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

Practical No:	3
Title:	Implementing Project using KANBAN method on JIRA Tool
Date of Performance:	10/8/2023
Roll No:	9641
Team Members:	Prashant Singh(9641), Mohtashim Ali(9644), Mark Tuscano(9645), Sania Tuscano(9646)

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:

Prashant Singh 9641

TE comps-B

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.

Creating Kanban Board

Creating Work items

Work in Progress Limits

Column limit

We'll highlight this column if the number of issues in it passes this limit.
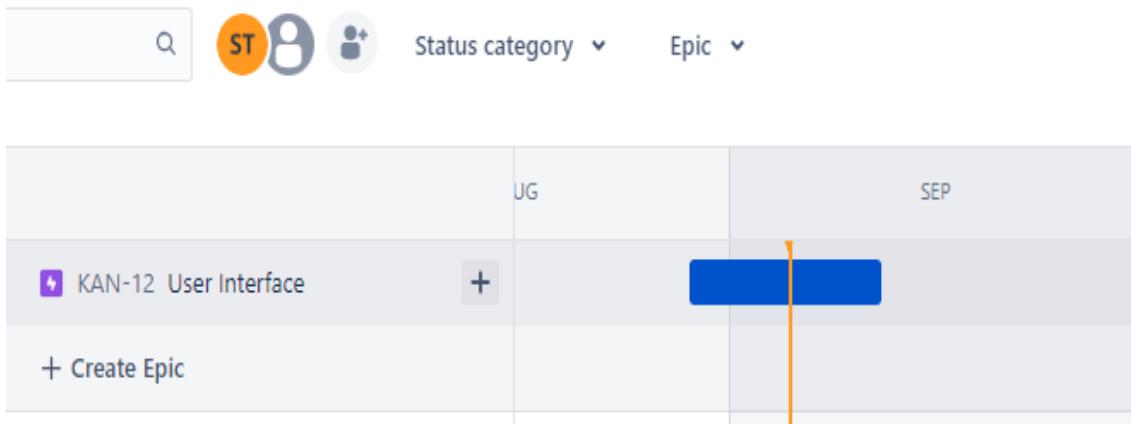
Maximum issues

| 3 | Clear limit |

Save   Cancel

Timeline

Timeline

| 🔍 | **ST** 👤 👤⁺ | Status category ∨ | Epic ∨ |

	UG	SEP
⚡ KAN-12 User Interface ➕		
➕ Create Epic		

POSTLABS

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 (e.g., 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 JIRA plugins or integrations. This can save time and reduce the risk of human error.

Clear Definition of Done:

Ensure that the "Definition of Done" for each task or user story is well-defined and agreed upon by the team. This clarity prevents misunderstandings and ensures quality work.

Metrics and Reporting:

Utilize JIRA's reporting features to track key performance metrics like lead time, cycle time, and throughput. These metrics can provide insights into the team's efficiency and areas needing improvement.

Training and Skill Development:

Identify skill gaps within the team and provide training or resources to address them. Skilled team members are more efficient and productive.

Regularly Update Board Configuration:

As the team evolves and its needs change, be open to adjusting the board configuration, such as adding or removing columns, swimlanes, or custom fields.

Feedback Loop:

Establish a feedback loop with team members to gather their input on the Kanban board's usability and effectiveness. Act on their suggestions for improvements.

Remember that continuous improvement is a fundamental principle of Kanban. The key is to regularly assess the process, make incremental changes, and adapt to the evolving needs of the team and the project. Your specific improvements will depend on the unique challenges and goals of your team.

Q3 ANS: Work In Progress (WIP) limits in a Kanban board play a crucial role in managing workflow and have a significant impact on a team's throughput and cycle time. Here's an evaluation of their impact:

1. Balancing Workload:

 Positive Impact: WIP limits help balance the workload across different stages of the workflow. By limiting the number of tasks that can be in progress at any given time, teams avoid overloading team members and prevent bottlenecks.

2. Improved Focus and Efficiency:

 Positive Impact: WIP limits encourage team members to complete work before starting new tasks. This improves focus and promotes task completion, leading to increased efficiency.

3. Faster Cycle Times:

 Positive Impact: With reduced multitasking and better focus, tasks move through the workflow more quickly. This results in shorter cycle times, meaning that work items are completed faster.

4. Identification of Bottlenecks:

 Positive Impact: When a stage reaches its WIP limit and cannot accept more work, it highlights bottlenecks in the process. This early identification allows the team to address issues promptly, optimizing the workflow.

5. Enhanced Flow Predictability:

 Positive Impact: WIP limits make the flow of work more predictable. Teams can estimate when tasks will move from one stage to another, aiding in better project planning and delivery forecasts.

6. Quality Improvement:

 Positive Impact: With WIP limits, there's more time available for each task in the "In Progress" stage, enabling team members to focus on quality. This can result in a reduction in defects and rework.

7. Reduced Task Switching:

 Positive Impact: Limiting WIP reduces the need for frequent task switching, which can be mentally taxing and decrease productivity. Team members can concentrate on completing tasks before moving to the next one.

8. Increased Collaboration:

 Positive Impact: When WIP limits are reached, team members are more likely to collaborate to resolve bottlenecks or help each other complete tasks. This fosters teamwork and knowledge sharing.

9. Possible Negative Impact on Throughput:

 Negative Impact: In some cases, strict WIP limits can initially reduce throughput, especially if the team is not used to working within limits. However, this is often a short-term effect as the team adjusts to the new approach.

10. Need for Regular Monitoring and Adjustment:

- Important Consideration: Teams should regularly monitor the impact of WIP limits and adjust them as needed. If limits are too high, bottlenecks may persist, and if they are too low, it can slow down productivity.

 In conclusion, WIP limits in a Kanban board have a predominantly positive impact on a team's efficiency, productivity, and cycle time. They promote a smoother workflow, reduce bottlenecks, improve focus, and enhance collaboration. However, it's essential to strike the right balance in setting WIP limits and to adapt them as the team's capabilities and needs evolve.