

**Department of**  
**Computer**  
**Engineering Academic**  
**Term: First Term**  
**2023-24**

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

<b>Practical No:</b>	2
<b>Title:</b>	<b>Software Requirement Specification</b>
<b>Date of Performance:</b>	27/07/2023
<b>Roll No:</b>	9540
<b>Team Members:</b>	

**Rubrics for Evaluation:**

<b>Sr . N o</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(don't 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.

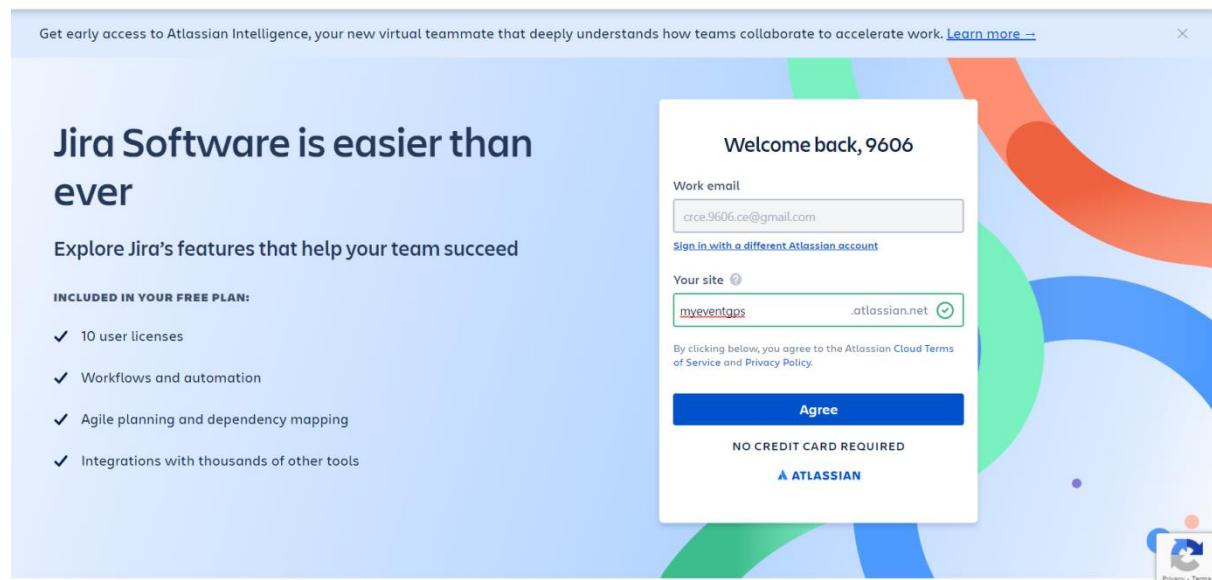
**Materials and Resources:**

- 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. Creating Project:

Jira Software



Get early access to Atlassian Intelligence, your new virtual teammate that deeply understands how teams collaborate to accelerate work. [Learn more →](#)

## Jira Software is easier than ever

Explore Jira's features that help your team succeed

**INCLUDED IN YOUR FREE PLAN:**

- ✓ 10 user licenses
- ✓ Workflows and automation
- ✓ Agile planning and dependency mapping
- ✓ Integrations with thousands of other tools

Welcome back, 9606

Work email  
crce.9606.ce@gmail.com

Sign in with a different Atlassian account

Your site [myeventgps](#) .atlassian.net

By clicking below, you agree to the Atlassian Cloud Terms of Service and Privacy Policy.

**Agree**

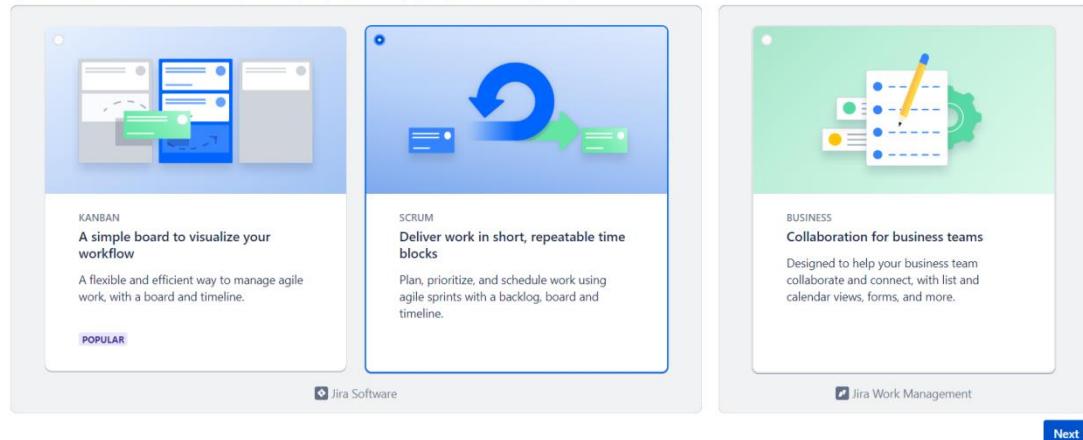
NO CREDIT CARD REQUIRED

ATLASSIAN

Privacy • Terms

### Select a template for your first project

If you're not sure what to choose, don't worry. You can quickly create a new project if this one's not right for you.



**KANBAN**  
A simple board to visualize your workflow  
A flexible and efficient way to manage agile work, with a board and timeline.

**POPULAR**

**SCRUM**  
Deliver work in short, repeatable time blocks  
Plan, prioritize, and schedule work using agile sprints with a backlog, board and timeline.

**BUSINESS**  
Collaboration for business teams  
Designed to help your business team collaborate and connect, with list and calendar views, forms, and more.

Jira Software

Jira Work Management

Next

## 2 Creating an epic

The screenshot shows the Jira interface for creating a new issue. At the top, there's a header with a lightning bolt icon and the text "event management system". To the right are minimize, maximize, and close buttons. Below the header, there are two dropdown menus: "Issue type \*" set to "Epic" and "Status" set to "TO DO". A "Learn more" link is also present. The "Summary \*" field contains the text "event management system". In the "Description" section, there's a rich text editor toolbar with options for font size, bold, italic, underline, and color. The main text area describes the system as an event management system for organizers and customers, detailing the process of publishing events, getting feedback, and booking tickets. At the bottom left is a checkbox for "Create another issue", and at the bottom right is a large blue "Create" button.

Issue type \*

Status

Epic

TO DO

Learn more

Summary \*

event management system

Description

Text color

Aa **B** *I* ... A +

Event management system for the organizer and the customers. Organizer publish his events ,get the feedback from the audience . Audience, customers select the type of events available and book the ticket ,pay the price and enjoy the event.

Create another issue

Create

### 3.Creating a story

 creating the frontend.

— ↗ ✖

Project \*

 myeventGPS (MYEV)

Issue type \*

 Story

Status

 TO DO

[Learn more](#)

Summary \*

creating the frontend.

Description

Aa<sup>v</sup> | B I ... | A<sup>v</sup> | ⚙<sup>v</sup> | +<sup>v</sup>

All the team members should start to learn the html, css, js basics.

Create another issue Create

#### 4. Assigning subtasks

#### creating the frontend.

Attach Add a child issue Link issue ...

##### Description

All the team members should start to learn the html, css , js basics.

##### Child issues

Order by ... +

0% Done

 MYEV-5	Shubham -understanding of the html ,...		 TO DO
 MYEV-6	Glison-Focus more on CSS		 TO DO
 MYEV-7	Pradyumnaa-Start with data base und...		 TO DO

What needs to be done?

Choose an existing issue

Create

Cancel

## 5 Sprint

### Start Sprint

2 issues will be included in this sprint.

Required fields are marked with an asterisk \*

Sprint name \*

MYEV Sprint 2

Duration \*

2 weeks



Start date \*

8/6/2023

8:19 PM



End date \*

8/20/2023

8:19 PM

Sprint goal

Start

Cancel



## Final:

Projects / myeventGPS

### MYEV Sprint 2

⚡ ⭐ ⏳ 10 days remaining

Complete sprint

...



GROUP BY

None



TO DO 2

ps sprint 4

MYEV-11

IN PROGRESS

ps sprint 8

MYEV-12

DONE ✓

Projects / myeventGPS

### MYEV Sprint 2

⚡ ⭐ ⏳ 10 days remaining

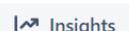
Complete sprint

...



GROUP BY

None



TO DO 1

ps sprint 8

MYEV-12

IN PROGRESS 1

ps sprint 4

MYEV-11

DONE ✓

### MYEV Sprint 2

⚡ ⭐ ⏳ 10 days remaining

Complete sprint

...



GROUP BY

None



TO DO 1

ps sprint 8

MYEV-12

IN PROGRESS

DONE 1 ✓

ps sprint 4

MYEV-11

Edit Summary



Postlab:

- a) Assess the effectiveness of the Scrum framework for managing software development project compared to traditional project management methodologies?

Sol:- The main difference between scrum and traditional project management methodologies can be summed up as fixed scope vs. iterative decision making. **Classic project management** calls for project managers to look at the development as a whole whereas Scrum has no problem dividing it up into segments. Unlike **classic project management** methods, Scrum focuses more on personal responsibility.

Scrum Framework management	Traditional Project Management
Emphasizes flexibility and adaptability	Emphasizes planning and predictability
Prioritizes customer satisfaction and working software	Prioritizes following a strict plan and meeting project requirements
Linear	Iterative
Large-scale	Small and Medium scale
Life cycle model	Evolutionary delivery model
Self-organizing teams	Hierarchically organized teams
Cross-functional teams	Functionally divided teams

- b) Analyse a Sprint Backlog in JIRA and identify any potential bottlenecks or issues that might hinder the team's progress during the sprint?

Sol:- The **backlog** of a Scrum board shows the issues for your project grouped into a backlog and sprints. In the Scrum backlog, you can create and update issues, drag and drop issues to rank them, or assign them to sprints, epics, or versions, manage epics, and more. You'd typically use the Scrum backlog when building a backlog of issues, planning a new version, and planning a sprint. An issue will only be visible in the Scrum backlog if:

1. the issue *isn't* a subtask,
2. the issue matches the board's saved filter,
3. the issue's status maps to one of the board's columns (but not the *Done* column), and
4. there's at least a status being mapped to the *right-most* column.

For example, if you have the columns *To Do*, *In Progress*, and *Done*, ensure that you have a status mapped to *In*

*Progress* at least. If you map all the statuses to the first (To Do) column, you won't see any issues in the Scrum backlog.

Some common issues that might hinder the team's progress during the sprint:

1. Overloaded Sprint Backlog: If the Sprint Backlog contains too many user stories or tasks, it can lead to overcommitment and make it difficult for the team to complete everything within the sprint timeline. This can lead to a decrease in overall productivity and quality.
2. Unclear or Unrefined User Stories: User stories that lack clarity or are not welldefined can cause confusion and lead to delays in development. The team may spend extra time seeking clarification or reworking the implementation.
3. Lack of Task Breakdown: If user stories are not broken down into smaller, manageable tasks, it can be challenging for team members to collaborate effectively, estimate efforts accurately, and track progress efficiently.
4. Dependencies and Blocked Items: Dependencies between user stories or tasks can create bottlenecks if the completion of one task is contingent on another. Similarly, blocked items (tasks that cannot progress due to external factors) can slow down the team's overall progress.
5. Insufficient Test Coverage: Lack of adequate testing or testing delays can lead to the accumulation of bugs, reducing the team's capacity to work on new items.
6. Communication and Collaboration Issues: Inadequate communication and collaboration between team members can lead to misunderstandings, duplication of efforts, and decreased productivity

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

Sol:- A Scrum Master (Scrum Master), popularly known as the “servant leader” is a coach, motivator and leader of an Agile team. The role of a Scrum Master is to educate the team on Agile processes and help team members follow Scrum practices religiously. A good Scrum Master helps to establish a high-performing team dynamic, a continuous flow, and an exponential improvement in processes. They are required to play a pivotal role and are responsible for the progressive development of a Scrum team. The Scrum Master collaborates both with the Product Owner (PO) who focuses on building the right product, and the development team that focuses on building the product right. A Scrum Master’s job is essentially to help everyone understand and imbibe Scrum values, principles, and practices and get the best product out to the customer. To facilitate conflict management in Scrum teams, Scrum masters can create and maintain a safe and positive team environment, model and promote behaviors

and attitudes that foster constructive conflict, educate and empower the team to prevent, minimize, and resolve conflict, mediate and facilitate the conflict resolution process when needed, provide feedback and guidance to the team and the individuals on how to improve their conflict management skills, and encourage and support the team to leverage conflict for improvement.

1. Conflict Resolution: Facilitation: The Scrum Master acts as a neutral facilitator during team interactions and helps team members communicate effectively. They create a safe and open environment where conflicts can be addressed constructively.

Mediation: When conflicts arise between team members, the Scrum Master mediates the situation, encouraging active listening and helping to find mutually acceptable resolutions. Conflict Management: The Scrum Master helps the team understand that conflict is a natural part of collaboration and can lead to growth and improvement when managed properly. They guide the team towards constructive conflict resolution.

2. Impediment Removal:

Identifying Impediments: The Scrum Master actively seeks out impediments that are hindering the team's progress. These impediments can be related to tools, processes, team dynamics, or external factors.

Team Empowerment: The Scrum Master empowers the team to identify impediments and encourages them to take ownership of resolving issues that are within their control. Escalation: For impediments that are beyond the team's control, the Scrum Master escalates the issues to the appropriate stakeholders or management to ensure they are addressed promptly.

3. Team Collaboration: Collaboration Facilitation: The Scrum Master promotes collaboration and effective communication within the team, helping to build a strong sense of teamwork and trust.

Cross-Functional Cooperation: They encourage cross-functional cooperation and help team members understand each other's roles and contributions.

4. Removing Obstacles:

Organizational Support: The Scrum Master collaborates with stakeholders and management to address organizational impediments that affect the team's

productivity. Resource Allocation: They work to ensure that the team has the necessary resources, tools, and support to deliver the project successfully.