing, regular feedback is obtained from the user and other stakeholders, which helps to enhance the software product quality.

Testers Role in Agile Ceremonies:

Testers play an integral role in all Agile ceremonies by providing quality assurance, identifying potential issues, clarifying requirements, and ensuring alignment with user stories and project goals. Their involvement includes participating in planning and reviews, providing feedback, and preparing for and following up on testing activities

SCRUM:

Scrum is a specific framework for implementing the broader philosophy of Agile project management, which emphasizes iterative development, collaboration, and responding to change

Key components of the Scrum framework

Sprints: Short, repeatable cycles (1-4 weeks) during which a team works to complete a set amount of work.

Roles: Clearly defined roles, including a Product Owner, a Scrum Master, and the development team.

Events: Regular, structured meetings such as the Daily Scrum (a 15-minute daily stand-up), Sprint Planning, Sprint Review, and Sprint Retrospective.

Artifacts: Tools like the product backlog and sprint backlog that provide transparency into the work.

Key Terminologies of Scrum

Here are some of key terminologies of Scrum:

Product Backlog: The product backlog is known to be the prioritized list of fixes as well as features that is included in the product's roadmap.

Sprint: Sprint is known as a time-box event which typically lasts from one week to four weeks, in this phase a product increment or iteration occurs.

Development Team: The development team is a group of individuals who are professional in their field and are responsible for product delivery.

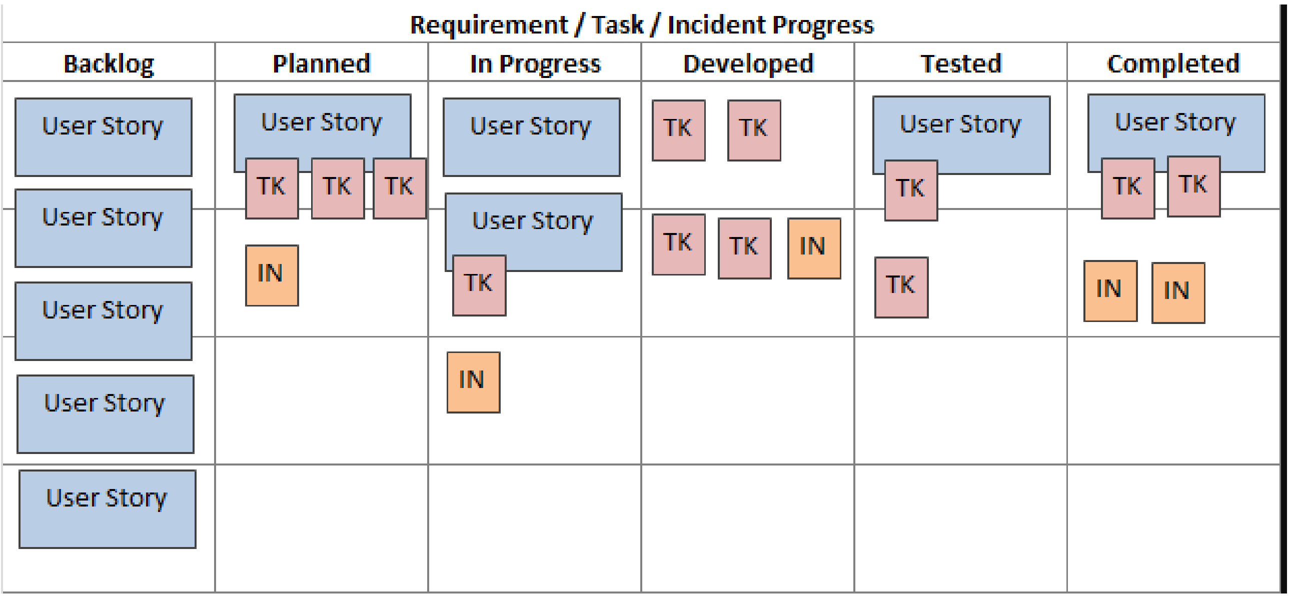
Daily Scrum: Daily scrum is a 15 minute daily meeting used by the development team to integrate activities and to create a plan for the next 24 hours of development.

Sprint Review: The sprint review is held at the end of the sprint in which the team presents all the work that is completed to their stakeholders and the stakeholders give back their feedback.

Sprint Retrospective: The sprint retrospective is a meeting concluded at the end of each sprint so that the team can discuss what went well and what could be improved as well as how to make those improvements.

Kanban:

Kanban is an agile methodology that visually manages workflow, limits work-in-progress, and improves efficiency. It works within the broader Agile philosophy by using a visual board to track tasks as cards moving through project stages (columns)



k

Feature	Scrum	Kanban
Structure	Prescriptive, with fixed events and roles	Flexible, with core principles focused on flow
Delivery	Delivers work in sprints or iterations	Delivers work continuously as it's completed
Change Management	Discourages changes mid-sprint	Allows for changes at any time
Metrics	Velocity (work completed per sprint)	Cycle Time (time to complete a task)