

## EXPERIMENT 6

**AIM:** To Study Agile Methodology and Test case Management using JIRA Tool.

### **THEORY:**

Agile is a methodology focused on delivering small, incremental improvements to a product through short, iterative cycles called sprints. The idea is to break the project into smaller, manageable pieces that can be developed and delivered quickly, allowing the team to gather frequent feedback and make necessary adjustments.

#### **The key benefits of Agile include:**

- Flexibility to adapt to changing requirements.
- Collaboration between cross-functional teams.
- Faster delivery of product increments to get early feedback from customers.
- Improved risk management since any issues can be detected early in the iterative process.

### **Agile Manifesto and Principles**

The Agile Manifesto lays out the philosophy behind Agile development, but it is also supported by 12 principles that further guide how to operate within the Agile framework:

1. Customer satisfaction through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently, with a preference to shorter timescales.
4. Business people and developers must work together daily.
5. Build projects around motivated individuals, give them the environment and support they need, and trust them to get the job done.
6. Face-to-face communication is the most effective way to convey information.
7. Working software is the primary measure of progress.

8. Sustainable development, where the pace of development can be maintained indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. Self-organizing teams produce the best architectures, requirements, and designs.
12. Regular reflection on how to become more effective, then tuning and adjusting behaviour accordingly.

## **Popular Agile Frameworks:**

### **Scrum**

Scrum is one of the most popular Agile frameworks, focusing on fixed-length iterations (called sprints) that typically last between 2–4 weeks. Scrum has defined roles and ceremonies that help teams organize and manage their work:

### **Roles:**

- **Product Owner:** Responsible for defining the product backlog (a prioritized list of features and requirements).
- **Scrum Master:** Facilitates the Scrum process, removes obstacles, and ensures the team adheres to Scrum practices.
- **Development Team:** A cross-functional group of individuals who work together to deliver product increments.

### **Ceremonies:**

- **Sprint Planning:** Define goals and break them down into tasks for the sprint.
- **Daily Stand-up (Daily Scrum):** A quick daily meeting to share progress and discuss any blockers.
- **Sprint Review:** A meeting held at the end of the sprint to demonstrate the work completed and gather feedback.
- **Sprint Retrospective:** A reflection meeting where the team discusses what went well, what could be improved, and actionable improvements for the next sprint.

## **Kanban:**

Kanban is a continuous flow-based methodology with an emphasis on visualizing work using boards to track the progress of tasks. It doesn't use fixed iterations like Scrum but instead focuses on limiting the number of tasks in progress to improve flow and efficiency.

## **Extreme Programming (XP)**

XP is designed to improve software quality and responsiveness to changing customer requirements. It includes practices such as:

- **Pair programming:** Two developers work on the same code, which increases code quality and spreads knowledge.
- **Test-driven development (TDD):** Writing tests before the code to ensure that the software meets its requirements.
- **Continuous integration:** Frequent code integration to ensure that the software is always in a working state.

## **JIRA:**

JIRA, developed by Atlassian, is widely used for issue tracking, project management, and as a tool for managing Agile processes. However, JIRA is highly customizable, and when combined with plugins such as Zephyr and Xray, it becomes a powerful tool for managing test cases and test execution.

JIRA allows teams to:

- Track bugs and issues.
- Organize tasks and sprints.
- Document test cases and log test results.
- Report on test execution and project progress.

## **Key JIRA Features for Test Case Management**

### ● **Issue Types:**

- **Test Case Issue:** In JIRA, a Test Case is typically modeled as an issue, often with the issue type set to "Test". This allows the creation of detailed descriptions of the test case, such as steps, expected outcomes, and priority.
- **Defects (Bugs):** These are logged if a test case fails, helping track problems related to functionality.

- **Boards:**

- **Scrum and Kanban Boards:** JIRA boards (whether Scrum or Kanban) help visualize the status of test cases, tasks, and bugs. You can customize these boards to track the progress of specific issues (such as test cases) through various stages, e.g., "To Do", "In Progress", "In Testing", "Done".

### **Filters and Dashboards:**

JIRA allows the creation of custom filters to find specific test cases or bugs. For example, you could filter by:

- Test cases linked to a specific Epic or User Story.
- Test cases with a status of "Failed".
- Test cases assigned to a specific tester or developer.

Dashboards allow you to create custom views where you can track important metrics like:

- Total number of test cases passed/failed.
- Number of defects reported per sprint.
- Burndown charts tracking test progress during the sprint.

### **Managing Test Cases in JIRA:**

#### **Creating Test Cases:**

In JIRA, test cases are often created as individual issues. With Zephyr or Xray plugins, you can add more sophisticated attributes to each test case:

- **Test Steps:** Each test case can include individual steps, expected outcomes, and actual outcomes.
- **Test Data:** Details on the data required for testing.
- **Priority:** Set the priority of the test case based on business importance.

#### **Executing Test Cases:**

- Once test cases are created, they are executed in Test Execution sessions.
- In Zephyr, testers can run the tests directly from the JIRA interface, and results are logged automatically.
- After execution, the results can be marked as:
  - **Pass** (if the test works as expected),

- **Fail** (if the test does not meet expectations),
- **Blocked** (if there's an issue preventing the test from running, such as a dependency on another task or defect).

### **Defects Reporting:**

- When a test case fails, JIRA allows testers to create a Defect issue, which is typically linked to the failing test case. This allows for traceability between test results and issues that need resolution.
- Defects can be assigned to developers, prioritized, and tracked through their lifecycle, ensuring that critical bugs are fixed in a timely manner.

### **Test Case Execution Reporting:**

JIRA generates various reports (e.g., Test Execution Report, Test Summary Report) to show how many test cases were executed, how many passed, and how many failed. This can be used for Sprint Review and to measure quality over time.

### **Best Practices for Test Case Management in JIRA:**

- **Use Custom Workflows:** Customize workflows to meet the needs of your testing process. For example, you can create a "Test Execution" workflow where each step (e.g., Test Passed, Test Failed, Defect Raised) has specific transitions and actions.
- **Version Control:** Ensure that test cases are versioned, especially when managing test cases across different releases. This helps avoid confusion when a test case changes, as teams can reference the test case by version.
- **Automation Integration:** Integrating JIRA with test automation frameworks like Selenium or Cucumber can provide real-time test results and save time in manual execution.
- **Traceability:** Ensure that each test case is linked to specific User Stories, Epics, or Requirements so that stakeholders can easily trace the test cases back to the original business requirements.

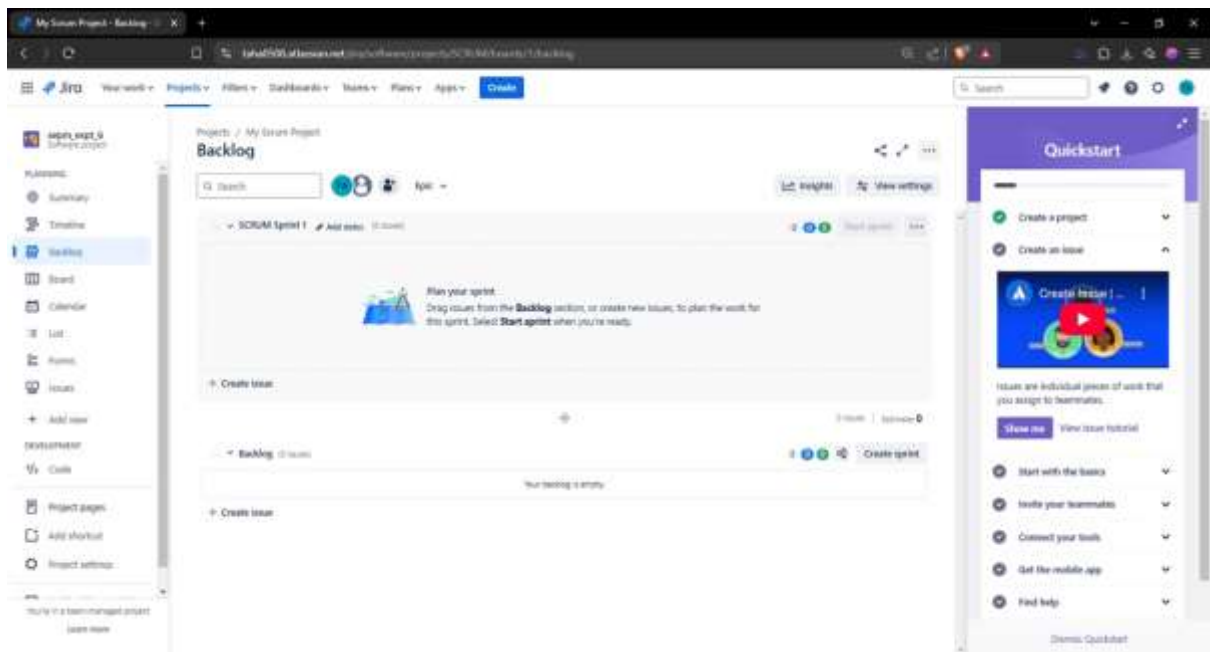
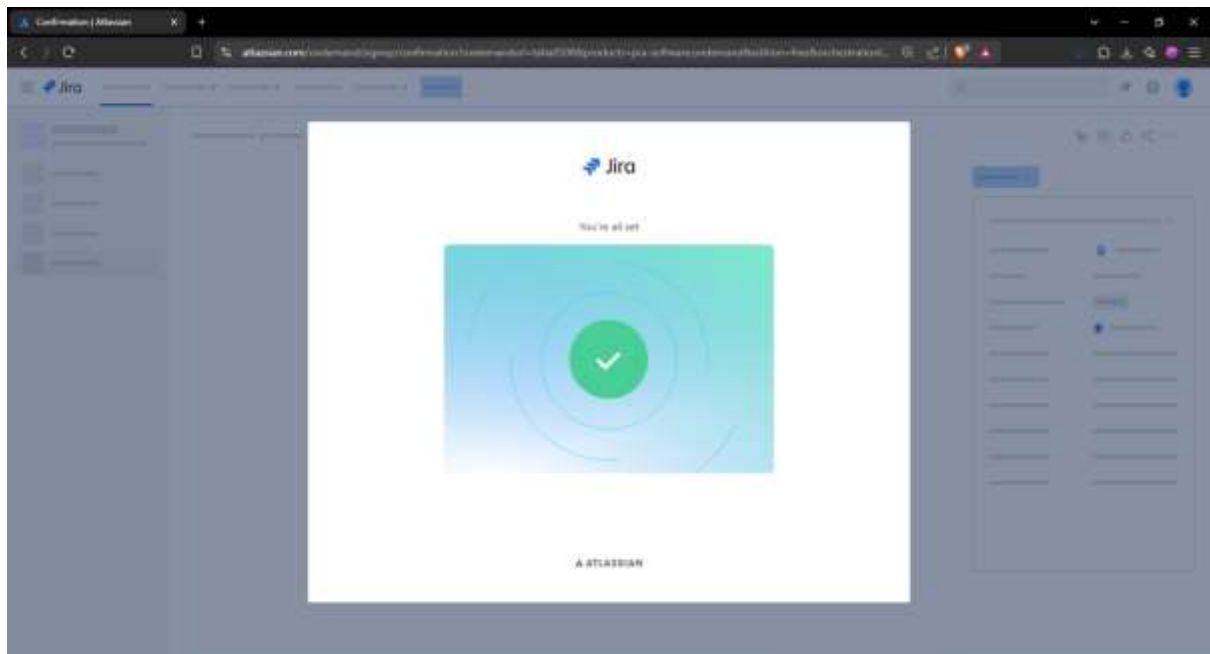
## Reporting and Metrics:

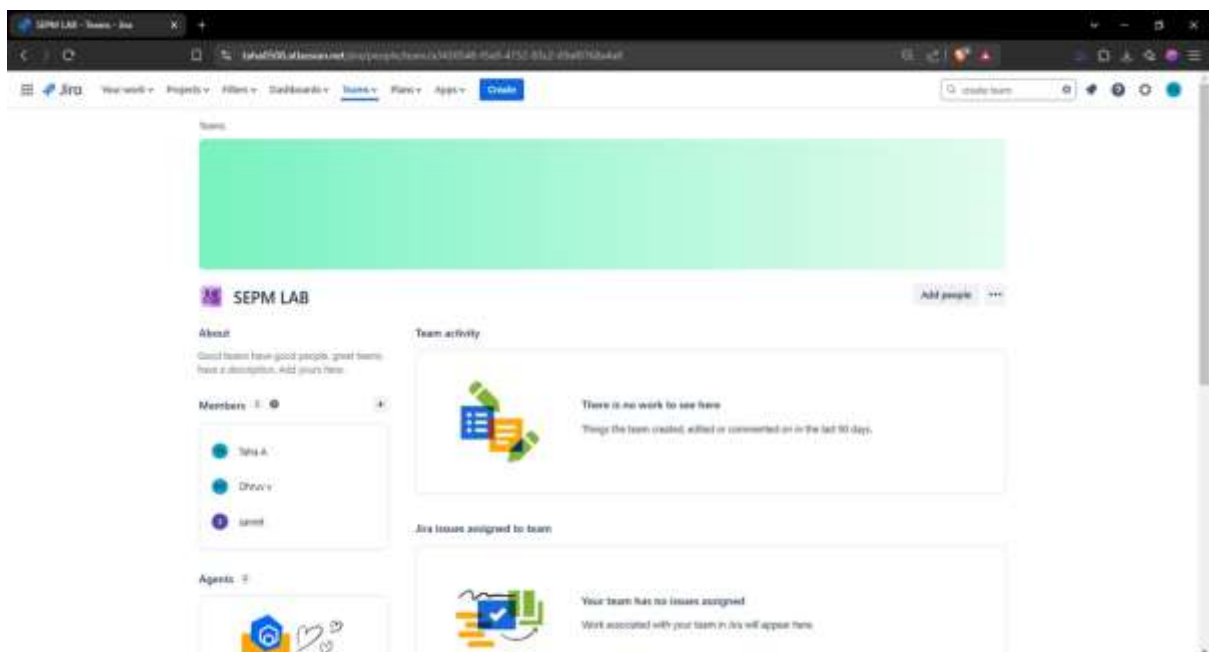
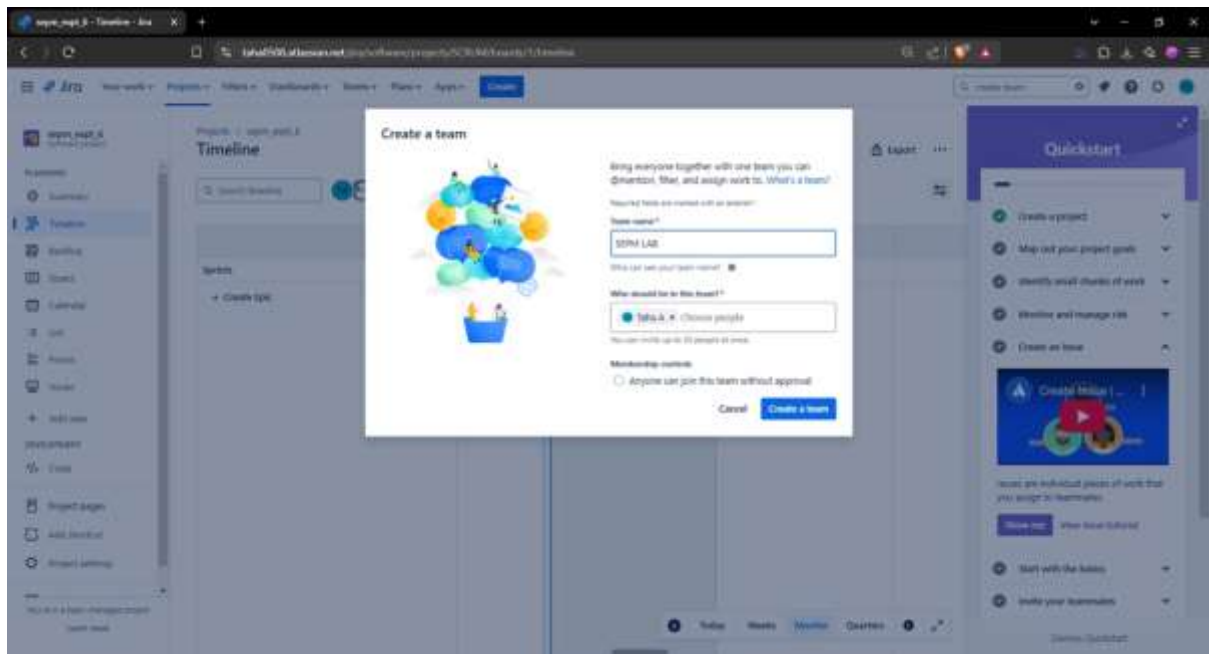
JIRA provides several out-of-the-box and customizable reports for test case management.

- **Burndown Charts:** Track the progress of test case execution across sprints.
- **Test Execution Reports:** Provides a summary of how many tests have passed, failed, or are blocked.
- **Defect Reports:** Track defects linked to test failures, and see trends in defect resolution.

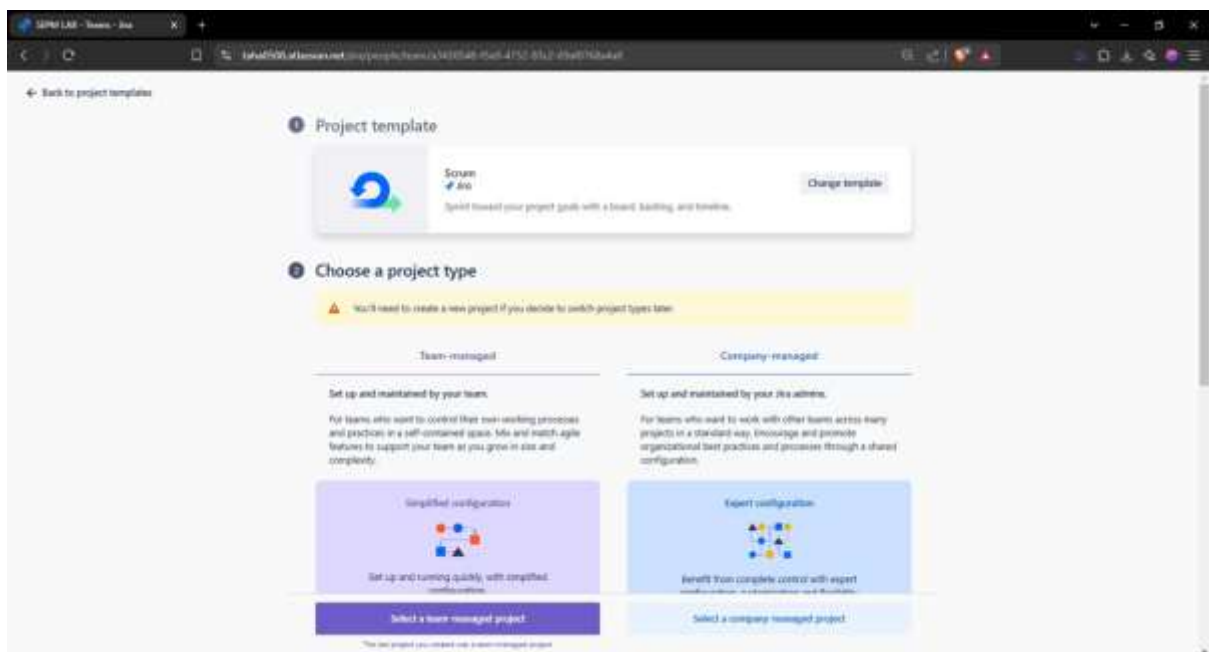
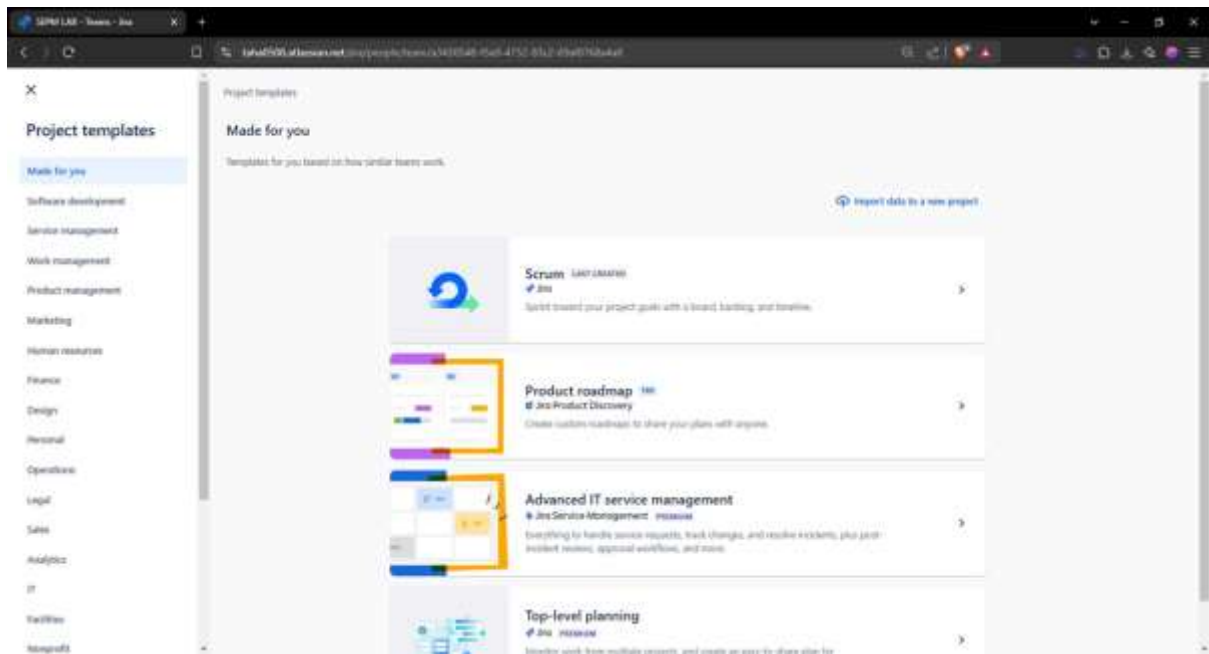
## DEMONSTRATION:

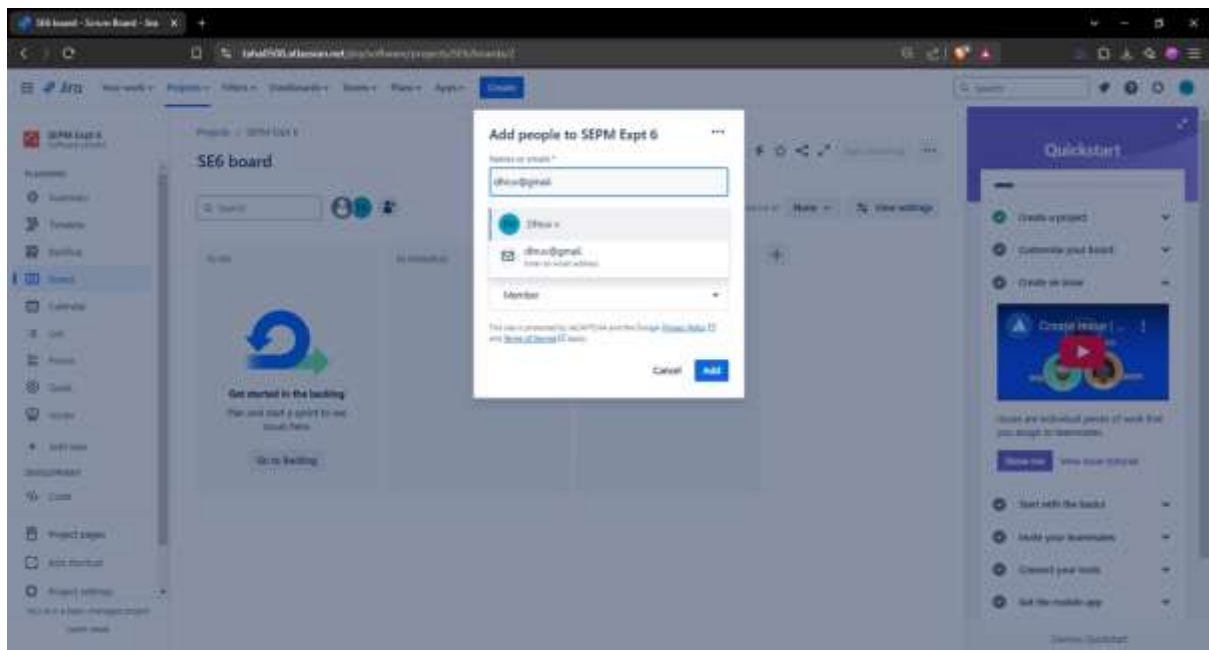
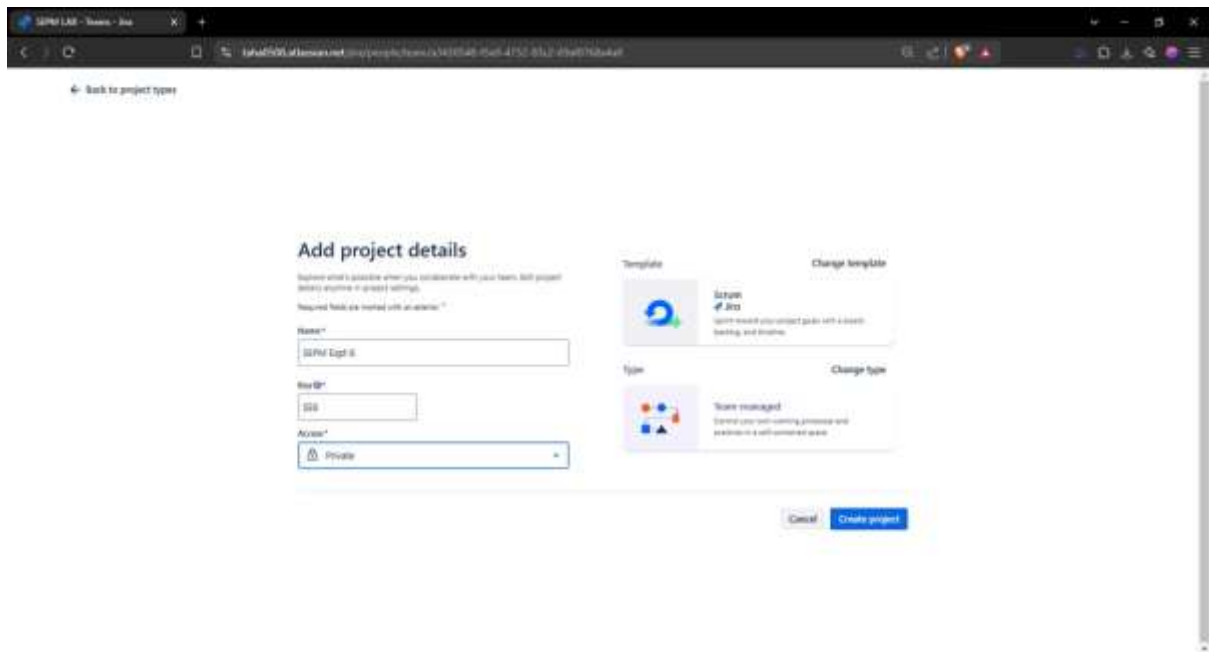


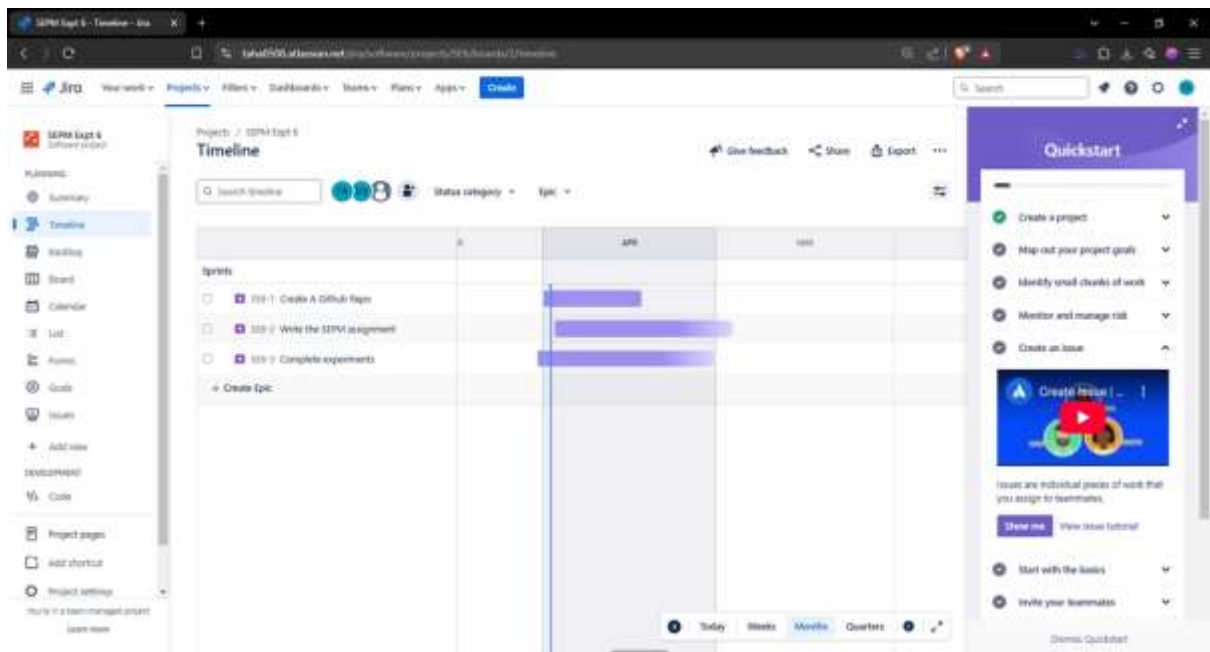
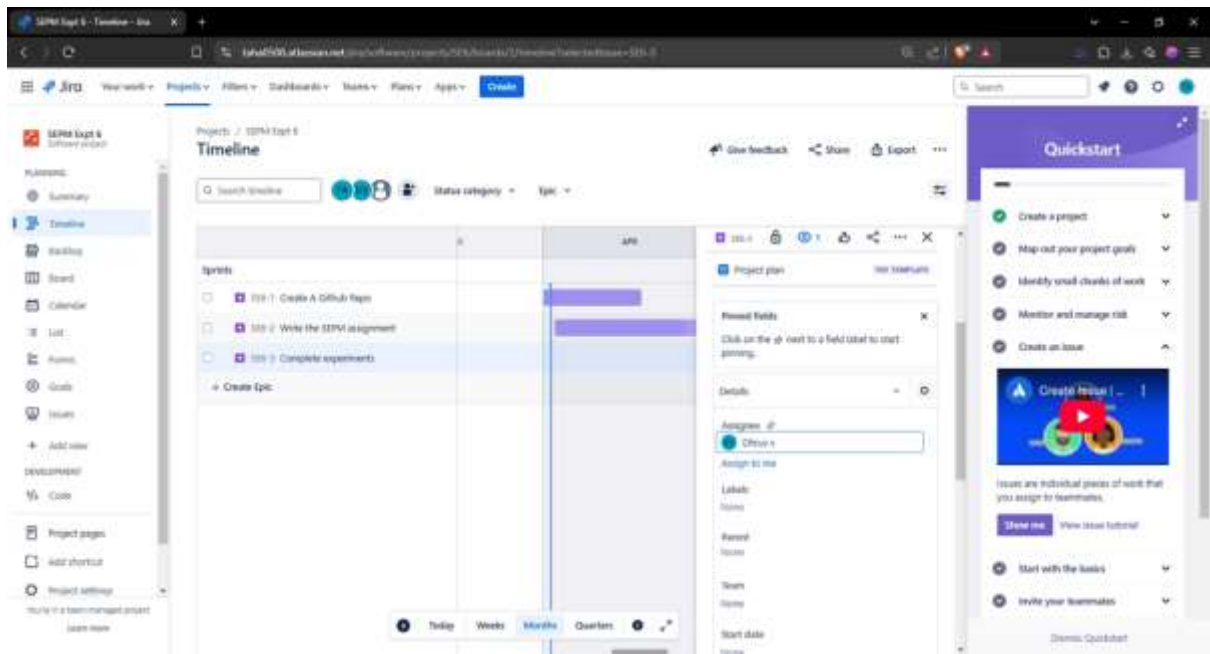


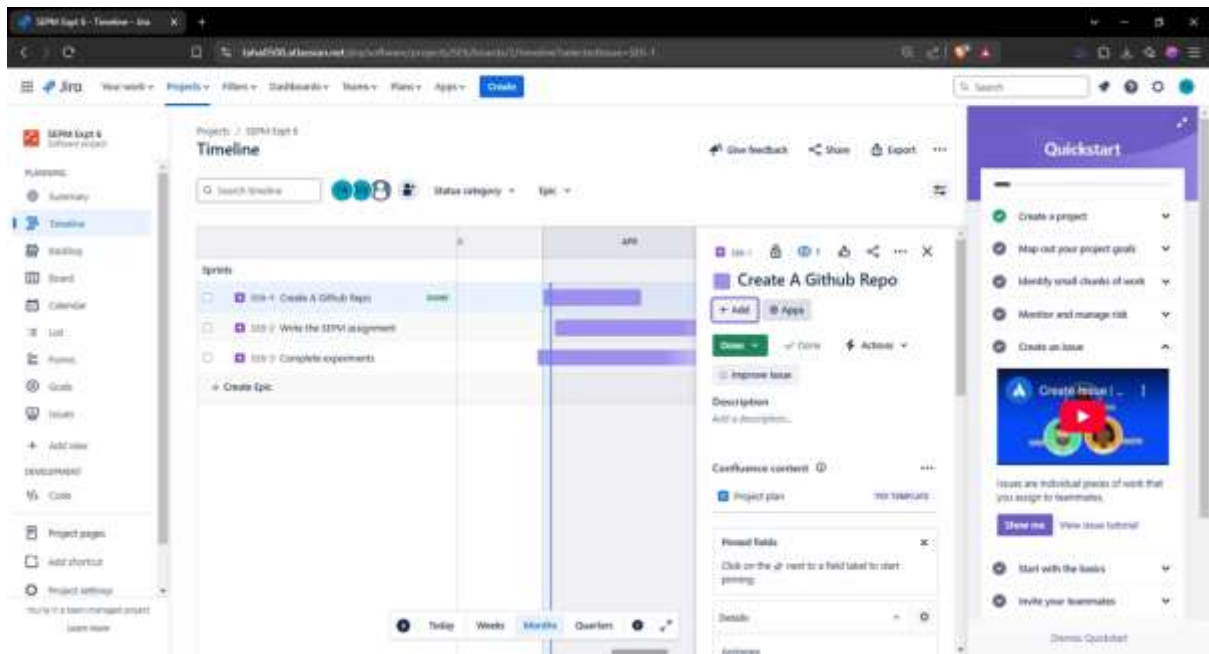












## CONCLUSION:

Hence, we have successfully implemented Agile Methodologies in JIRA.