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**
Date of Performance:	**10/08/23**
Roll No:	**9611**
Team Members:	

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:

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.

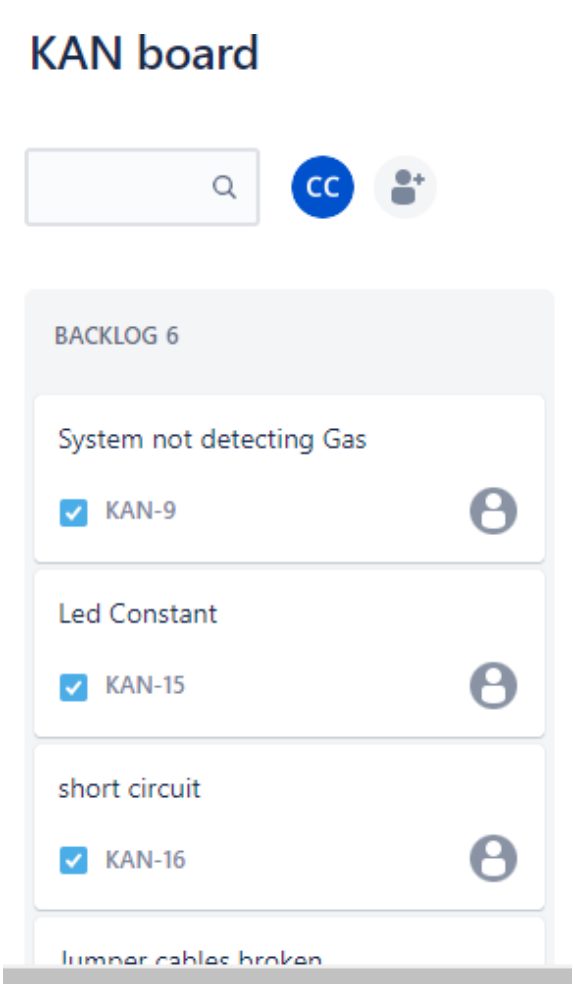
Materials and Resources:

- 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

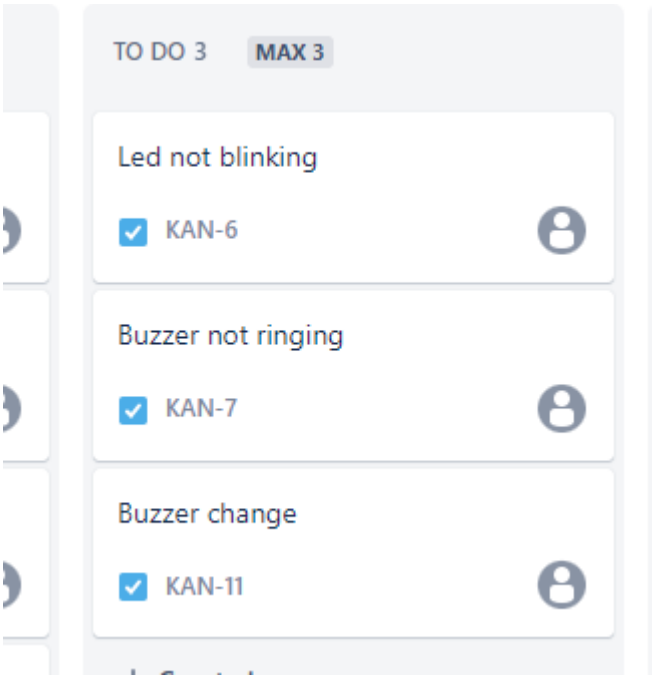
Conclusion: The lab experiment on implementing a project using the Kanban method on the JIRA tool provides students with practical insights into agile project management. By applying Kanban principles and utilizing JIRA's capabilities, students learn to visualize their work, manage flow efficiently, 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.

Timeline

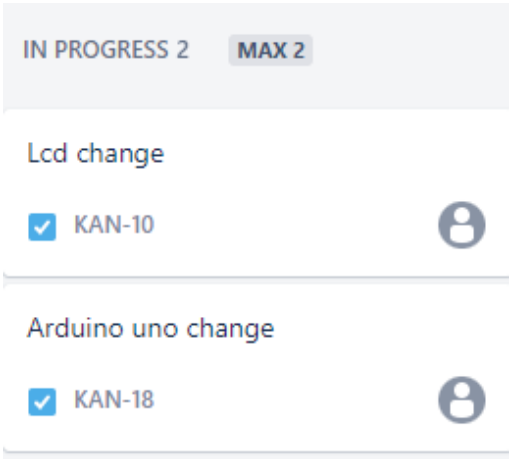
Backlogs in the project



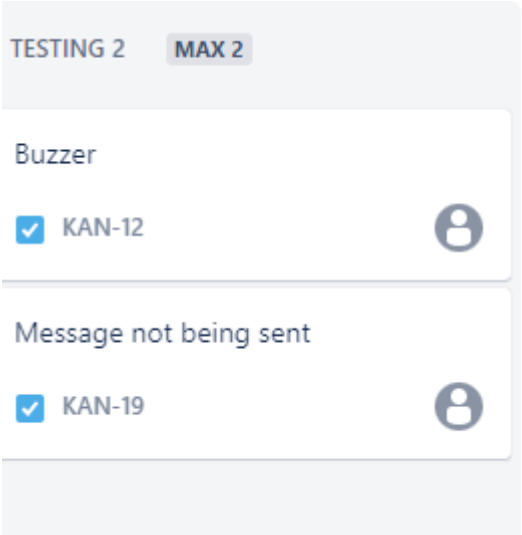
To dos in the project



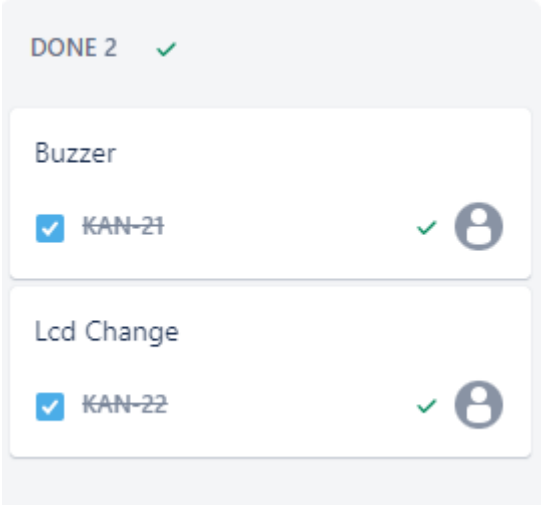
In progress project



Testing phase of project



Done or Delivered Projects



Postlab:

a) Compare and contrast the Kanban and Scrum methodologies in terms of flexibility, adaptability, and workflow management in different project scenarios.

Kanban and Scrum are both popular Agile methodologies used for project management, but they have distinct differences in terms of flexibility, adaptability, and workflow management. Here's a comparison and contrast of these two methodologies in various project scenarios:

1. Flexibility:

- Kanban: Kanban is highly flexible and adaptable. It allows for continuous changes and adjustments throughout the project. You can add or remove tasks from the backlog at any time, making it suitable for projects with evolving requirements or where priorities frequently shift.

- Scrum: Scrum is relatively less flexible during a sprint. Once a sprint starts, the scope and goals are generally fixed until the sprint ends. However, between sprints, there is room for flexibility to adjust the product backlog based on feedback and changing requirements.

2. Adaptability:

- Kanban: Kanban is excellent for environments where adaptability is crucial, such as support or maintenance projects. It can quickly respond to incoming tasks and allocate resources accordingly. Changes can be made on the fly to meet changing customer needs.

- Scrum: Scrum is more structured and may require more effort to adapt within a sprint. However, it provides a predictable cadence and a defined period (sprint) to review and adapt. It is well-suited for projects where a balance between adaptability and predictability is needed.

3. Workflow Management:

- Kanban: Kanban focuses on visualizing and optimizing workflow. Work items move through various stages on a Kanban board, from "To Do" to "Done." This approach is particularly useful for projects with a continuous flow of tasks, where the emphasis is on minimizing work in progress (WIP) and reducing bottlenecks.

- Scrum: Scrum divides work into time-boxed sprints, typically 2-4 weeks long. Within each sprint, there is a set of ceremonies and roles to manage the work. Scrum provides a more structured approach to managing the workflow and encourages teams to complete a defined set of features by the end of each sprint.

4. Project Scenarios:

- Kanban:

  - Well-suited for projects with unpredictable workloads and frequent changes.

  - Effective for maintenance, support, or operations teams.

- Works in environments where continuous delivery is essential.

- Scrum:

  - Ideal for projects with a clear vision and well-defined requirements.

  - Effective for teams that require structured roles, ceremonies, and sprint planning.

  - Works well for projects where stakeholders prefer regular, predictable releases.

b) Analyse a Kanban board in JIRA and propose improvements to optimize the team's efficiency and productivity.

Analyzing a Kanban board in JIRA and proposing improvements to optimize a team's efficiency and productivity requires a deep understanding of the team's workflow, goals, and current challenges.

1. Understand the Current Workflow:

- Begin by thoroughly understanding the team's current workflow. This includes identifying the different stages of work, from backlog to completed tasks, and how work items move through these stages.

2. Identify Bottlenecks and Delays:

- Look for bottlenecks and delays in the workflow. These are stages where work items tend to get stuck or take longer than expected to complete. Common issues include dependencies, resource constraints, or process inefficiencies.

3. Analyze WIP (Work in Progress) Limits:

- Check if the team is effectively using WIP limits on their Kanban board. WIP limits help prevent overloading team members and promote a smooth flow of work. Ensure that these limits are set appropriately for each stage.

4. Monitor Cycle Time and Lead Time:

- Use JIRA's reporting features to track and analyze cycle time (the time it takes to complete a single task) and lead time (the time it takes for a task to move from request to completion). Identify trends and areas where improvements can be made.

5. Review Card Types:

- Examine the types of cards on the board (e.g., user stories, bugs, tasks). Ensure that they are clearly defined, prioritized, and have sufficient information attached to them. This prevents misunderstandings and rework.

6. Promote Continuous Improvement:

- Encourage the team to hold regular retrospective meetings to discuss what's working and what isn't. Use these insights to make incremental improvements to the process.

7. Automate Repetitive Tasks:

- Identify any manual and repetitive tasks that can be automated using JIRA or other tools. Automation can save time and reduce the risk of human error.

8. Implement Visual Metrics:

- Consider adding visual metrics, such as cumulative flow diagrams or burn-up charts, to the Kanban board. These can provide quick insights into the team's performance and help in making data-driven decisions.

9. Foster Collaboration:

- Promote collaboration among team members. Ensure that the board is visible to everyone, and encourage discussions about the work items. Collaboration can help in resolving issues faster and improving overall efficiency.

10. Regularly Review and Adjust:

- Lastly, remember that Kanban is about continuous improvement. Regularly review the Kanban board and its performance metrics, and be prepared to make adjustments as needed.

c) Evaluate the impact of Work In Progress (WIP) limits on a Kanban board and how it affects the team's throughput and cycle time.

Work in Progress (WIP) limits are a fundamental aspect of Kanban and have a significant impact on a team's throughput and cycle time. Here's an evaluation of how WIP limits affect these key performance metrics:

1. Improved Focus and Efficiency:
   - WIP limits restrict the number of work items that can be in progress at any given time for each stage of the workflow. This restriction promotes focus and encourages team members to complete work before starting new tasks.
   - As a result, team members are less likely to become overwhelmed by multitasking, which can lead to reduced productivity and quality issues. WIP limits encourage them to work on tasks one at a time, increasing efficiency.

2. Reduced Bottlenecks:

   - WIP limits help identify bottlenecks and constraints in the workflow. When a stage consistently reaches its WIP limit, it's a clear signal that this area needs attention.
   - By addressing bottlenecks promptly, teams can prevent work from piling up, reduce delays, and maintain a more consistent flow of work through the Kanban system.

3. Shorter Cycle Time:
- WIP limits generally lead to shorter cycle times. Because team members are focused on completing work rather than starting new tasks, work items move through the workflow stages more quickly.
- Shorter cycle times mean that work items are completed faster, allowing the team to deliver value to customers more promptly.

4. Improved Predictability:
- WIP limits provide a sense of predictability. Teams can estimate, with greater confidence, when a work item will move from one stage to the next and when it will be completed.
- This predictability is valuable for managing customer expectations, making commitments, and planning releases more accurately.

5. Enhanced Quality:
- By reducing the number of tasks in progress, WIP limits enable team members to focus on quality rather than rushing through tasks to clear their queues.
- Higher-quality work results in fewer defects and rework, ultimately improving the overall efficiency and productivity of the team.

6. Continuous Improvement:
- Teams regularly review and adjust WIP limits based on their observations and performance metrics. This iterative process fosters a culture of continuous improvement.
- Over time, teams can fine-tune their WIP limits to achieve optimal flow and efficiency.

7. Visualize Work Overload:
- WIP limits make it visually apparent when a stage is overloaded. When a limit is consistently exceeded, it serves as a red flag, prompting the team to investigate and take corrective actions.

8. Team Collaboration:
- WIP limits encourage collaboration among team members. When a team member finishes a task, they can help a colleague who may be facing challenges, ensuring that work items continue to move smoothly.