

Department of Computer Engineering

Academic Term: First Term 2023-24

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

| | |
|-----------------------------|---|
| Practical No: | 3 |
| Title: | Implementing Project Using Kanban Method on JIRA Tool in Software Engineering |
| Date of Performance: | 08/08/2023 |
| Roll No: | 9625 |
| Team Members: | Mohtashim Ali, Aditya Dhikale, Siddhant Murade (9644) (9531) (9625) |

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:

Department of Computer Engineering

Academic Term: First Term 2022-23

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

Signature of the Teacher:

Lab Experiment 03

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

Objective: The objective of this lab experiment is to introduce students to the Kanban method and

its implementation using the JIRA tool. Students will gain practical experience in managing a software project using Kanban principles and learn how to utilize JIRA as a project management tool

to visualize workflow, manage work items, and improve team productivity.

Introduction: Kanban is an agile project management method that emphasizes visualizing work,

limiting work in progress, and continuously improving the workflow. JIRA is a popular tool that

supports Kanban practices, allowing teams to manage their tasks and activities effectively.

Lab Experiment Overview:

1. Introduction to Kanban: The lab session begins with an overview of the Kanban method, including the principles of visualizing work, managing flow, and making incremental improvements.

2. JIRA Tool Introduction: Students are introduced to the JIRA tool and its features for implementing Kanban. They learn to create boards, swimlanes, columns, and customize workflows.

3. Defining the Project: Students are assigned a sample software project and create a Kanban board

in JIRA to visualize their workflow. They set up columns to represent different stages of their development process.

4. Creating Work Items: Students create work items (tasks, user stories, or issues) on the Kanban

board, representing the work that needs to be done.

5. Managing Workflow: Students move work items through the columns on the Kanban board as

they progress through their development process. They monitor work in progress limits to maintain an efficient workflow.

6. Continuous Improvement: Students conduct regular team meetings to discuss the workflow,

identify bottlenecks, and make improvements to enhance their efficiency.

7. Completion and Review: At the end of the lab experiment, students review their project progress

on the Kanban board. They discuss their experiences with implementing the Kanban method on

JIRA and share insights on its effectiveness.

8. Conclusion and Reflection: Students reflect on their experience with Kanban and JIRA, discussing the benefits and challenges they encountered during the project. They also consider

how Kanban principles can be applied to future software development projects.

Learning Outcomes: By the end of this lab experiment, students are expected to:

Understand the Kanban method and its application in agile project management.

Gain practical experience in using the JIRA tool to implement Kanban boards and workflows.

Learn to visualize work, manage flow, and limit work in progress using Kanban principles.

Develop team collaboration skills by continuously improving the workflow through regular team meetings.

Appreciate the importance of visualizing and managing work items for better project management.

Pre-Lab Preparations: Before the lab session, students should familiarize themselves with the

Kanban method and the basics of the JIRA tool. They should review Kanban principles, visualizing

workflows, and the features of JIRA relevant to Kanban 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 Kanban concepts

Conclusion: The lab experiment on implementing a project using the Kanban method on the JIRA

tool provides students with practical insights into agile project management. By applying Kanban

principles and utilizing JIRA's capabilities, students learn to visualize their work, manage flow efficiently, and continuously improve their development process. The hands-on experience with

Kanban and JIRA fosters teamwork, collaboration, and adaptability, enabling students to effectively

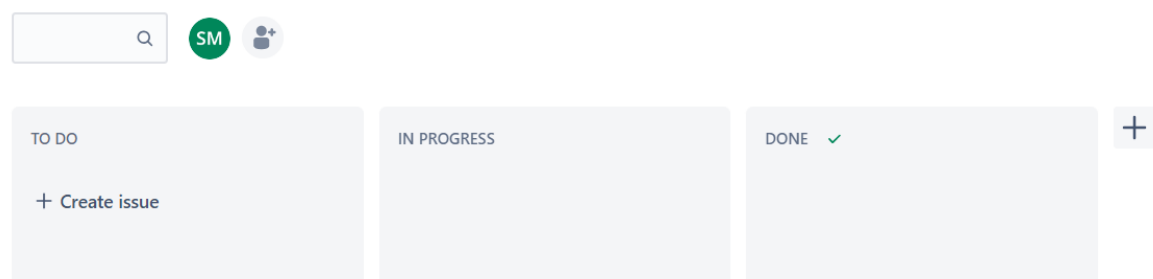
manage software projects with a focus on efficiency and quality. The lab experiment encourages

students to adopt Kanban's lean principles, promoting a culture of continuous improvement and

optimizing their workflow to deliver valuable software products.

Projects / Wildcart-Online shopping site

WOSS board



Creating Kanban Board

Creating Work Items

Projects / Wildcart-Online shopping site

WOSS board

SM

GROUP BYNone

Insights

TO DO 3

Product Catlogue

WOSS-1

User experience

WOSS-4

Payment Gateway

WOSS-7

Create issue

IN PROGRESS 3

UI design Refinement

WOSS-2

Backend Development for product database

WOSS-5

Shopping cart development

WOSS-8

DONE 3

user registration complited

WOSS-3

product listing integrated

WOSS-6

Marketing and Promotion

WOSS-9

Timeline

Projects / Wildcart-Online shopping site

Timeline

SM

Status category

Epic

| | SEP | OCT |
|------------------------------------|-----|-----|
| <div>WOSS-10 Payment Gateway</div> | | |
| <div>Create Epic</div> | | |

Q1 ANS: Kanban and Scrum are both agile methodologies, but they have distinct characteristics when it comes to flexibility, adaptability, and workflow management in various project scenarios.

Flexibility:

Kanban: Kanban is highly flexible and focuses on continuous improvement. It allows teams to adapt to changing priorities and requirements easily. New tasks can be added to the backlog and pulled into the workflow as capacity allows

Scrum: Scrum is more rigid in terms of its time-boxed iterations (sprints). Changes to the sprint backlog are discouraged during a sprint to maintain stability, but they can be considered for the next sprint,

making it less flexible within a sprint.

Adaptability:

Kanban: Kanban is designed for incremental, evolutionary change. It's suitable for projects with

constantly changing requirements or where adaptability is a top priority.

Scrum: Scrum encourages teams to inspect and adapt at the end of each sprint, allowing for regular adjustments. However, changes are typically planned for the next sprint, which may not be as responsive as Kanban's continuous adaptation

Workflow Management:

Kanban: Kanban visualizes the entire workflow, making it easier to identify bottlenecks and optimize processes. Work items flow through stages based on their priority, and there are no prescribed ceremonies or roles

Scrum: Scrum prescribes specific roles (Product Owner, Scrum Master, and Development Team) and ceremonies (Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective) to manage the workflow. It provides a more structured approach to work management.

Project Scenarios:

Kanban: Kanban is well-suited for support and maintenance projects, where tasks arrive unpredictably. or for teams that need to balance multiple concurrent projects. Scrum: Scrum works best for projects with well-defined goals and a stable backlog. It's effective for

delivering features or products in a series of iterations.

In summary, Kanban offers greater flexibility and adaptability, making it suitable for scenarios with changing priorities and ongoing optimization, Scrum provides a more structured approach with fixed sprint durations, making it better for projects with defined goals and less

frequent changes. The choice between them depends on the specific needs and nature of the project. Some teams even combine elements of both methodologies to create a hybrid approach that suits their unique circumstances

Q2 ANS: Analyzing a Kanban board in JIRA and proposing improvements to optimize efficiency and productivity involves a detailed examination of the current workflow. Here are some steps to help you

with this analysis and suggestions for improvements:

Review the Current Workflow:

Start by examining the existing Kanban board in JIRA. Identify the columns representing different stages of work (eg. To Do, In Progress, Testing, Done).

WIP (Work in Progress) Limits:

Ensure that Work in Progress limits are defined for each column. If not, establish and enforce these

limits. WIP limits prevent overloading team members and maintain a smoother flow of work.

Cycle Time Analysis:

Analyze the historical data on cycle times (the time it takes for a task to move from one column to another). Identify any bottlenecks or consistently delayed stages.

Classify and Prioritize Work Items:

Categorize work items based on priority, complexity, or type. Use swimlanes or labels to differentiate between different types of tasks. This can help in focusing on high-priority items.

Visualize Dependencies:

Use links, sub-tasks, or other visual cues to indicate dependencies between tasks. This helps the team

see and manage interrelated work items more effectively.

Regular Standup Meetings:

If not already in place, introduce daily standup meetings for the team to discuss progress, blockers, and

potential improvements. These meetings promote transparency and collaboration.

Continuous Improvement

Encourage the team to regularly review and reflect on their process. Hold retrospective meetings after a

certain number of completed tasks or at the end of each sprint to identify areas for improvement.

Automate Repetitive Tasks:

Identify any repetitive or manual tasks that can be automated using IRA plugins or integrations. Thi