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
Date of Performance:	03-08-23
Roll No:	9596
Team Members:	9596 Reanne Dcosta, 9595 Atharva Dalvi, 9602 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:

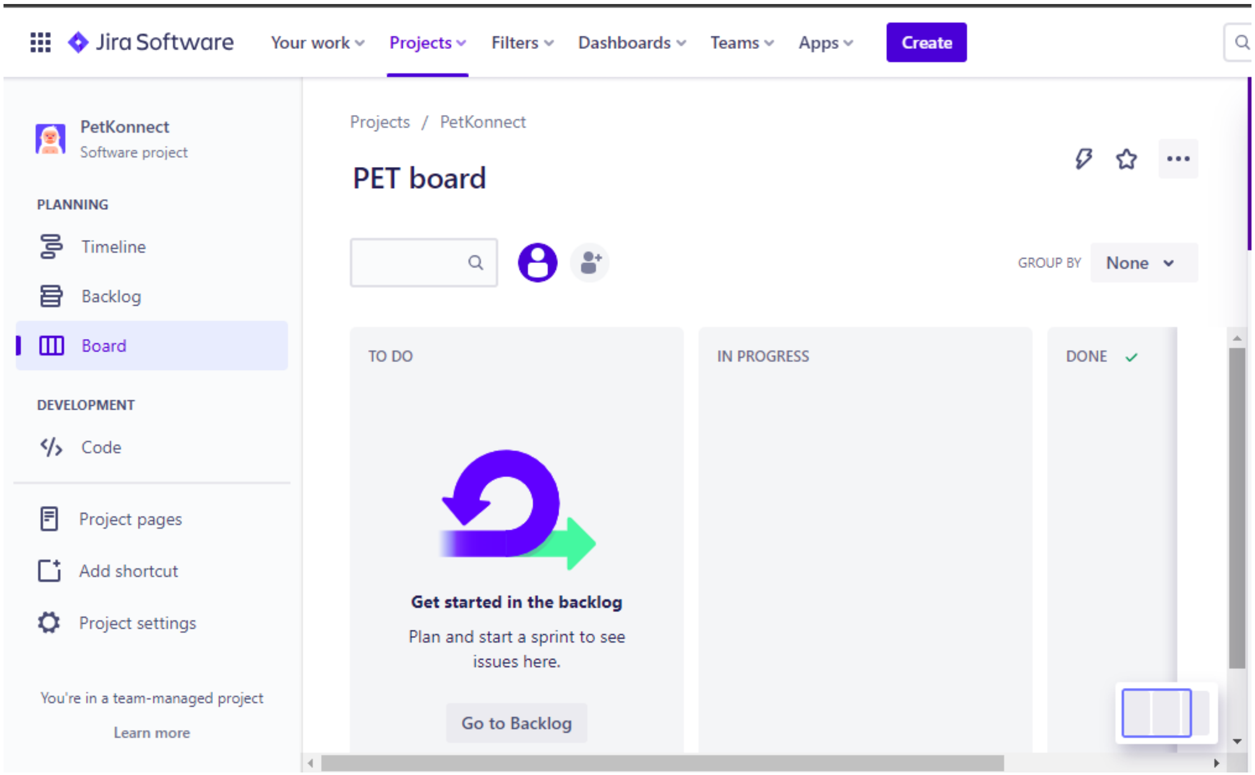
- Computers with internet access for accessing the JIRA tool
- Project brief and details for the sample software project
- Whiteboard or projector for explaining Scrum concepts

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.

I Defining the Project

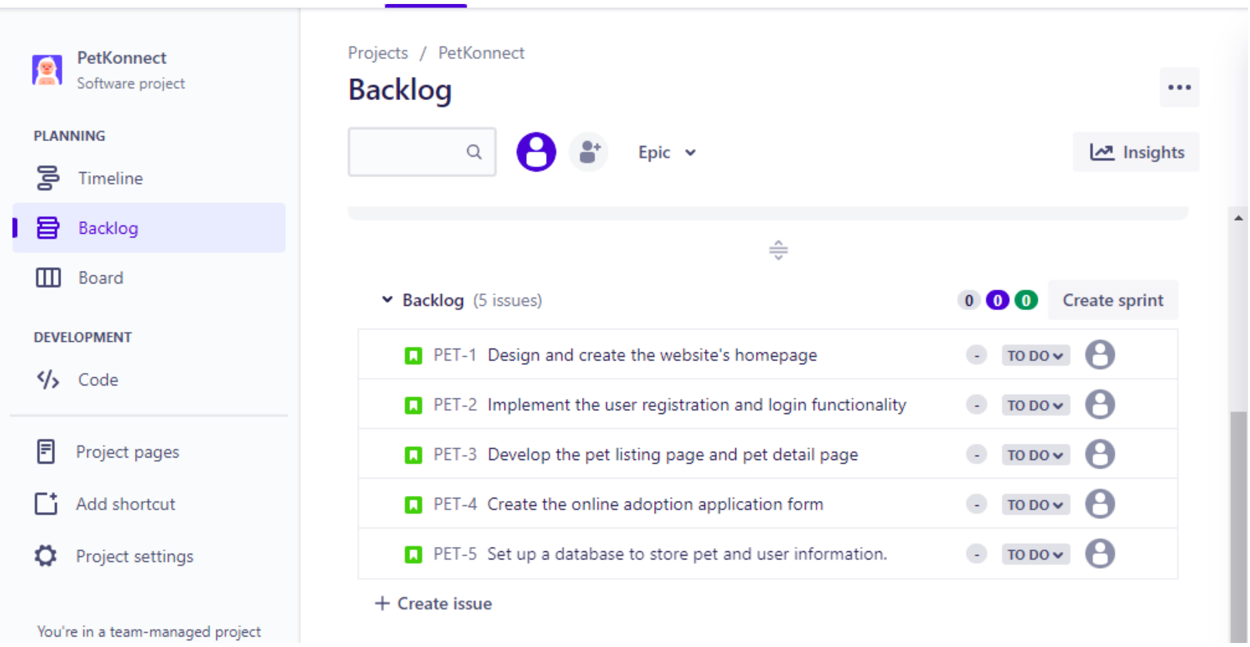
1. Creating a Scrum Project



Get started in the backlog

Plan and start a sprint to see issues here.

2. Creating backlogs

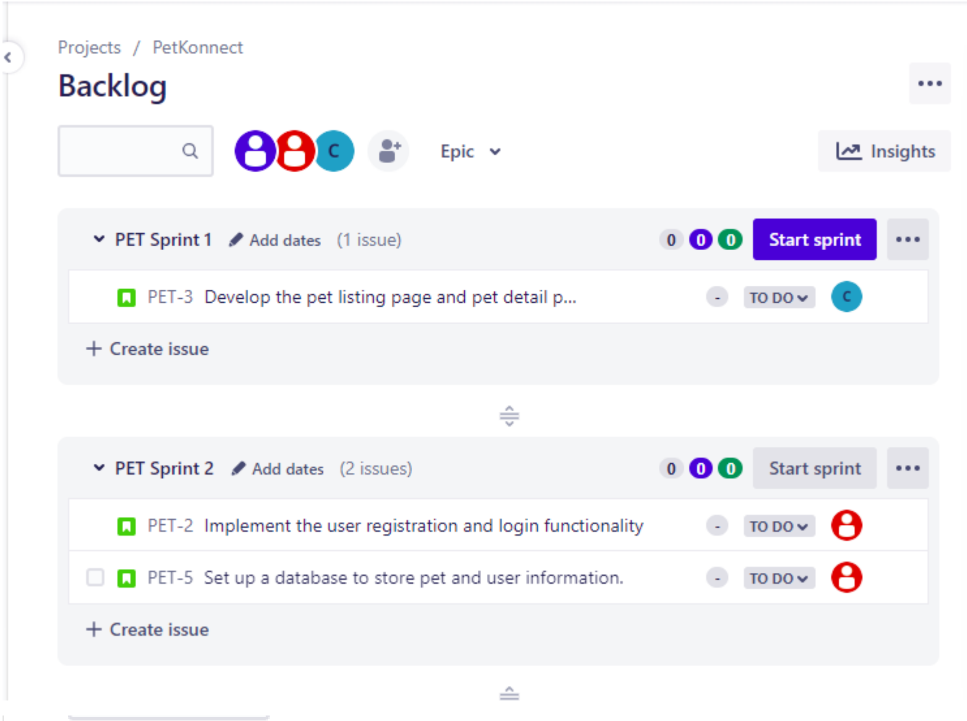


Backlog (5 issues)

PET-1	Design and create the website's homepage	-	TO DO
PET-2	Implement the user registration and login functionality	-	TO DO
PET-3	Develop the pet listing page and pet detail page	-	TO DO
PET-4	Create the online adoption application form	-	TO DO
PET-5	Set up a database to store pet and user information.	-	TO DO

II Sprint Planning

1. Creating Sprints and assigning to team members



Projects / PetKonnect

Backlog

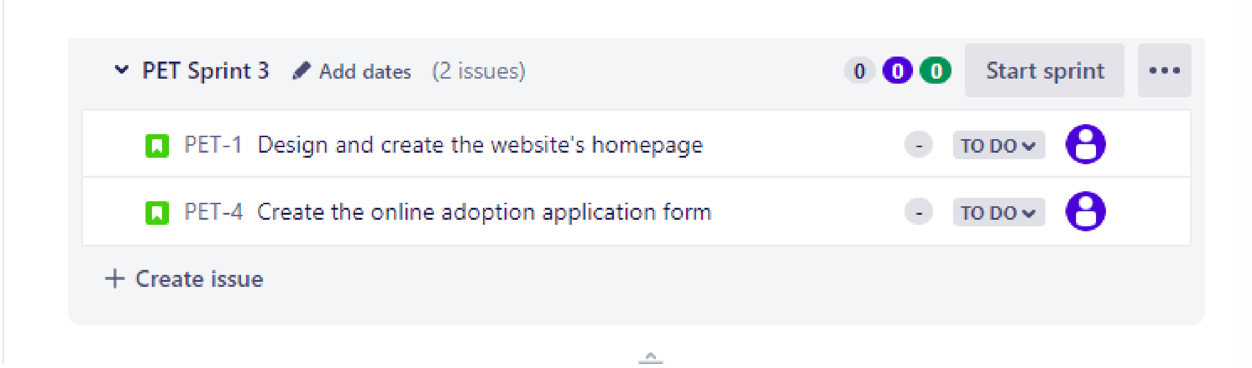
PET Sprint 1 Add dates (1 issue) 0 0 0 Start sprint

PET-3 Develop the pet listing page and pet detail p... - TO DO C

PET Sprint 2 Add dates (2 issues) 0 0 0 Start sprint

PET-2 Implement the user registration and login functionality - TO DO

PET-5 Set up a database to store pet and user information. - TO DO

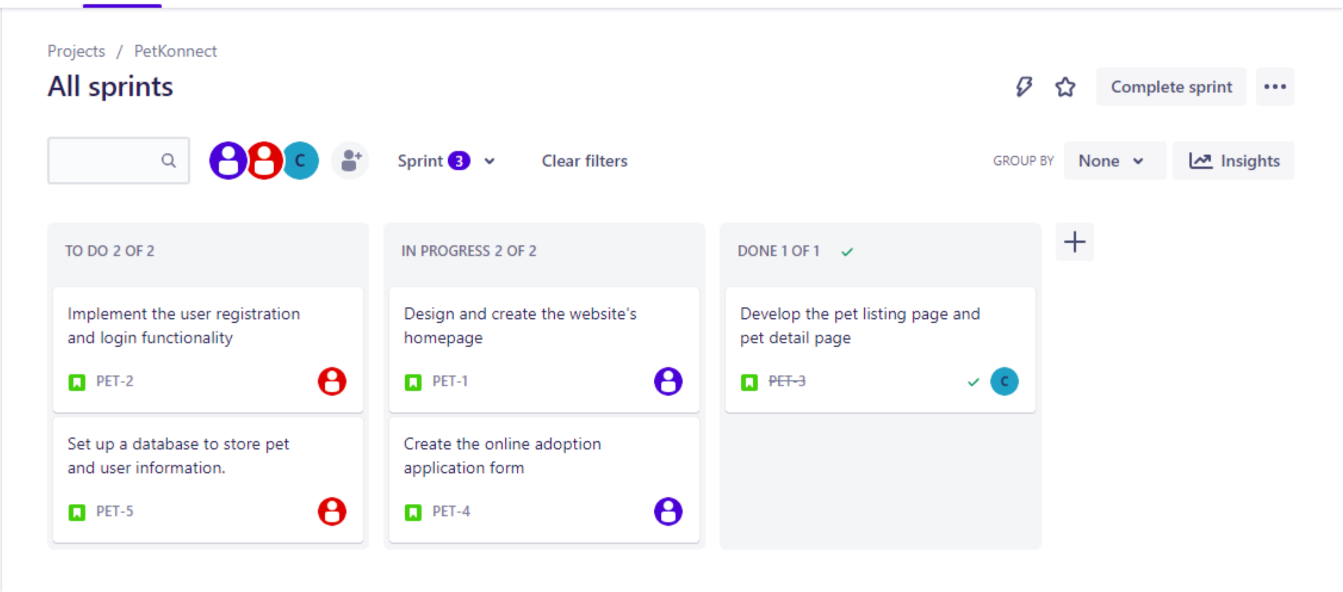


PET Sprint 3 Add dates (2 issues) 0 0 0 Start sprint

PET-1 Design and create the website's homepage - TO DO

PET-4 Create the online adoption application form - TO DO

2. Status of Sprints



Projects / PetKonnect

All sprints

TO DO 2 OF 2

Implement the user registration and login functionality

PET-2

Set up a database to store pet and user information.

PET-5

IN PROGRESS 2 OF 2

Design and create the website's homepage

PET-1

Create the online adoption application form

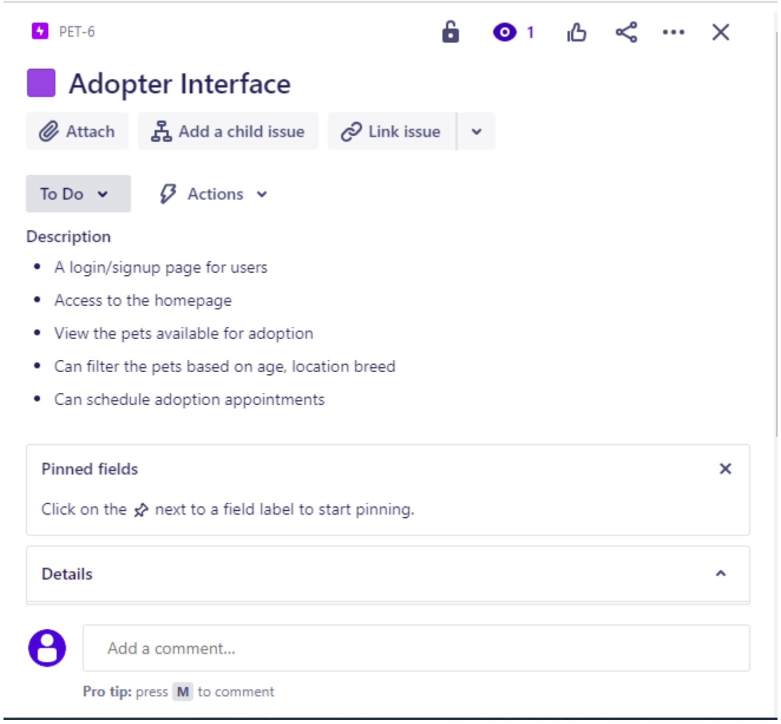
PET-4

DONE 1 OF 1 ✓

Develop the pet listing page and pet detail page

~~PET-3~~ ✓

Epic



PET-6

Adopter Interface

Description

- A login/signup page for users
- Access to the homepage
- View the pets available for adoption
- Can filter the pets based on age, location breed
- Can schedule adoption appointments

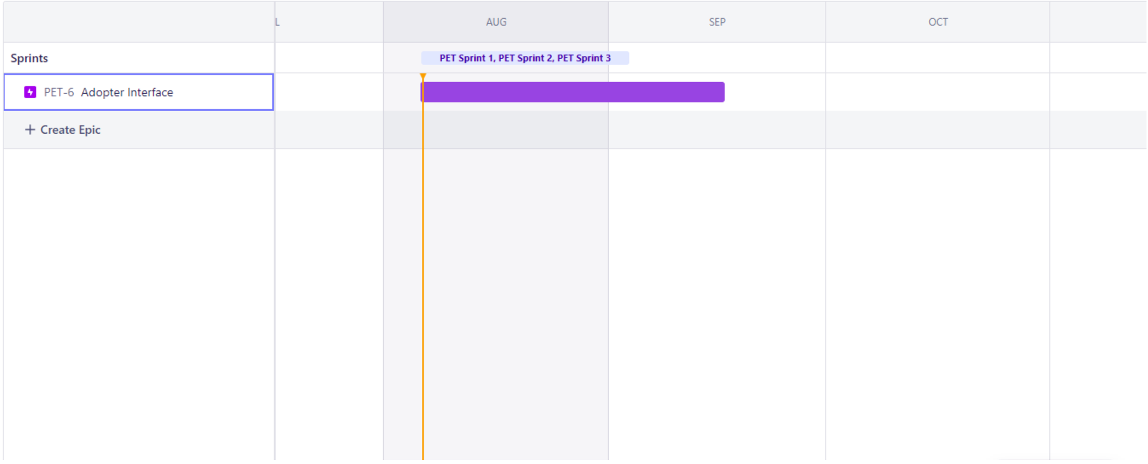
Pinned fields

Click on the next to a field label to start pinning.

Details

Pro tip: press M to comment

Timeline


L
AUG
SEP
OCT
Sprints
PET Sprint 1, PET Sprint 2, PET Sprint 3
PET-6 Adopter Interface
+ Create Epic

POSTLAB

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

The effectiveness of the Scrum framework versus traditional project management methodologies depends on the context of the project:

Adaptive and Complex Projects: Scrum is highly effective for complex projects with evolving requirements, uncertain environments, and high customer involvement.

Innovation and Time-to-Market: Scrum's iterative approach facilitates faster delivery, making it well-suited for innovative projects and products requiring quick time-to-market.

Small to Medium-Sized Projects: Scrum's focus on self-organizing teams and streamlined processes is particularly effective for small to medium-sized projects.

Predictable and Stable Projects: Traditional methodologies may be more suitable for projects with stable and well-understood requirements, predictable outcomes, and low customer involvement.

Regulatory and Compliance Projects: Traditional methodologies, with their emphasis on documentation and change control, can be better aligned with regulatory and compliance-driven projects.

Large and Complex Projects with Clear Requirements: Traditional methodologies may provide better control and predictability in managing large projects with well-defined requirements and less frequent changes.

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

Some common bottlenecks and issues :

Overloaded Resources: If certain team members are assigned too many tasks compared to others, uneven workload distribution can lead to delays and hinder progress.

Blocked Tasks: Tasks that are blocked due to dependencies on other tasks or external factors. Blocked tasks can halt progress and require immediate attention.

Scope Creep: Monitor for any new tasks or user stories that are added to the Sprint Backlog during the sprint. Unplanned scope changes can impact the team's ability to complete all committed work.

Unclear or Ambiguous Tasks: Ensure that each task in the Sprint Backlog has clear and unambiguous descriptions. Unclear tasks can lead to misunderstandings and delays.

Lack of Prioritization: Verify that tasks are prioritized appropriately, with the most critical items at the top. Poor prioritization can result in focusing on less valuable tasks while more important ones are left unaddressed.

External Interruptions: Consider any external factors or interruptions that might affect the team's ability to work on the Sprint Backlog items, such as urgent production issues or support requests.

Dependency Bottlenecks: Identify tasks that have many dependencies or are dependent on other tasks. Delays in any of these tasks can cause a ripple effect on the entire Sprint Backlog.

Unrealistic Estimates: Review task estimates and verify if they are realistic and aligned with the team's capacity. Underestimating tasks can lead to incomplete work, while overestimating can result in underutilized resources.

Communication Issues: Check for any communication gaps within the team or with stakeholders. Miscommunication can lead to incorrect assumptions and delays.

Lack of Skill Balance: Ensure that the team has the necessary skills and expertise to handle the tasks in the Sprint Backlog. A lack of required skills can slow down progress.

3. 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 critical role in handling conflicts within the development team and resolving impediments to maintain a smooth project flow in the Scrum framework.

1. Conflict Resolution:

- **Mediator:** The Scrum Master acts as a mediator in case of conflicts or disagreements among team members. They facilitate open communication and ensure that all team members have a chance to express their perspectives.
- **Active Listening:** The Scrum Master actively listens to all parties involved, understanding their concerns and emotions. They create a safe environment where team members can discuss issues without fear of reprisal.
- **Conflict Facilitation:** By facilitating discussions, the Scrum Master encourages the team to find solutions collaboratively. They guide the team towards a resolution that aligns with the Scrum values and principles.
- **Empowerment:** The Scrum Master empowers the team to take ownership of their conflicts and encourages them to find their own resolutions whenever possible.

2. Impediment Resolution:

- **Identification**: The Scrum Master identifies impediments that hinder the team's progress. These impediments can be related to process, communication, resources, or external dependencies.
- **Removal:** The Scrum Master works diligently to remove or minimize impediments. They collaborate with stakeholders and other teams to address issues that are beyond the team's control.
- **Obstacle Tracking:** The Scrum Master maintains a visible list of impediments, often referred to as the "Impediment Backlog." This list is regularly reviewed and updated during Scrum ceremonies.
- **Coaching and Support:** The Scrum Master provides coaching and support to team members, helping them navigate challenges and offering guidance on overcoming impediments.

3. Facilitating Project Flow:

- **Process Facilitation:** The Scrum Master ensures that the Scrum events (Sprint Planning, Daily Stand-ups, Sprint Review, Sprint Retrospective) are conducted effectively, promoting transparency and collaboration.
- **Continuous Improvement:** The Scrum Master encourages a culture of continuous improvement within the team. They facilitate retrospectives to identify areas of improvement and support the implementation of changes.
- **Protecting the Team:** The Scrum Master shields the team from external distractions and interruptions, allowing them to focus on sprint goals and commitments.
- **Facilitating Cross-Team Collaboration:** When necessary, the Scrum Master facilitates collaboration between multiple teams or stakeholders to address dependencies and ensure smooth project flow.