# 6 Practices: Kanban Rules of Engagement

Published on 20 March 2023 at 11:45

In the realm of Agile project management methodologies, Kanban stands out for its simplicity, flexibility, and emphasis on visualizing work and optimizing flow. Originating from lean manufacturing principles, Kanban has gained popularity across various industries and teams seeking to improve their productivity and efficiency. Central to the success of Kanban is understanding and adhering to its rules of engagement. Let's delve into these rules and explore how they can help teams unlock the full potential of Kanban.

## 1. Visualize Your Workflow:

The first rule of Kanban is to visualize your workflow. This means creating a visual representation of the work items moving through different stages of the process, typically on a Kanban board. The board is divided into columns representing each stage of the workflow, with cards or sticky notes representing individual work items. By visualizing the workflow, teams gain insight into the status of work, identify bottlenecks, and make informed decisions to improve flow.

A diagram of a circular chart

AI-generated content may be incorrect.

## 2. Limit Work in Progress (WIP):

The second rule of Kanban is to limit work in progress (WIP). Setting WIP limits ensures that the team focuses on completing existing work before taking on new tasks, preventing overloading and multitasking. WIP limits help maintain a steady flow of work through the system, reduce lead times, and improve predictability. Teams can experiment with adjusting WIP limits based on capacity and throughput to optimize flow and productivity.

## 3. Manage Flow:

The third rule of Kanban is to manage flow. Kanban encourages teams to prioritize work items based on customer value and urgency, ensuring that high-priority items flow smoothly through the system. Teams monitor and manage flow by visualizing queues, tracking cycle times, and identifying and addressing bottlenecks or blockers promptly. By managing flow effectively, teams can deliver value to customers faster and more predictably.

## 4. Make Process Policies Explicit:

The fourth rule of Kanban is to make process policies explicit. Teams define and document the rules, policies, and guidelines governing how work is done and how decisions are made. Process policies may include criteria for prioritizing work, definitions of done, quality standards, and escalation procedures for handling exceptions. Making process policies explicit promotes transparency, consistency, and alignment within the team.

## 5. Implement Feedback Loops:

The fifth rule of Kanban is to implement feedback loops. Kanban encourages continuous improvement through feedback mechanisms that enable teams to reflect on their process, identify areas for improvement, and make adjustments iteratively. Feedback loops may take various forms, such as regular retrospectives, daily stand-up meetings, customer feedback sessions, or metrics and analytics dashboards. By embracing feedback loops, teams foster a culture of learning, experimentation, and adaptation.

## 6. Improve Collaboratively, Evolve Experimentally:

The sixth rule of Kanban is to improve collaboratively and evolve experimentally. Kanban encourages teams to continuously experiment with their process, practices, and tools to find what works best for them. Teams collaborate cross-functionally to identify improvement opportunities, test hypotheses, and implement changes incrementally. By embracing a mindset of continuous improvement and experimentation, teams can adapt to changing needs and challenges effectively.

Kanban boards are used to visualize workflow as a living status report. Stakeholders and team members can see which stories are currently being worked on and where they are in the process. They can also see a prioritized list of backlog items that have yet to be selected for work. Kanban boards are sequential in nature, displaying to all team members and stakeholders each stage and gate that a project must pass through before it is finally available to end users. Kanban boards improve resource utilization while reducing confusion.

Kanban aims to give team members just enough work so that the team is always working at full capacity. Kanban teams benefit from flexible planning, sharper focus, and complete transparency because whatever is on the top of the backlog is the top priority. Items that developers are working on can be seen. Kanban is ideal for operational teams that are focused on continuous delivery while dealing with shifting priorities.

## Kanban Board

The Kanban board is made up of lanes (stages) through which work will flow to be completed. The project manager and team can decide which lanes to include in the Kanban board. The team should decide whether to move the stories at any time or only during the daily standup meetings.

 An example of how a Kanban board is as follows:

A group of sticky notes on a board

AI-generated content may be incorrect.

* **New -** As problems/issues arise, they are added to the **New** lane as a user story. Stories should be small enough to be completed in a reasonable amount of time (decided by the team). If the scope of the stories exceeds what has been decided, they should be divided into smaller stories.  During the backlog grooming meeting, the user story gets reviewed, prioritized and moved to the **Backlog** lane.
* **Backlog** - Prioritizing the items in the backlog is the responsibility of the project manager or product owner. Items with higher priority are listed first, followed by items with lower priority.  Team members choose items at the top of the backlog to work as they are highest in priority.
* **In Progress** – When a user story is chosen to be worked on, the developer assigns it to themself and moves it to the Kanban board's**In Progress** lane. To document what is required to complete the user story, the developer should add individual tasks that must be completed to the user story. This will serve as a check list for the developer and will also inform the peer review developer and tester about the changes made to complete the user story.
* **Peer Review** – When the developer completes the user story, it will be moved to the Peer Review Lane. Another developer who chooses to do the peer review will be assigned the story or they can assign the user story to themself. When the reviewer has finished the peer review, he/she will move the user story card to the testing lane.
* **Testing**– The user story will be moved to In Test once it has passed the peer review process. A QA team will assign the user story to themself and test it to ensure that the acceptance criteria are met.
* **Done** – The user story will be moved to done once it has passed testing.
* **Blocked** – If a user story is blocked at any point during the Kanban process, it will be moved to the blocked lane. When items become blocked, team members are encouraged to assist in unblocking the story. Unblocking a story usually entails gathering more information from stakeholders about the issue that is preventing the task from being completed.

## Managing the Kanban Board

1. **Minimal Changes to the Kanban Board** - Once the team has agreed on the stage lanes, there should be little to no change to the stages. Only the project manager or product owner should be able to rearrange, add, or remove columns. Only the project manager or product owner should be allowed to prioritize items in the backlog.
2. **Entering a new story into the backlog** – A new story can be added to the Kanban board by any team member. In the backlog grooming meeting, user stories will be reviewed and prioritized and moved to the backlog lane by the project manager and product owner.  Using new and backlog lanes takes some of the confusion as to whether the story has been groomed and prioritized.
   1. The following format should be used while adding a user story.   ***As a****[type of user],****I want****[a goal or objective],****so that****[customer benefit or value].*
   2. Acceptance Criteria must be included. Acceptance criteria are a set of conditions that must be met for a user story to be completed.
   3. When a story is moved to the backlog, it should not be assigned to anyone because the user story will be assigned once it has been chosen by a developer.
3. **Estimation and Prioritization of items in the backlog**
   1. Regular backlog review/grooming meetings are required to estimate and prioritize the user stories in the backlog.
   2. Any questions from the team can be added to the story. As the questions are answered and the story is updated, this will add a trail of communication.
   3. The team can reach an agreement on the size of a story. The stories can be sized using T-shirt sizing or Fibonacci sequences. If the stories are too large, they should be divided into smaller stories that must also be sized and prioritized.
   4. As the priority of work changes over time, the project manager or product owner can move items in the backlog up or down the prioritization list on a regular basis.
4. **Selecting a story to work on**
   1. Developers should choose stories at the top of the backlog as they are higher in priority than ones that are lower in the list.
   2. The developer will assign the story to themself and move it to the In-progress lane of the Kanban board.
   3. The developer will add tasks to the story that will need to be done to get the story completed.
   4. Once the developer finishes working on the user story, he/she will move the user story to the Peer Review stage.
5. **Peer Review** – When a developer moves a user story to the peer review stage, any developer other than the one doing the initial change can select the story to peer review. The developer doing the peer review will assign the user story to themself.  After the peer review is completed, the developer doing the peer review will move the story to the In Test Stage.
6. **Test**– QA testers select a story that is in the In Test stage and assign it to themself. Once the QA testing is complete, the story is ready to be demonstrated to the Product Owner (PO) for approval.  After PO approval, the user story is moved to the Done stage.  If the tester or product owner finds something that is not right during the testing and acceptance phase, the user story is moved back to the In Progress stage and reassigned back to the original developer to correct any issues.  Any testing and acceptance issues should be clearly noted in the user story.
7. **Done –** When the user story has been accepted by the product owner, they will be moved to the done lane.
8. **Blocked –**When a developer is unable to continue working on a user story, the user story can be marked as blocked by moving it to the blocked lane. It is critical to include notes in the user story explaining why the user story is blocked so that others on the team can assist in unblocking the story.

## Conclusion

In conclusion, mastering Kanban requires understanding and adhering to its rules of engagement. By visualizing workflow, limiting WIP, managing flow, making process policies explicit, implementing feedback loops, and embracing a culture of collaborative improvement and experimentation, teams can harness the power of Kanban to optimize their processes, deliver value to customers, and achieve their goals with greater efficiency and effectiveness.

#KanbanPractices #VisualizeWorkflow #LimitWIP #ManageFlow #ExplicitPolicies #FeedbackLoops #ContinuousImprovement #AgileProjectManagement #LeanThinking #KanbanBoard #WorkflowOptimization #TeamCollaboration #AgileTransformation #ProcessEfficiency #KanbanMethod