Unit Assignment 2 (Agile Methodologies and Practices)

1. List and briefly explain the three main roles, artifacts in Scrum.

Main roles

Product Owner: Represents the stakeholders and customers. Defines the product backlog (a prioritized list of features and tasks). Ensures the development team works on the most valuable items first.

Scrum Master: Acts as a facilitator, mentor, and servant leader for the Scrum Team. The Scrum Master makes sure that the team follows the methods and principles of Scrum, conducts meetings, removes obstacles, and keeps the team away from extraneous distractions. Their main aim is to increase the productivity of the team and to deliver high-quality increments.

Development Team: A cross-functional team of 5-9 persons is assigned the task of planning, designing, and testing the product increment. The team is self-organizing, responsible for its sprint target, and competent in various fields that include coding, testing, and design.

Artifacts:

Product Backlog: An evolving, prioritized list of project features, tasks, and defects. It represents everything that could be added to the product, as monitored by the Product Owner. The backlog is frequently updated to reflect new requirements and changing priorities.

Sprint Backlog: A subset of the Product Backlog that the team commits to deliver within a single sprint. It includes all the detailed tasks and user stories defined at the Sprint **Planning meeting.** The development team is responsible for the Sprint Backlog.

Increment: A potentially shippable piece of software that is delivered at the end of each sprint. Incremental product development, with the progression becoming evidently incremental on top of one another.

2. List and briefly explain the five main Scrum events.

Sprint Planning: What the team needs to deliver during the next sprint and how.

Daily Scrum: A short daily meeting to discuss progress, plans, and any challenges.

Sprint Review: The team presents the finished work and takes input from the stakeholders.

Sprint Retrospective: It reflects on the previous sprint and identifies improvement areas for the current one.

The Sprint: It is a time-boxed iteration that is 1-4 weeks in length, wherein the team focuses on certain backlog items.

3. Create a sprint timeline for a 2-week sprint, including all Scrum events.

Day 1 (Morning): Sprint Planning - The team selects the tasks to be finished, sets a target for a sprint, and produces the Sprint Backlog.

Day 1-13: Daily Scrum- The team stands up in a 15-minute stand-up in the morning and discusses their progress and impediments.

Day 14 (Morning): Sprint Review - The team presents the completed work to stakeholders and requests their comments.

Day 14 (Afternoon): Sprint Retrospective - The team reviews what is successful and what needs work then adds actionable items to the board for next sprint.

4. Why is the “Definition of Done” critical in Scrum? Provide an example

It establishes quality standards for coding, testing, and documentation, ensuring consistency and completeness across work items.
Example: For a password reset feature, the DoD may include:
Code checked and tested.
Documentation is updated, and the feature was successfully deployed.

5. Explain the concept of User Personas and their role in Agile requirements gathering.

User personas are fictitious profiles that depict the many sorts of users that will engage with the product. They are built on actual facts concerning user behaviour, goals, and obstacles. Personas help the team better understand the target audience and prioritise features that meet unique user demands.

Role in Agile:
Guiding Development: Assists the team in keeping the end user in mind when planning and creating features.
Prioritisation: Determines which features are most important depending on the user's demands.
Empathy Building: Encourages team members to consider the user's point of view, hence increasing user experience.

6. Convert this requirement into a user story “Users should reset passwords via email”.

As a registered user, I want to reset my password, so that I can regain access to my account
This user story addresses a specific user need related to password management

7. Explain the 3Cs (Card, Conversation, Confirmation) concept in user story creation**.**

Card: A card represents a user story printed on a real or digital card. The method for describing a feature from the user's perspective is: "As a [user], I want [some feature] so that [some benefit]."
Conversation: Refers to talks between the Product Owner and the team to clarify specifics of the user narrative. Conversations can inform the story's breadth.
Confirmation: Sets acceptance criteria for the tale to be considered complete.

8. What does the INVEST acronym stand for in relation to user stories, and why is it important?

Independent, **N**egotiable, **V**aluable, **E**stimable, **S**mall, and **T**estable
The **INVEST** criteria ensure that user stories are clear, manageable, and focused on delivering value. **Independent** stories allow for easier prioritization and development. **Negotiable** stories encourage collaboration between the Product Owner and team. **Valuable** stories ensure that each task contributes to the project’s goals, while **Estimable** and **Small** stories help with

accurate planning and delivery within a sprint. **Testable** stories provide clear acceptance criteria, ensuring the team knows when the work is complete.

9. What is story mapping? Illustrate with an e-commerce example
Story mapping organises user tales based on their journey and helps visualise how various features impact the overall product experience. It allows teams to comprehend the product's larger context and prioritise development based on what provides the most value to consumers.

Example of E-Commerce:
Browse Products: Search for goods by category and browse product information.
Add to Cart: Select products and add them to your shopping basket.
Checkout: Enter your shipping information and finish the payment.

10. Explain following Agile methodologies, mention at least 1 unique feature and provide 2 use cases for each methodology where they shall be most suitable
a. Scrum
b. Lean
c. XP
d. Kanban
a. Scrum: Scrum is one of the most widely used Agile methodologies, especially for managing software development projects. It focuses on delivering small increments of the product through well-defined roles, events, and artifacts .
Unique Feature: Scrum is structured around time-boxed sprints and predefined roles such as the Product Owner, Scrum Master, and Development Team .
Use Cases: Software product development, educational platforms.

b. Lean: Lean Software Development is an Agile methodology that takes its principles from Lean manufacturing, originally developed by Toyota. It focuses on eliminating waste, improving flow, and delivering value as quickly as possible
.Unique Feature: Lean emphasizes identifying and eliminating the "Eight Wastes," which can hinder the process
.Use Cases: Manufacturing systems, process optimization in startups.

c. XP (Extreme Programming):
Extreme Programming (XP) Improves the quality of software as well as their responsiveness to customer's changing needs. XP also stresses technical, teamwork, and rapid releases
Distinctive Practice: Core practices are Test-Driven Development, Pair Programming, and Continuous Integration
. Use cases: High-risk projects, real-time systems.

d.Kanban is an Agile method which focuses on visualization of work, limiting the WIP (Work in progress), and constant optimization of workflow tasks.
Unique Feature: Kanban is represented on visual boards showing the tasks under various stages: "To Do," "In Progress," and "Done."

Use Cases: IT support, marketing campaigns, maintenance tasks.

11. Explain the concept of Test-First Development and its benefits. Give example with a code snippet.

Concept of Test-First Development (TFD):
Test-First Development (TFD) is a method in which tests are developed prior to actual production code. The core idea is to make the code testable right from the beginning. TFD emphasizes writing tests first but doesn't always adopt an iterative approach strictly. The design and implementation can be more front-end.
Advantages of Test-First Development:
Improves Code Quality: Tests ensure that the code is up to standards prior to the implementation phase.
Reduces Bugs: Developers are able to identify errors earlier because tests were written prior to the code, resulting in less defects in production.
Improves Maintainability: Code is simpler to refactor and extend with good tests, so future changes are safer.
Promotes Improved Design: Testing prior to implementation results in better-organized, modular, and loosely coupled code.

```python
import unittest

class TestCalculator(unittest.TestCase):
   def test_add(self):
       self.assertEqual(Calculator.add(2, 3), 5)

# Run the tests
if __name__ == '__main__':
   unittest.main()
```

Explanation:

1. The test case is written before implementing the `add` method.
2. Running this test will **fail initially** since the `Calculator` class does not exist yet.
3. The developer then writes the following **minimal implementation** to make the test pass:

```python
class Calculator:
   @staticmethod
   def add(a, b):
       return a + b
```

After implementing the method, the test is run again, and **it should now pass**.
Once the test passes, the developer can refactor the code if needed, ensuring the test still succeeds

12. Describe Pair Programming and its advantages in Agile development.
Pair Programming is a central practice in Extreme Programming (XP) where two programmers sit at one machine. One programmer, the "driver," types in the code, and the other, the "observer" or "navigator," checks each line of code as it is typed. The roles are often reversed to keep both programmers interested and exchange knowledge.
Benefits of Pair Programming:
Higher Code Quality: Regular review picks up errors early, resulting in fewer bugs in production.
Faster Problem Solving: Two heads are better than one when it comes to solving complex problems.
Knowledge Sharing: New developers learn from senior ones, resulting in improved skills across the team.
Better Design Decisions: Pairing enables live discussion, which results in improved architecture and code design.

13. Describe the key components of a Kanban board and how it facilitates Agile project management.
A Kanban board is an Agile visual tool that helps to track tasks and improve workflow by restricting Work in Progress (WIP)
.
The Major Elements of a Kanban Board:
- To Do: The planned but yet to be begun tasks.
- In Progress: The work being undertaken currently.
- Done: The accomplished tasks that satisfy the acceptance criteria

.
How It Aids Agile Project Management:
- Visualizes Workflow: Offers a real-time, transparent picture of tasks, which helps the teams stay organized.
- Limits Work in Progress (WIP): Does not overload the teams with lots of things simultaneously, helping increase focus.
- Better Collaboration: Allows everyone to get an instant status of ongoing work, which reduces miscommunications within teams.

14. Compare the responsibilities of Architect, a Release Manager, a Project Manager, a Product Manager in Agile projects.

- **Architect:** Defines the system architecture and ensures scalability and quality.
- **Release Manager:** Plans and coordinates software releases, ensuring quality and risk management.
- **Project Manager:** Oversees timelines, manages risks, and coordinates across multiple teams.
- **Product Manager:** Defines product vision and aligns it with business goals.

15. What is code coverage, and how do tools help in measuring it?

Code coverage is a metric to define the extent to which the source code of a program is covered by a specific test suite.

It gives a measure of how much code is being executed during the test and indicate untested areas of the codebase.

Major Code Coverage Metrics:

- Line Coverage: The ratio of lines of code run by the tests.
- Branch Coverage: The ratio of branches (e.g., if-else statements) exercised.
- Function Coverage: The ratio of functions/methods called during testing

Common Code Coverage Tools:

- Cobertura: A popular Java code coverage tool that determines the percentage of code executed during tests.
- JaCoCo: An open-source tool for calculating Java code coverage.
- Coverage.py: A Python tool that reports which parts of the code have been run

16. An edtech team struggles with frequent bugs. Which XP practices would you recommend?

For an edtech development team that is plagued with recurring bugs, Extreme Programming (XP) methods such as Test-Driven Development (TDD) and Pair Programming can be a huge boon to software quality.

- Test-Driven Development (TDD): Testing first before coding guarantees correctness and keeps defects out of production.
- Pair Programming: Having two coders collaborate enhances code quality by fixing errors early.
- Continuous Integration (CI): Regular integration of code lets bugs get caught early and be fixed soon.

These habits minimize mistakes, enhance cooperation, and provide a quality learning platform for the users.

17. A fintech startup uses Scrum but faces scope creep. How would Kanban solve this?

A fintech business that is being affected by scope creep (when changes that cannot be controlled continuously add to the scope of work) can gain from Kanban since it constrains Work in Progress (WIP) and makes sure new work is introduced only when in-progress work is done.

- Restricts WIP: Makes sure the team completes current work before working on new work, to avoid a surfeit of work.
- Enhances Focus: Encourages ongoing delivery by making the workflow effective and transparent.

- Increases Visibility: A Kanban board tracks all work, and it's quicker to pick up on unwanted additions to scope.
- Through the use of Kanban, the startup will avoid getting work in excess, maintaining priority focus on work and providing quality financial solutions.

18. Design acceptance criteria for: "Search flights by price range."

Acceptance criteria establish the requirements that a user story should fulfill in order to be complete.

User Story: "As a traveler, I want to search for flights in a given price range so that I can find flights that are within my budget."

Acceptance Criteria:

- Users can input a minimum and maximum price in the system.
- Flights within the given range will be shown in the results.
- Search results are updated in real time depending on the input price range.
- If there are no flights in the chosen range, the system shows a corresponding message.
- The filtered results can be sorted by price, duration, and airline by users.
- These conditions guarantee the feature operates as intended and offers a smooth search experience for users.

.