

**Department of Computer Engineering**

**Academic Term: First Term 2023-24**

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

<b>Practical No:</b>	<b>2</b>
<b>Title:</b>	<b>Implementing Project Using Scrum Method on JIRA Tool in Software Engineering</b>
<b>Date of Performance:</b>	09/08/2023
<b>Roll No:</b>	9622
<b>Team Members:</b>	Soham Mane, Sanket Mane, Nishant Patil, Joshua Lewis

**Rubrics for Evaluation:**

<b>Sr. No</b>	<b>Performance Indicator</b>	<b>Excellent</b>	<b>Good</b>	<b>Below Average</b>	<b>Total Score</b>
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:**

**Department of Computer Engineering**

**Academic Term: First Term 2022-23**

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

## **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.

**Signature of the Teacher:**

**Department of Computer Engineering**

**Academic Term: First Term 2022-23**

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

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

**Signature of the Teacher:**

**Department of Computer Engineering**

**Academic Term: First Term 2022-23**

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

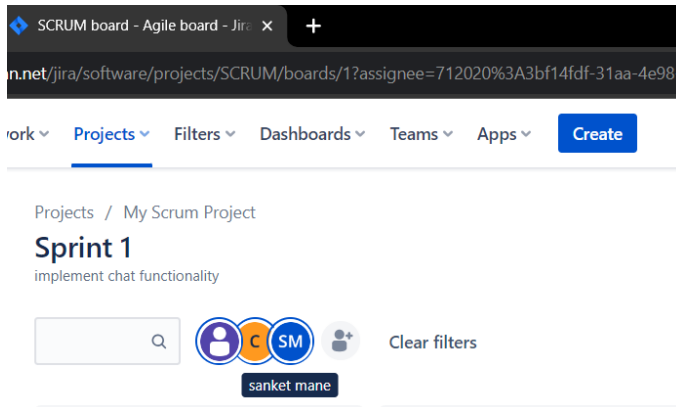
of agile methodologies, equipping them with valuable skills for real-world software development projects. The lab experiment encourages students to embrace the agile mindset, promoting continuous improvement and customer-centric software development practices.

**Signature of the Teacher:**

## Lab Experiment 02

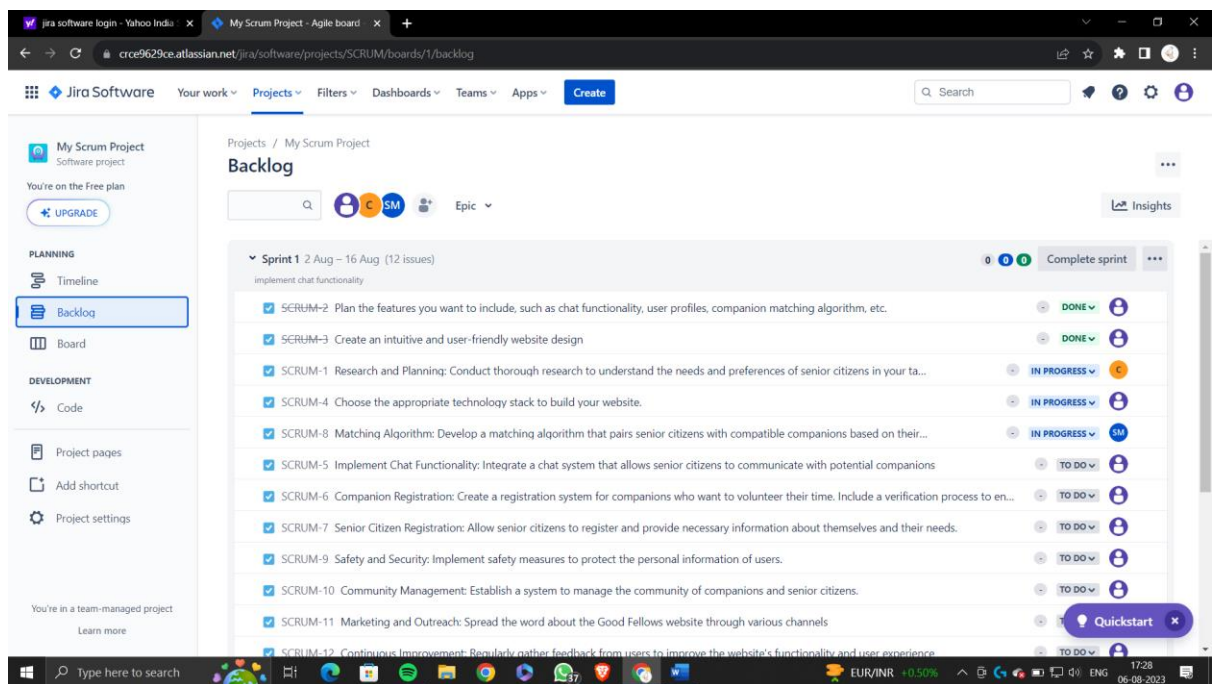
Experiment Name: Implementing Project Using Scrum Method on JIRA Tool in Software Engineering.

1) Users in the project:

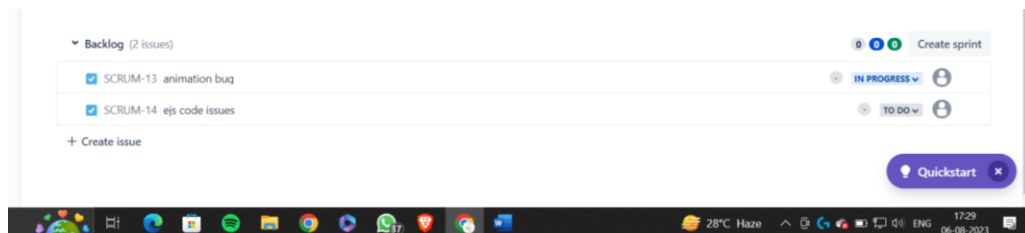


2) Issues:

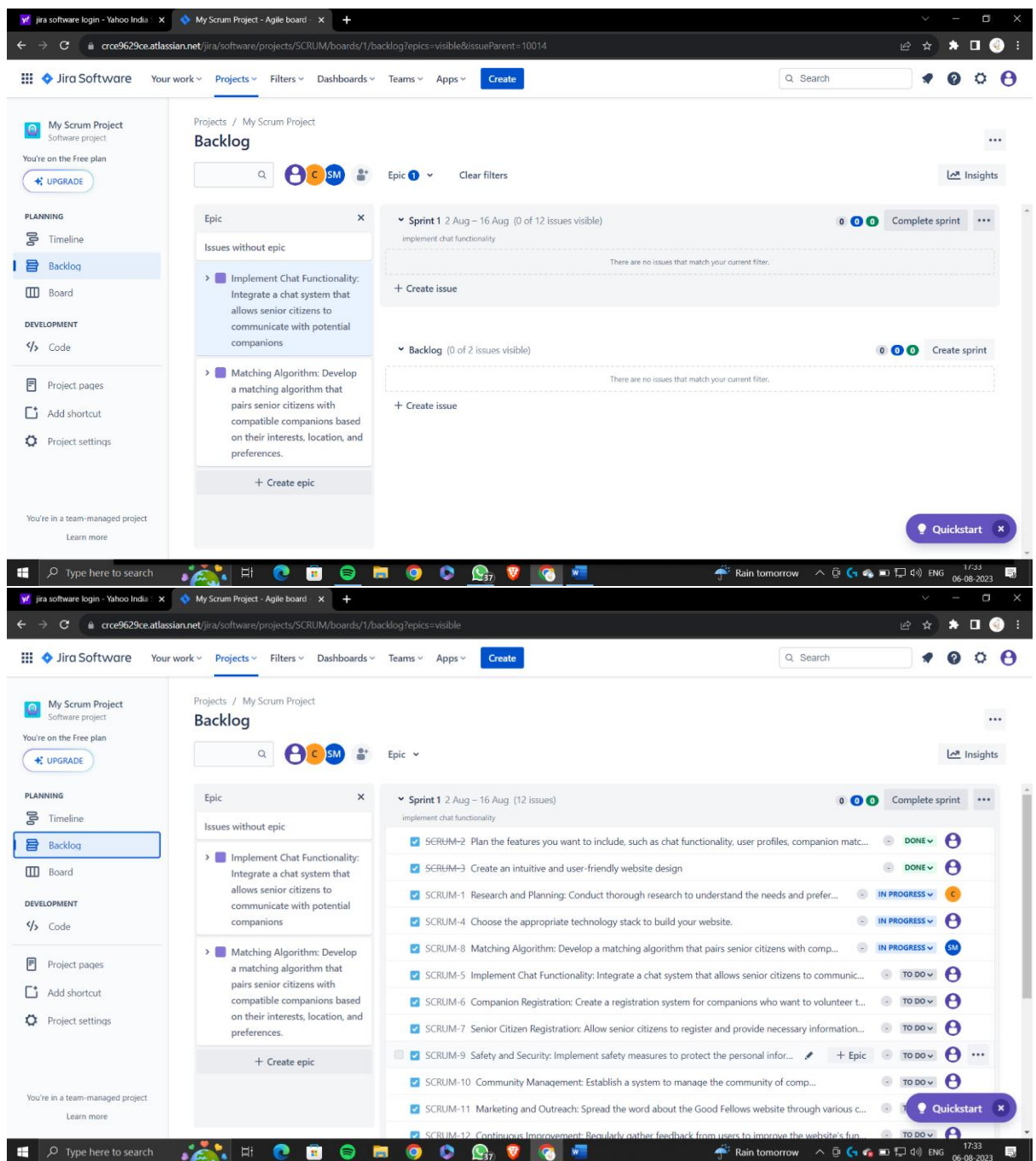
Methodology of the project.



3) Backlog:

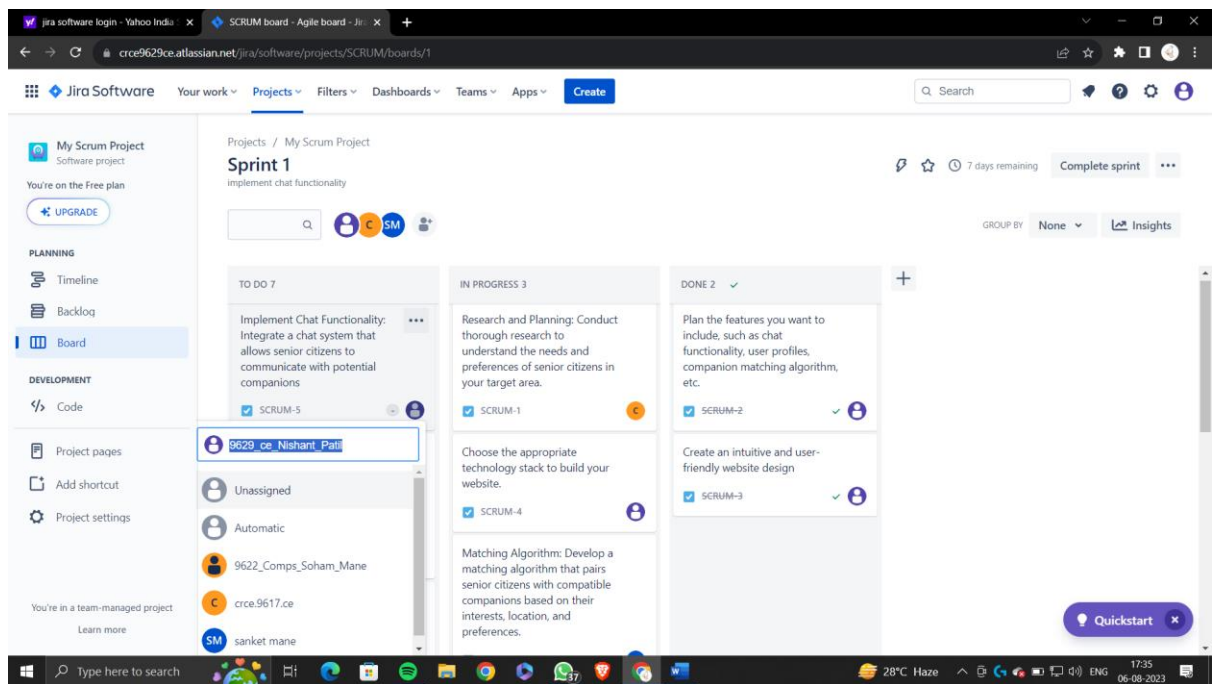


4) Epic:

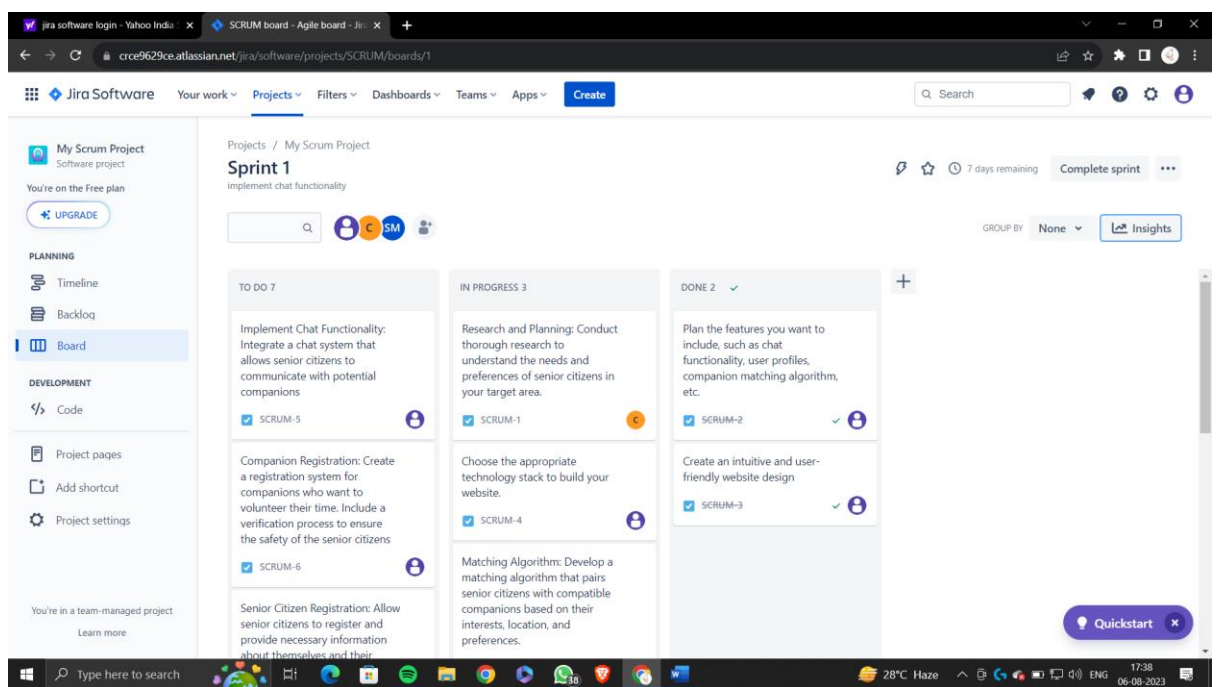


## 5) Board:

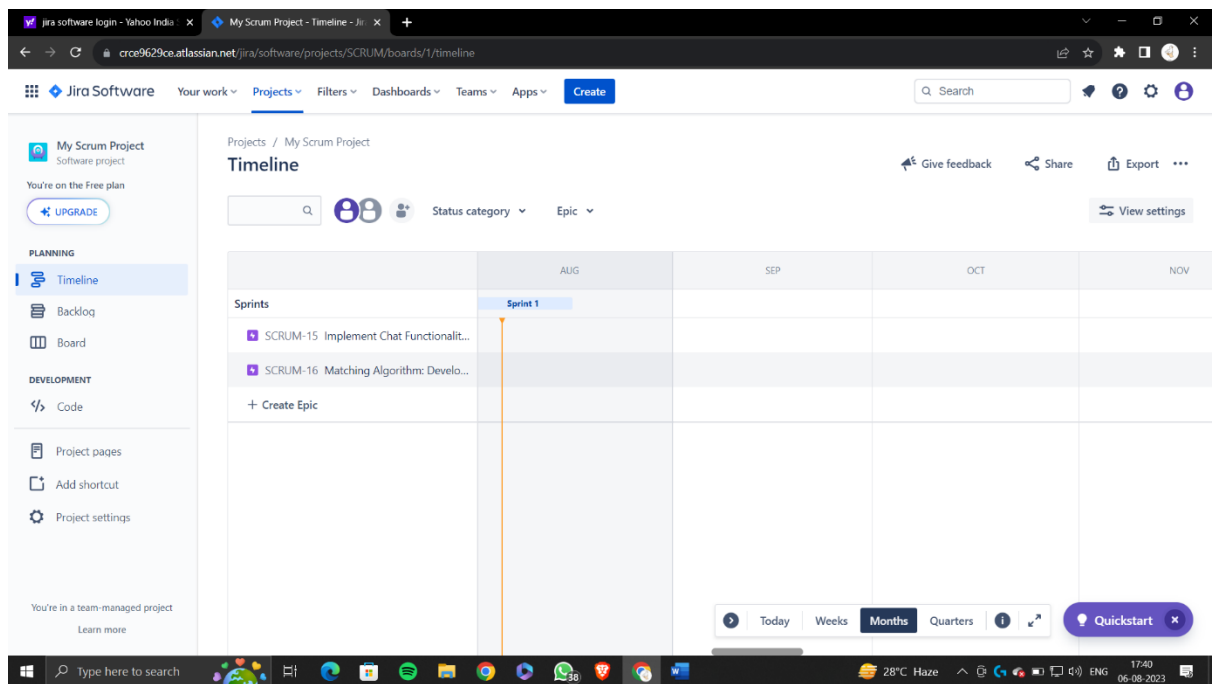
Work being assigned to the team



## 6) Board Status Page (To do, Progress, Done):



## 7) Timeline



## POST LAB EXP 2

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

Scrum has been widely adopted and utilized in the software development industry as an Agile framework. It offers an iterative and incremental approach to managing projects, focusing on flexibility and customer collaboration. To assess the effectiveness of the Scrum framework compared to traditional project management methodologies, let's consider some key aspects:

**Flexibility and Adaptability:** Scrum is highly flexible and allows teams to adapt to changing requirements and priorities. It empowers teams to adjust their course of action during the project's development, making it well-suited for dynamic environments where requirements can evolve rapidly. In contrast, traditional project management methodologies, like the Waterfall model, are more rigid and less adaptable to changes once the project plan is set.

**Incremental Delivery:** Scrum emphasizes delivering software in small, incremental releases called "sprints," typically lasting 1-4 weeks. This allows stakeholders to see progress regularly and provides early opportunities for feedback and course correction. Traditional methodologies often involve delivering the entire project at the end, which can lead to longer development cycles without feedback until the final delivery.

**Customer Collaboration:** Scrum encourages continuous collaboration with customers and stakeholders throughout the project. This engagement helps ensure that the product being developed aligns with customer needs and expectations. Traditional methodologies may not



emphasize customer collaboration as frequently, potentially leading to misalignment between the final product and customer requirements.

**Team Empowerment:** Scrum empowers development teams by allowing them to self-organize and make decisions on how to accomplish their goals. This autonomy fosters a sense of ownership and responsibility, leading to higher team morale and productivity. Traditional methodologies may be more hierarchical, with decisions made by managers or project leaders.

**Risk Management:** Scrum's iterative nature allows teams to identify and address risks early in the development process. By regularly reviewing progress, issues can be detected and resolved promptly. Traditional methodologies might not offer such early risk mitigation opportunities, leading to potential project delays or failure.

**Communication and Transparency:** Scrum emphasizes transparent communication among team members and stakeholders through various ceremonies like Daily Standups, Sprint Reviews, and Sprint Retrospectives. This enhances visibility into the project's progress and challenges. Traditional methodologies may not have such built-in communication structures, leading to potential information gaps.

**Size and Complexity of Projects:** Scrum is well-suited for smaller to medium-sized projects or projects with evolving requirements. For larger, more complex projects, traditional methodologies might offer better predictability and control.

**Q2) Analyse 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 too many user stories or tasks, the team may struggle to complete them all within the sprint time frame. This can lead to unfinished work and decreased productivity.

1. **Unclear or ambiguous user stories:** If user stories in the sprint backlog lack clear acceptance criteria or have ambiguous requirements, the team may face difficulty in understanding what is expected, leading to delays and rework.
2. **External dependencies:** If the sprint backlog includes tasks that are dependent on external teams, services, or resources, any delays from these dependencies can hinder the team's progress.
3. **Scope creep:** If new tasks or requirements are added to the sprint backlog during the sprint, it can disrupt the team's focus and may lead to unfinished work on previously committed tasks.
4. **Lack of skills or expertise:** If the team lacks the necessary skills or expertise to complete certain tasks in the sprint backlog, it can result in delays and may require additional training or collaboration with other team members.

5. Bottlenecks in workflow: If there are bottlenecks in the development process, such as code reviews or testing, it can slow down the progress of tasks and impede the completion of the sprint backlog.
6. Inadequate communication: If there is a lack of clear communication within the team or with stakeholders, misunderstandings can occur, leading to wasted efforts and rework.

To overcome these potential bottlenecks or issues:

- Conduct sprint planning meetings effectively to ensure a manageable sprint backlog.
- Ensure user stories are well-defined with clear acceptance criteria.
- Identify and resolve external dependencies early or plan for alternatives.
- Avoid scope creep by sticking to the agreed-upon sprint scope.
- Encourage continuous skill development and cross-training within the team.
- Monitor the workflow closely and address any bottlenecks promptly.
- Foster open and transparent communication within the team and with stakeholders.

### **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 critical role in handling conflicts within the development team and resolving impediments to maintain a smooth project flow in the Scrum framework. Their primary focus is on facilitating the Scrum process and supporting the team to achieve their goals. Here's how the Scrum Master fulfills these responsibilities:

1. Facilitating Communication and Collaboration: The Scrum Master ensures open and transparent communication within the team. They create an environment where team members feel comfortable expressing their concerns and ideas. By fostering collaboration, the Scrum Master helps prevent conflicts from escalating and encourages the team to work together effectively.
2. Conflict Resolution: When conflicts arise within the development team, the Scrum Master acts as a mediator and facilitator. They actively listen to all parties involved and help them understand each other's perspectives. By addressing conflicts early and constructively, the Scrum Master prevents them from becoming major roadblocks to progress.
3. Removing Impediments: The Scrum Master identifies and resolves impediments that hinder the team's progress. These impediments can be related to technical issues, organizational challenges, or process inefficiencies. The Scrum Master works closely with the team and stakeholders to eliminate or mitigate these impediments, ensuring a smooth project flow.
4. Coaching and Mentoring: The Scrum Master serves as a coach and mentor for the development team. They help team members enhance their skills, adopt Agile practices, and understand the Scrum framework. By providing guidance and support, the Scrum Master enables the team to self-organize and self-manage effectively.
5. Protecting the Team: The Scrum Master shields the development team from external interference and distractions. They ensure that the team can focus on their work during the sprint without undue pressure from stakeholders or external parties.
6. Facilitating Scrum Events: The Scrum Master facilitates various Scrum events, such as Daily Standups, Sprint Planning, Sprint Reviews, and Sprint Retrospectives. These events provide opportunities to address concerns, assess progress, and improve the team's performance.
7. Continuous Improvement: The Scrum Master encourages a culture of continuous improvement within the team. They help the team reflect on their performance and identify

areas for enhancement. By promoting a learning mindset, the Scrum Master assists the team in delivering better results in each subsequent sprint.

8. **Servant Leadership:** The Scrum Master embodies the principle of servant leadership, focusing on the needs of the team above their own. They support the team's development and well-being, ensuring that the team has what it needs to succeed.