Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	2
Title:	Implementing Project Using Scrum Method on JIRA Tool in Software Engineering
Date of Performance:	03 - 08 - 2023
Roll No:	9595 ,
Team Members:	Atharva Dalvi , Reanne Dcosta , Nicole Falcao

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01(rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Lab Experiment 02

Experiment Name:
Implementing Project Using Scrum Method on JIRA Tool in Software Engineering
Objective:
The objective of this lab experiment is to introduce students to the Scrum framework and its implementation using the JIRA tool. Students will gain practical experience in managing a software project using Scrum principles and learn how to utilize JIRA as a project management tool to track and organize tasks, sprints, and team collaboration.
Introduction: Scrum is an agile project management methodology that promotes iterative development, collaboration, and continuous improvement. JIRA is a widely used tool that supports Scrum practices, providing teams with features to plan, track, and manage software projects effectively.
Lab Experiment Overview:
1. Introduction to Scrum: The lab session begins with an overview of the Scrum framework, including its roles (Product Owner, Scrum Master, and Development Team), events (Sprint Planning, Daily Standup, Sprint Review, and Sprint Retrospective), and artifacts (Product Backlog, Sprint Backlog, and Increment).
2. JIRA Tool Introduction: Students are introduced to the JIRA tool and its capabilities in supporting Scrum project management. They learn to create projects, epics, user stories, tasks, and sub-tasks in JIRA.
3. Defining the Project: Students are assigned a sample software project and create a Product Backlog, listing all the required features, user stories, and tasks for the project.
4. Sprint Planning: Students organize the Product Backlog into Sprints, selecting user stories and tasks for the first Sprint. They estimate the effort required for each task using story points.
5. Implementation in JIRA: Students use the JIRA tool to create a Sprint Backlog, add the selected user stories and tasks, and assign them to team members.
6. Daily Standup: Students conduct a simulated Daily Standup meeting, where they update the progress of their tasks and discuss any impediments they are facing.
7. Sprint Review and Retrospective: At the end of the Sprint, students review the completed tasks, demonstrate the implemented features, and gather feedback from their peers. They also conduct a Sprint Retrospective to identify areas of improvement for the next Sprint.
8. Continuous Iteration: Students continue implementing subsequent Sprints, repeating the Sprint Planning, Daily Standup, and Sprint Review & Retrospective events.
9. Conclusion and Reflection: At the end of the lab experiment, students reflect on their experience with Scrum and JIRA, discussing the advantages and challenges they encountered during the project.
Learning Outcomes:
By the end of this lab experiment, students are expected to:

- Understand the Scrum framework and its principles in agile project management.
- Gain practical experience in using the JIRA tool for project management in a Scrum environment.
- Learn to create and manage Product Backlogs, Sprint Backlogs, and track progress using JIRA.
- Develop collaborative skills through Daily Standup meetings and Sprint Reviews.

- Gain insights into the iterative nature of software development and the importance of continuous improvement.

Pre-Lab Preparations: Before the lab session, students should familiarize themselves with the Scrum framework and the basics of the JIRA tool. They should review Scrum roles, events, and artifacts, as well as the features of JIRA relevant to Scrum implementation.

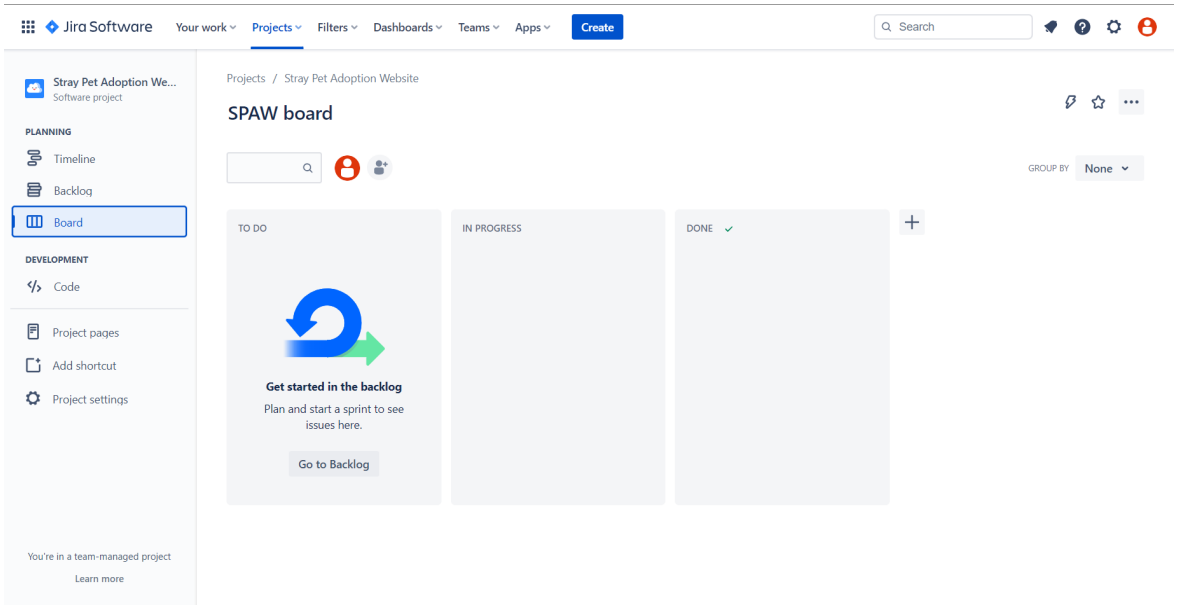
Materials and Resources:

- Computers with internet access for accessing the JIRA tool
- Project brief and details for the sample software project
- Whiteboard or projector for explaining Scrum concepts

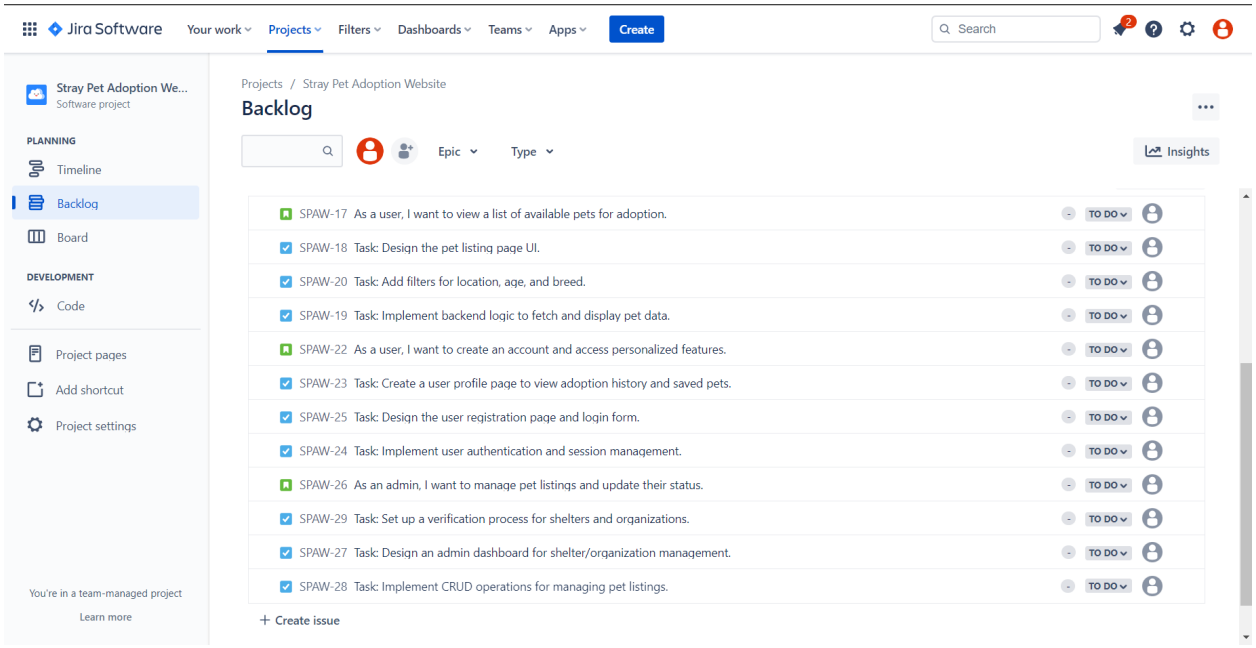
Conclusion:

The lab experiment on implementing a project using Scrum on the JIRA tool offers students a hands-on experience in agile project management. By utilizing Scrum principles and JIRA's capabilities, students learn to collaborate effectively, manage tasks efficiently, and adapt to changing requirements. The practical exposure to Scrum and JIRA enhances their understanding of agile methodologies, equipping them with valuable skills for real-world software development projects. The lab experiment encourages students to embrace the agile mindset, promoting continuous improvement and customer-centric software development practices.

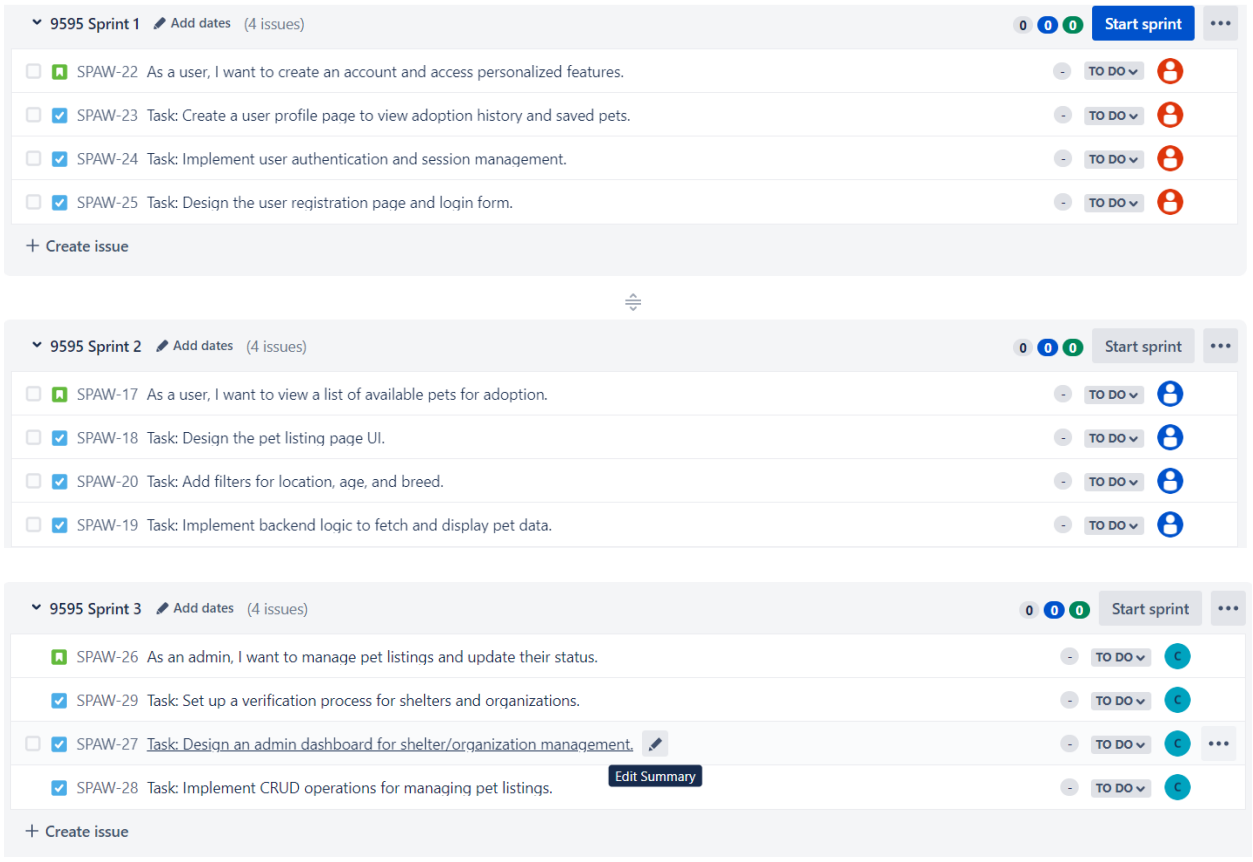
1. **Create a Scrum Board:**



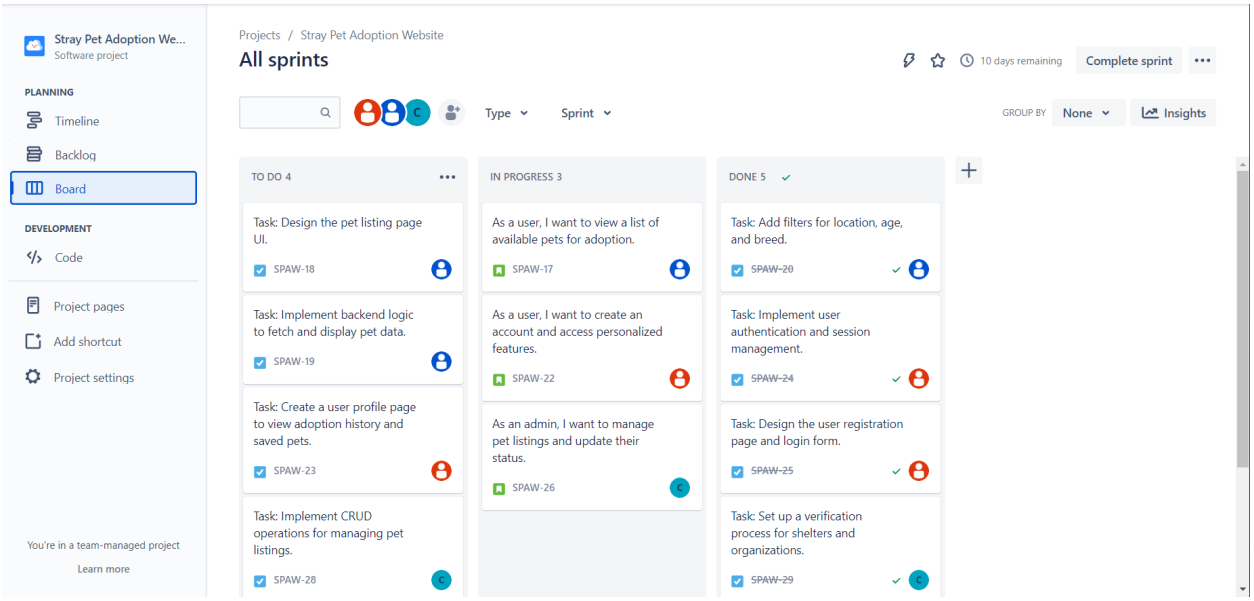
2. Define the Sprint Backlog:



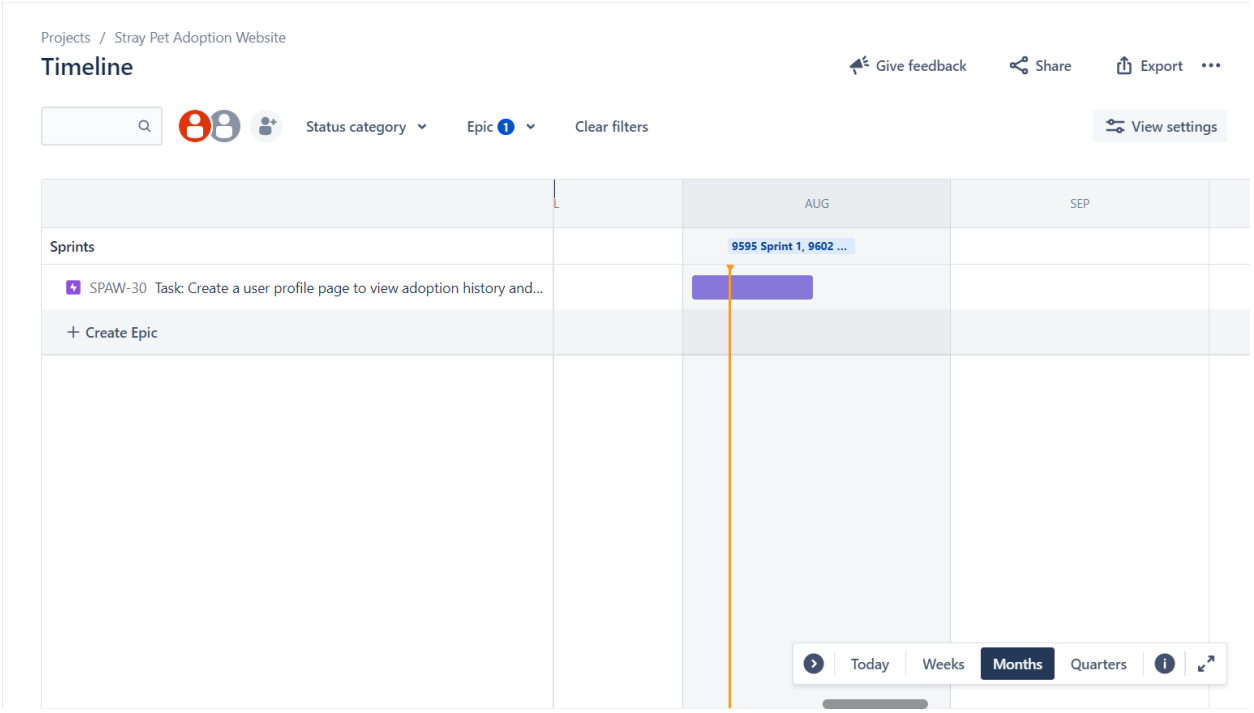
3. Sprint Planning:



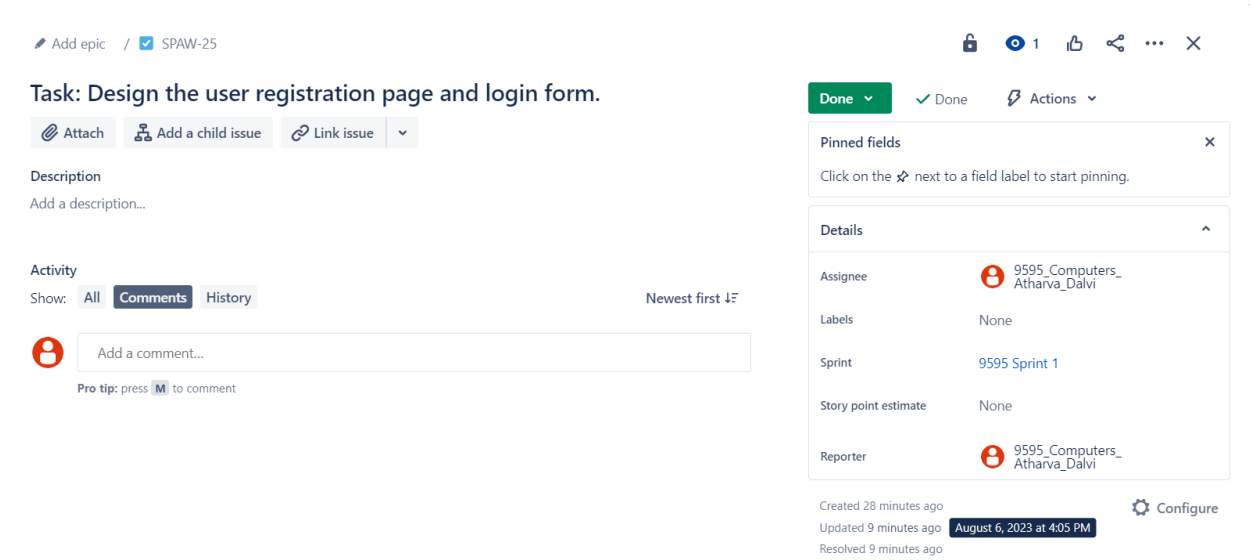
5. Sprint Status:



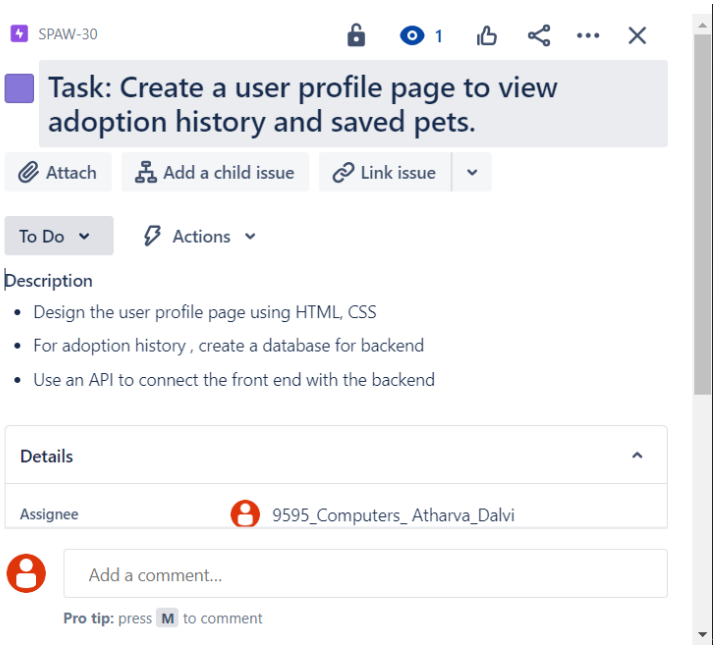
4. Timeline:



5. Tasks:



7. Epics:



POSTLABS:

a. **Assess the effectiveness of the Scrum framework for managing software development projects compared to traditional project management methodologies.**

The Scrum framework for managing software development projects has proven to be highly effective compared to traditional project management methodologies. Here are some key points of comparison:

1. Flexibility:
   Scrum is highly flexible and adaptable, allowing teams to respond quickly to changing requirements and priorities. Traditional methodologies often follow a rigid plan, which can lead to difficulties in accommodating changes.

   - User Profiles: The website should allow users to create personalized profiles with options to add information about their preferences, experience with pets, and any existing pets they have.
   - Upload Options: Pet owners should have flexibility in uploading pictures and videos of their pets to showcase their personalities, which can help potential adopters connect with the animals emotionally.

2. Transparency:
   Scrum encourages transparency through its various ceremonies, such as daily stand-ups and sprint reviews. This transparency helps identify and address issues promptly, promoting a more productive and accountable team environment.

   - Clear Information on Pets: Provide comprehensive and accurate information about each pet available for adoption. This includes details about the pet's age, breed, size, health condition, temperament, and any behavioral issues or special needs.
   - Up-to-Date Profiles: Ensure that the pet profiles are regularly updated to reflect the most current information. Pets may undergo changes in health or behavior, and the website should reflect these updates promptly.

b. **Analyze a Sprint Backlog in JIRA and identify any potential bottlenecks or issues that might hinder the team's progress during the sprint.**

- Overloaded Sprint Backlog: If the Sprint Backlog contains an excessive number of user stories or tasks, it can lead to overload and reduced focus. The team might struggle to complete all the planned work within the sprint timeframe.

- Unclear or Ambiguous User Stories: If the user stories in the Sprint Backlog are unclear, lack detailed acceptance criteria, or have ambiguities, the team may face delays in understanding what needs to be done, leading to slower progress.

- Dependency Hell: If the user stories have complex interdependencies, completing one task might be dependent on another task being finished first. This can create bottlenecks and hinder progress if the dependencies are not managed effectively.

- Skill Gaps or Over-specialization: If the team lacks the necessary skills to handle specific tasks or if certain team members are over-specialized, it can lead to delays when those tasks are assigned to individuals who are not equipped to handle them efficiently.

c. Evaluate the role of the Scrum Master in handling conflicts within the development team and resolving impediments to maintain a smooth project flow.

The Scrum Master plays a crucial role in handling conflicts within the development team and resolving impediments to maintain a smooth project flow in the Scrum framework. In short:

- Conflict Resolution: The Scrum Master acts as a mediator and facilitator, helping team members address conflicts constructively. They promote open communication, active listening, and empathy to foster a positive team environment. By addressing conflicts promptly, the Scrum Master ensures that the team stays focused on achieving the sprint goals.

- Impediment Removal: The Scrum Master is responsible for identifying and removing impediments that hinder the team's progress. They work proactively to eliminate roadblocks, such as resource constraints, technical challenges, or external dependencies, enabling the team to work efficiently and meet their commitments.

- Coaching and Support: The Scrum Master coaches and supports the development team to enhance their productivity and self-organization. They encourage continuous improvement, provide guidance on Scrum practices, and help the team optimize their workflow.

- Facilitating Ceremonies: The Scrum Master facilitates various ceremonies, such as Daily Stand-ups, Sprint Planning, Sprint Review, and Sprint Retrospective. By ensuring these

events run smoothly and effectively, the Scrum Master enables the team to collaborate, reflect, and adapt to changes.

- Promoting Collaboration: The Scrum Master fosters collaboration among team members, stakeholders, and product owners. They encourage cross-functional teamwork, promote transparency, and create an environment where everyone's contributions are valued.