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:	**10/8/2023**
Roll No:	**9604**
Team Members:	**9561,9540**

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.

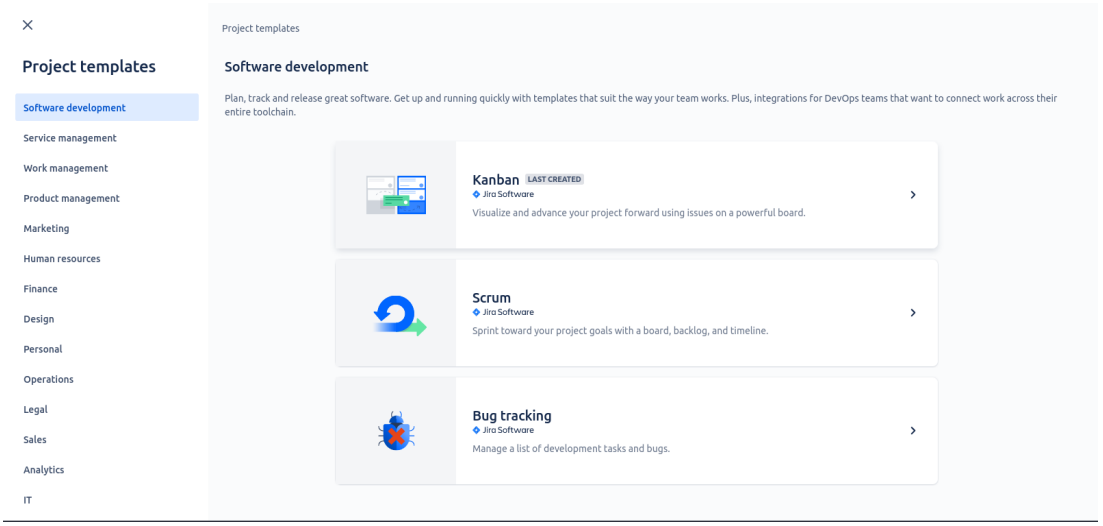
- Develop team collaboration skills by continuously improving the workflow through regular team meetings.
- Appreciate the importance of visualizing and managing work items for better project management.

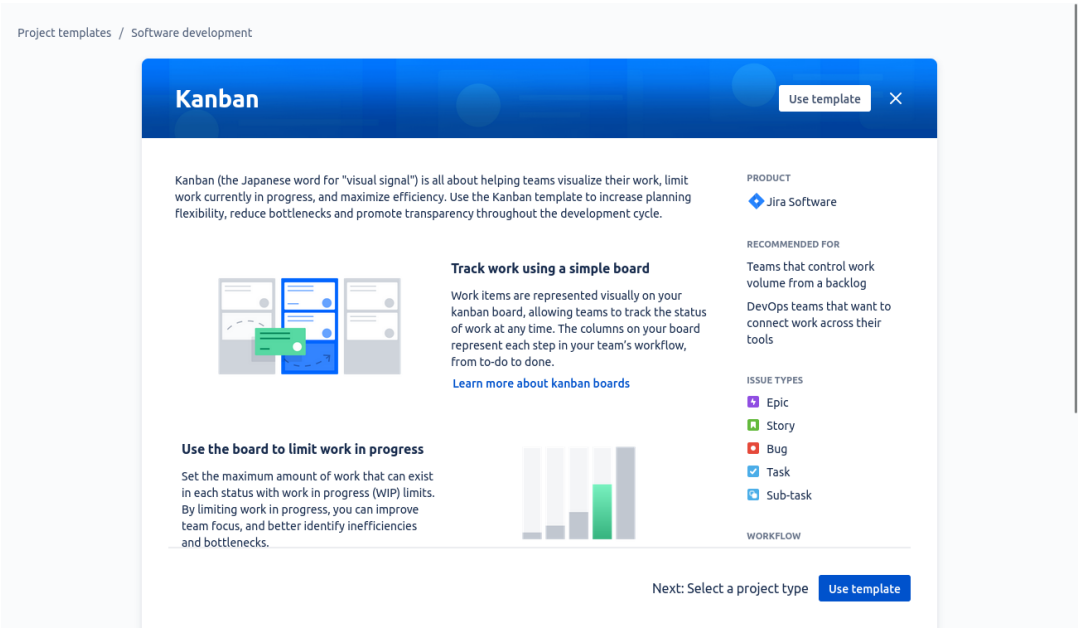
Pre-Lab Preparations: Before the lab session, students should familiarize themselves with the Kanban method and the basics of the JIRA tool. They should review Kanban principles, visualizing workflows, and the features of JIRA relevant to Kanban implementation.

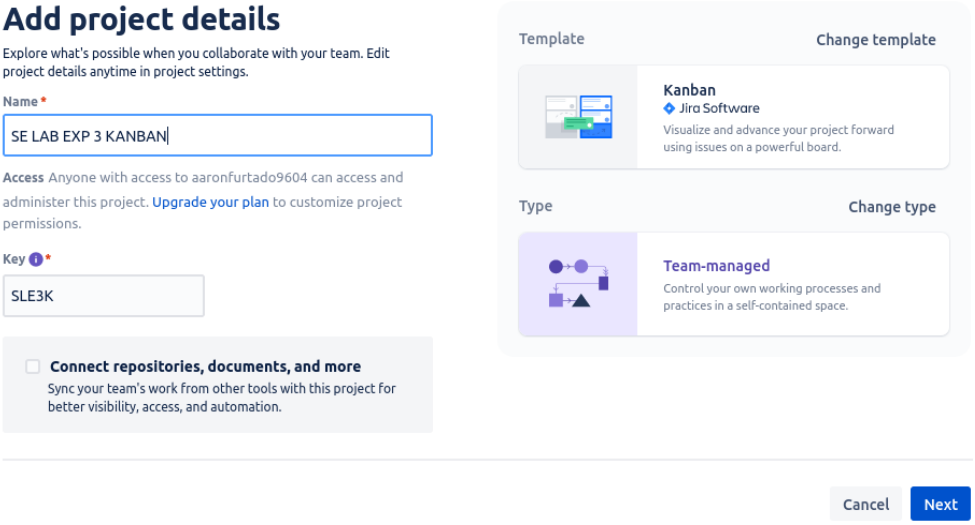
Materials and Resources:

- 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

1. Creating Project

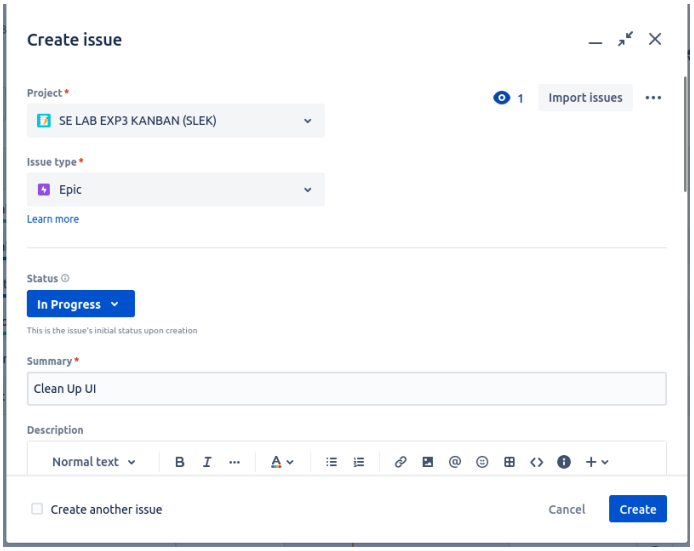


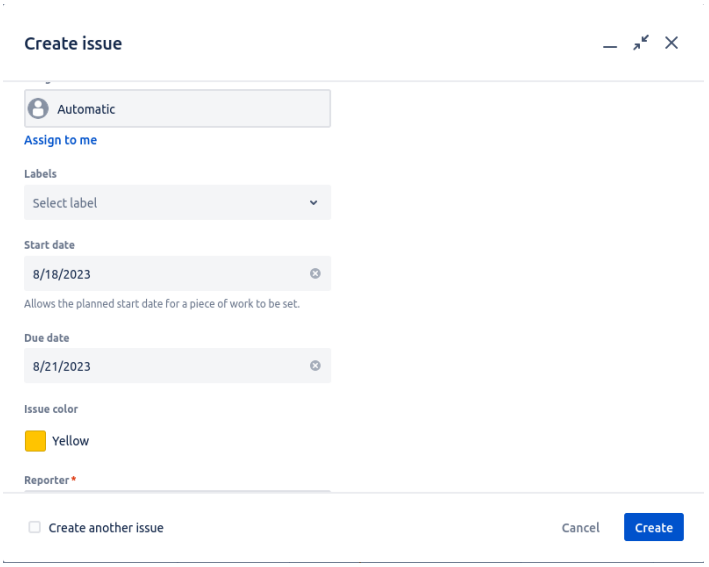


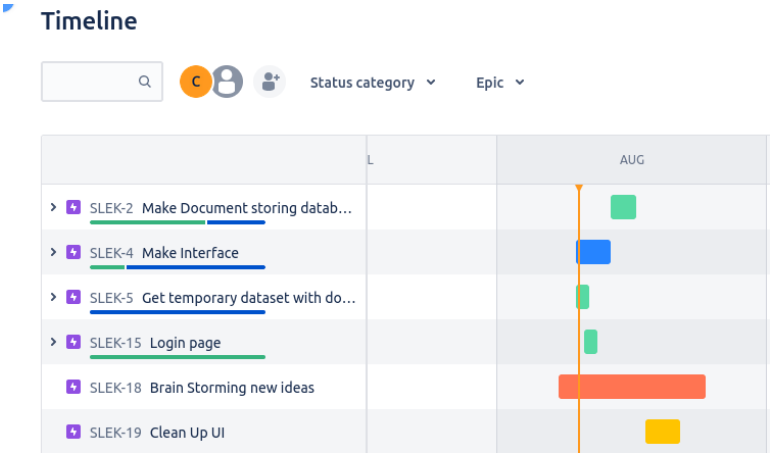


2. Managing Workflow

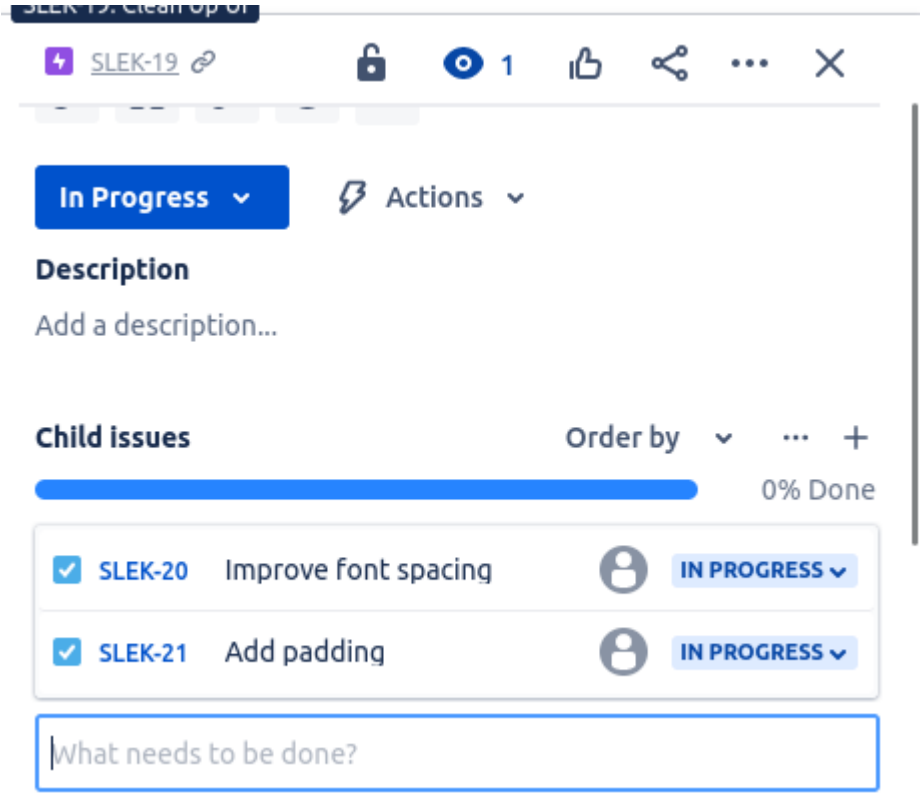
2.1 Creating Epic



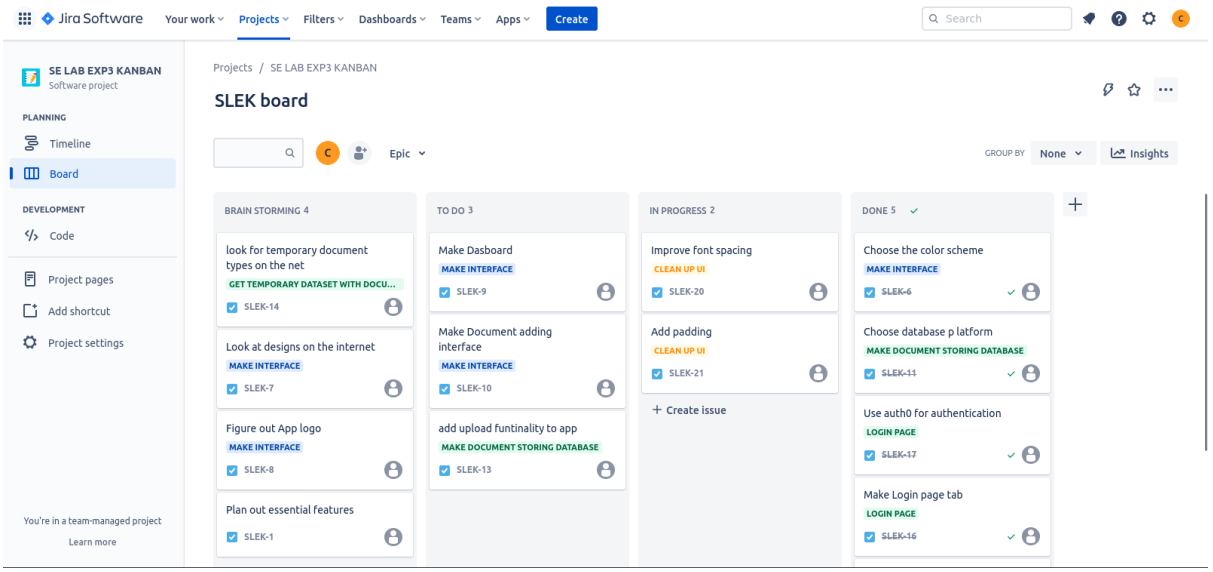




2.2 Create Subtasks

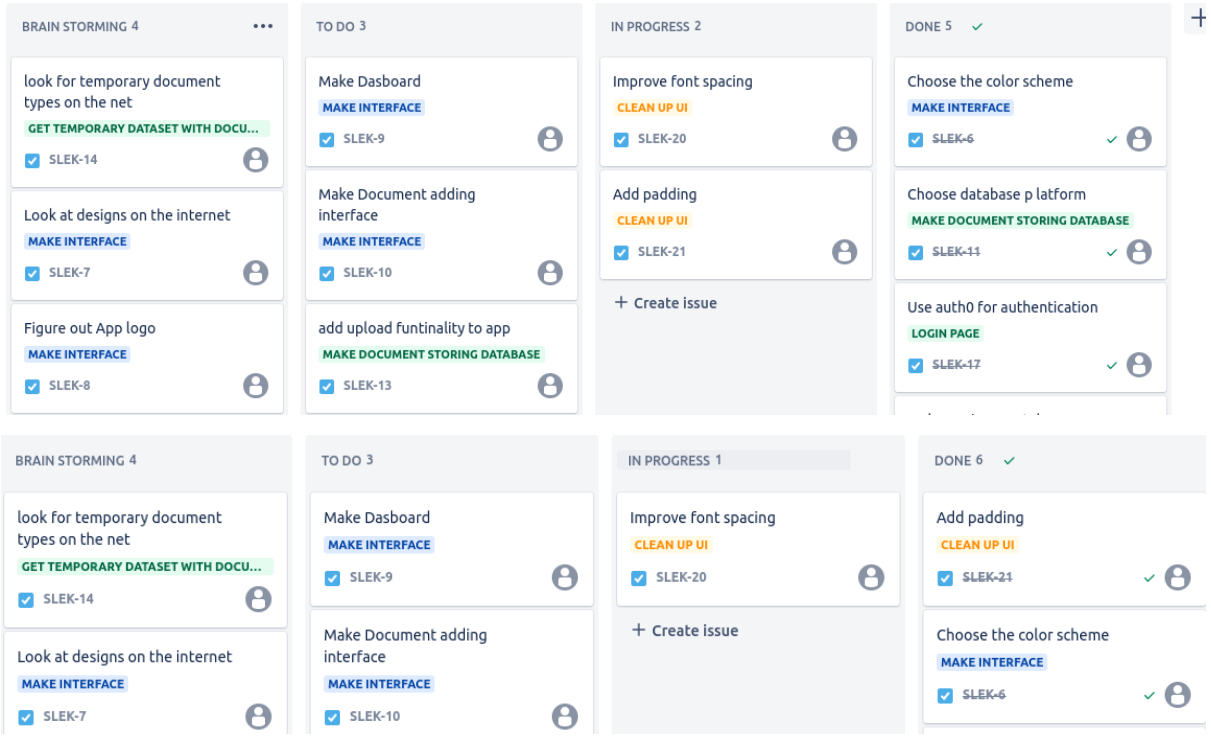


3. View Board



4. Columns According to Workflow(Brainstorm created)

5. Move tasks from one column to another based on requirement



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.

Postlab

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

Ans:

Kanban and Scrum are both popular Agile methodologies used for managing and organizing projects, but they have different approaches and strengths when it comes to flexibility, adaptability, and workflow management in various project scenarios. Let's compare and contrast them in these aspects:

1. Flexibility:

- Kanban: Kanban is highly flexible and adaptable. It focuses on visualizing the workflow and allows for continuous, incremental improvements. Teams using Kanban can make changes to the process as needed without strict time-boxed iterations. This flexibility makes Kanban suitable for projects with unpredictable or changing requirements, where the ability to quickly adjust priorities and respond to new demands is crucial.

- Scrum: Scrum provides a structured framework with predefined roles, events, and artifacts. While it offers some flexibility within its time-boxed iterations (sprints), major changes to the process are typically discouraged during a sprint to maintain stability. Scrum is well-suited for projects with a more defined scope and where stakeholders can commit to a set of requirements for each sprint.

2. Adaptability:

- Kanban: Kanban is designed to adapt to changing circumstances and priorities. Work items flow through the board based on demand, and the process can be adjusted in real-time. This adaptability makes Kanban effective for projects that involve continuous improvement, support/maintenance tasks, or situations where requirements evolve frequently.

- Scrum: Scrum provides a stable structure within each sprint, making it less adaptable during the sprint itself. Changes to the sprint are typically discouraged to maintain predictability. Adaptation in Scrum is mainly driven by refining the

backlog and adjusting priorities between sprints. Scrum is well-suited for projects where a balance between structure and adaptability is required.

3. Workflow Management:
- Kanban: Kanban focuses on visualizing and optimizing the flow of work. Work items move through different stages of the process, and the emphasis is on minimizing bottlenecks and maximizing efficiency. Kanban is effective for projects that require a continuous stream of work, such as customer support, where tasks have varying sizes and priorities.

- Scrum: Scrum provides a fixed iteration length (sprint) during which a predefined set of backlog items are completed. The focus is on delivering a potentially shippable product increment by the end of each sprint. Scrum's structured approach to workflow management is suitable for projects that benefit from regular, time-boxed releases and where a defined cadence is important.

4. Project Scenarios:
- Kanban: Kanban is well-suited for projects with evolving requirements, support/maintenance tasks, continuous improvement initiatives, and situations where work items have varying priorities and sizes.

- Scrum: Scrum is suitable for projects with a relatively stable scope, where regular iterations and predictable releases are important. It works well for teams that require a structured approach to manage and deliver work.

In summary, Kanban offers greater flexibility and adaptability for projects with evolving requirements and a need for real-time adjustments. It excels in scenarios where continuous improvement and optimizing workflow are crucial. On the other hand, Scrum provides a more structured framework with fixed iterations and a focus on regular, predictable releases. It is suitable for projects that can commit to a set of requirements for each sprint and value a well-defined cadence. The choice between Kanban and Scrum depends on the specific needs and characteristics of the project and the team.

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

Ans:

While I can't directly visualize or interact with your specific Kanban board in JIRA, I can provide you with some general guidelines and suggestions for optimizing your team's efficiency and productivity using a Kanban board in JIRA. Keep in mind that these recommendations may need to be adapted based on your team's unique context and needs.

1. Visualize Workflow Clearly:
- Ensure your Kanban board accurately represents your team's workflow stages. Each column should represent a specific stage in your process (e.g., "To Do," "In Progress," "Review," "Done").
- Consider using color coding to make it easier to distinguish between different types of tasks or priorities.

2. Limit Work in Progress (WIP):
- Set WIP limits for each column to prevent bottlenecks and encourage a steady flow of work. This helps to balance workloads and reduces multitasking.
- Regularly review and adjust WIP limits based on the team's capacity and throughput data to optimize the flow of work.

3. Define Clear Policies:
- Each column should have explicit policies that define what type of work is allowed in that stage and what criteria must be met for a task to move to the next stage.
- Clear policies minimize confusion, reduce unnecessary back-and-forth, and improve decision-making.

4. Focus on Cycle Time:
- Measure and track the cycle time (the time it takes for a task to move from one stage to another) for different types of work items.
- Analyze the cycle time data to identify bottlenecks, streamline processes, and optimize the workflow.

5. Continuously Improve:

- Schedule regular retrospective meetings to discuss what's working well and what can be improved in your Kanban process.
- Implement changes based on feedback and data analysis to enhance your team's efficiency and effectiveness.

6. Visualize Blocked Tasks:
- Implement a clear mechanism to highlight and visualize blocked tasks on the Kanban board. This helps the team identify and address issues quickly.
- Consider adding a "Blocked" column or using color-coding to indicate blocked tasks.

7. Use Swimlanes and Filters:
- Utilize swimlanes to categorize tasks based on different criteria, such as priority, assignee, or type of work.
- Set up filters to easily view specific subsets of tasks or to focus on high-priority items.

8. Integrate Automation:
- Explore JIRA automation features to automate repetitive tasks or notifications, such as moving tasks to the next stage when specific conditions are met.
- Automation can reduce manual effort and increase the speed of task progression.

9. Prioritize Backlog Items:
- Keep the backlog well-groomed and prioritize tasks based on business value and urgency.
- Ensure that high-priority items are readily available for the team to pull into the "To Do" column.

10. Provide Training and Support:
- Ensure that all team members understand how to use the Kanban board effectively and efficiently.
- Offer training sessions or documentation for new team members and provide ongoing support.

Remember that the key to optimizing efficiency and productivity lies in regular monitoring, analysis, and continuous improvement. Adapt these suggestions to

fit your team's specific context and monitor the impact of changes to ensure you're achieving the desired outcomes.

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.
Ans:
Implementing Work In Progress (WIP) limits on a Kanban board can have a significant impact on a team's throughput and cycle time. WIP limits are a core concept in Kanban, designed to optimize workflow, reduce bottlenecks, and improve overall efficiency. Let's evaluate how WIP limits affect these aspects:

1. Throughput:
Throughput refers to the rate at which work items are completed and delivered. WIP limits can impact throughput in the following ways:

- Improved Focus: By setting WIP limits for each workflow stage, the team is forced to concentrate on a limited number of tasks at a time. This promotes focus and prevents the team from taking on too much work simultaneously.

- Reduced Multitasking: WIP limits discourage multitasking, where team members juggle multiple tasks concurrently. This reduction in context switching can lead to more efficient work, as it allows team members to complete tasks more quickly.

- Steady Flow: WIP limits maintain a steady flow of work through the process. As tasks are completed, new tasks can be pulled into the workflow, ensuring a balanced workload distribution.

2. Cycle Time:
Cycle time is the duration it takes for a work item to move through the entire workflow from start to finish. WIP limits influence cycle time as follows:

- Reduced Variability: WIP limits help reduce variability and unpredictability in cycle time. By preventing excessive work accumulation at different stages, tasks move more consistently and predictably through the workflow.

- Faster Delivery: With WIP limits in place, tasks are more likely to be completed sooner, as there's a focus on completing existing tasks before starting new ones. This can lead to shorter cycle times and faster delivery of work items.

- Identification of Bottlenecks: WIP limits can highlight bottlenecks in the workflow, as stages that consistently reach their limits indicate areas that may need attention. Identifying and addressing bottlenecks can lead to shorter cycle times.

3. Overall Efficiency:
Implementing WIP limits contributes to the overall efficiency of the team's workflow:

- Optimized Flow: WIP limits encourage the team to optimize their workflow, streamline processes, and collaborate to remove obstacles that hinder progress. This continuous improvement leads to increased efficiency over time.

- Reduced Lead Time: Lead time, the time from task request to task completion, is often reduced with WIP limits due to the steady and focused flow of work. This can result in quicker response times to customer needs.

- Enhanced Predictability: WIP limits make the workflow more predictable, allowing the team to forecast completion dates more accurately and make more informed commitments to stakeholders.

In summary, WIP limits in a Kanban system have a positive impact on a team's throughput and cycle time by promoting focus, reducing multitasking, maintaining a steady flow of work, and enhancing overall efficiency. By creating a controlled and balanced workflow, teams can achieve faster delivery, shorter cycle times, and improved predictability, ultimately leading to higher customer satisfaction and better alignment with business goals.