cprime

Kanban Quick Start Guide

By: Eugene Lai



cprime.com | US +1.877.800.5221 | UK +44.0.203.811.0424 | ©2021 Cprime Inc. No copying without express written permission.



New Agile teams with limited experience usually encounter many of the same challenges during initial stages of formation. One of the biggest challenges is the formation of the operational workflow for managing work.

This quick start guide is designed to empower teams to leverage the Kanban model and to launch the team as quickly as possible. In this paper, we will explore key factors to consider when selecting the type of Agile model to adopt (e.g. Scrum or Kanban) as well as the optimal way to launch a team with minimal effort. We will also explore a general iterative process for optimizing flow, which is intended to enable new Kanban teams to learn how to adopt a continuous improvement mindset. Lastly, we will discuss techniques for maintaining and refining a Kanban board and tactics for addressing inefficient workflow conditions.



Part 1: How Mature is This Agile Team?

This one may cause you to raise your eyebrows a bit, but the level of maturity of the team in terms of experience with Agile/Scrum is a big factor to consider. Many experienced practitioners believe that mature teams are more likely to be successful following Kanban, while teams that are new (or less experienced) in Agile will be more successful using Scrum. Due to lack of experience or understanding, often times new Agile practitioners believe Kanban is simpler, hence it is easier to adopt. While Kanban may appear simpler in theory, it does not provide the structure and rigor that a new Agile team usually needs. On the other hand, Scrum provides many core practices that are simple to learn and understand. Although difficult to master, Scrum offers more guidance that helps less experienced teams get going and start accumulating small victories that build confidence.

For a brand-new Agile team that has never applied the Agile approach before, Scrum is generally a better choice over Kanban due to the additional structure. New teams often need more discipline so that they can adopt to a new way of thinking and learn to become productive relatively quickly. Scrum offers basic constructs such as Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective which provides a foundation with which a team can begin to practice Agile.

The table below provides additional scenarios and team characteristics to consider when selecting the preferred Agile approach.

Scenario/Team Characteristics	Potentially Preferable Agile Approach
The team is consistently disrupted by unplanned work after it has committed to a plan	**Kanban**
The team is expected to regularly react to high-priority requests and respond to them very quickly (within one week or less)	**Kanban**

The team consistently struggles with the amount of time required to conduct formal planning and review sessions	**Kanban**
The team consists of primarily part-time resources with unpredictable capacity/ availability	**Kanban**
The team is more productive when specific goals and/or milestones are determined and set as targets on a regular basis	**Scrum**
The customer expects regular inspection of the work products completed by the team at predetermined intervals	**Scrum**
The team prefers to track metrics that are available through a tool without doing additional work	**Scrum**
The customer expects a long-term, multi-month roadmap or release plan with a forecast of deliverables	**Scrum**

Table 1 – Factors to consider when selecting an Agile approach

Changing Agile Approaches - Good or Bad?

Some teams begin their Agile journey by following Scrum because it made sense to adopt the most popular Agile approach as a starting point. However, after encountering issues such as not gaining the benefit they expect, they decide to switch from Scrum to Kanban due to perceived lower level of complexity.

While changing methods should be an option for any team, this decision should not be made without proper understanding of why the current approach is not working. If the root cause of the current issues is related to culture, organizational constraints, or technology, changing the framework/model will likely not solve the problem for the team.

For example, if the team is struggling due to the team members being assigned too many projects at the same time, a change in the Agile model will not improve the situation.

Changing the approach should be targeted specifically to address concrete issues such as address excessive unplanned work, improving ability to forecast future completion of backlog, etc.

In summary, choosing between Scrum or Kanban may seem like an easy choice, but there is more involved than expected. Carefully evaluate where the team is currently at, and what will help them learn and successfully deliver positive outcomes in the shortest amount of time.

It is okay to change approaches when the time is right, as long as the team clearly understands the purpose of making such a change, and is confident that a different approach will improve the probability of team success.

Part 2: What Kind of Work is the Team Going to Do? Planned Work or Unplanned Work?

The type of work the Agile team expects to do makes a tremendous difference in terms of which model will give the team the most benefit.

If this team is doing new development (i.e. building a new system, service, or solution), the team will likely benefit from knowing the average velocity of the team over time so the future forecast can be improved. In this case, Scrum may be a better choice than Kanban, because Scrum offers a simpler method for computing team velocity. Conversely, if the team is primarily doing unplanned work, such as defect repair or production support type of work that is introduced sporadically and without warning, Kanban may be superior for this team.

Part 3: How Big are the Work Units?

In addition to the type of work the team is doing, the size of the work units is important to understand also. What do I mean by "size" of the work? In this context, I'm referring to the relative, approximate effort and complexity of the work item. If the work generally takes several days to be completed, Scrum may work better. However, if the work is smaller in scope and complexity, small enough to be completed within 1 or 2 days generally, Kanban may be a better fit.

Generally, it is acceptable to have work items of different sizes for either Kanban or Scrum; it is usually beneficial to have a mixture of work with a variety of difficulty so that the team has more opportunity to move work around within the sprint to maximize overall output.

Design Patterns for Kanban Boards

Most new Agile teams who adopt Kanban or Scrum encounter an important decision at some point during initial startup – How to set up the team board.? One of the biggest
challenges during this process is the decision on how many columns should a team create for this board, which is heavily influenced by the underlying workflow.

Many new Agile teams make the common mistake of creating too many columns on their board very early in the process, which means they create a process flow with several steps (or states). While there is benefit in having a very detailed process that tracks the various stages of each work item, there is a cost – the more columns/states are on the board, the more effort is required to manage this data, thus increasing the administrative burden on the team.

The concept of the Kanban board is essentially the same whether a team is following Scrum or Kanban Agile methods, so for simplicity, the term "Kanban board" will be used for this guide. The main purpose of a Kanban board is to enable the team to:

1. Visualize all the work that the team must do.
2. Manage the work on an operational basis.
3. Identify inefficiencies in the form of bottlenecks in the delivery process.
4. Refine the process/steps as needed to improve efficiency and/or quality of the output to the customer.

If a board has too many columns/steps, the risk of the increased overhead elevates, and so does the risk of the team losing confidence in the value of the board. If the team decides to abandon the board, instead of having very detailed visibility into progress, the team will lose visibility due to the board not being used. This is a scenario that teams must work hard to avoid.

Most Agile Management Tools start with a default set of states, usually 3 or 4. If the team uses Jira, Jira's Kanban board is very simple, and has only 3 steps: "To Do", "In Progress", "Done". Most software teams will add to this and create a custom workflow that may include additional states such as "Analysis", "Development", "Unit Test", "In Test", "Test Complete", etc. Once customization of the flow begins to take a life of its own, it is easy to keep going and let things get out of control.

How do we know when to stop adding states/columns? How many is enough?
That depends on many factors.

One of the key things to consider is how often is the work being completed from start to finish; from the time that a request is put on the board to the time that the request (or work item) is fulfilled and delivered to the customer, does this timespan typically take 1 day, 1 week, or longer?

A popular recommendation is that the longer the cycle time is (time required to complete a work item), the more columns is preferred on the board. Note that this is very subjective and will vary depending on the type of work the team is doing, but generally, the longer time the work item is represented on the board, the likelihood of getting benefit from more granularity increases. A counterpoint to this recommendation is if the work is taking too long, the work may have been defined too broadly, which means it could be decomposed into smaller units of work in order to improve flow through the process. Breaking work down into smaller sizes is usually beneficial for most teams.

For a new team practicing Kanban, it is generally advisable to start with no more than 4 states so that the team can gain a level of familiarity with how things work. If the team observes that work is consistently "stuck" in a particular state for several days, when the total number of active work items is fairly small, the team should consider experimenting by adding one column (and state) to help the team understand where the bottleneck is. The team may find that the bottleneck in the process flow is actually different from what was had expected.

For example, if the board only has 3 states: "To Do", "In Progress", and "Done", there will likely be many work items sharing the same state "In Progress", but are in varying degrees of maturity; some work may have been "code complete" while others are waiting for "documentation" to be done. In this case, if the state is changed from "In Progress" to "In Development" and "In Test", the board could reveal that there is too much work piled up within the "In Test" state because there are more developers than testers, and the development is being completed faster than the Test Engineers can keep up with.

The general rule of thumb is to start simple, experiment, inspect and then adapt. By following this approach, most teams should be able to quickly hone in to the optimal process flow for that specific team.

Common Techniques for Optimizing Flow of Work/Value to the Customer

By utilizing a Kanban board, the team is able to gain valuable information about how it is performing. Various metrics can be derived and tracked, which can enable the team to find better ways of doing work. The following are a few common techniques for optimizing team output; note that these techniques may be applied to both Kanban and Scrum teams.

Optimization Technique #1 – Swarm on tasks

Traditional team workers are accustomed to being assigned tasks to be worked on an individual basis. As a result, when work is completed, the individual requests another assignment from the "work manager" who delegates the task. Within an Agile environment, the flow of work is optimized by a less rigid approach which encourages a "pull" approach that reduces wasteful idle time. This can also be optimized further through a collaboration technique of "swarming".

This technique encourages teamwork through paired work; multiple team members work together to complete tasks in an efficient manner, which generally lead to higher quality as well as more effective knowledge transfer between team members. This type of collaboration also improves communication and morale.

Optimization Technique #2 – Limit WIP (Work in Progress)

Controlling the amount of work being done at any given time is a core tenet of optimizing flow. Through the use of the Kanban board, the team can quickly understand bottlenecks in the process and take steps to eliminate such an obstacle. For example, if the team is consistently completing coding tasks faster than the code can be peer-reviewed by the reviewers, the team may consider one of the following actions:

- Reduce the amount of coding being done (i.e. limiting the WIP for coding tasks).
- Increase the WIP for the peer review step.

Optimization Technique #3 – Create multiple service levels

In some situations, it may benefit the team to categorize the work into multiple separate workflows with different priority levels. By segregating "standard" and " expedited" work items into their own distinct flows on the Kanban board, the team is able to more easily identify high-priority work items and focus on completing those items first and bringing the benefits to the end customers as early as possible. Some teams may inject excessive complexity by creating several levels of service (i.e. 3 or more). For new teams, it may be advantageous to start simple with no more than two service levels so that the team can learn to operate in an efficient manner.

Cyclical Approach for Optimizing Flow and Throughput

While the basic concepts of Kanban and flow are generally easy to apply, many teams that are new to this approach struggle with a key technique that is often missed – maximizing the work delivered through fine-tuning. A Kanban process flow is designed to be adaptive and fluid, which allows practitioners the opportunity to discover and experiment with different configurations in order to achieve the desired outcomes.

Optimizing the work completed and delivered is the most important objective of following a flow-based system, but the system is so flexible that it can often be challenging to decide how to accomplish this goal. Following the principles of empirical mindset, optimization of flow will usually take the form of a continuous improvement approach, which typically follows a cyclical pattern of operation.

Step 1 – Measure current throughput (Avg Cycle Time)

Once the Kanban flow with key steps have been identified, the team should follow this process for several days (or cycles/Sprints) in order to collect data on its performance. Average Cycle Time is generally the simplest and most effective metric to start with. This will enable the team to understand its initial throughput that can be used for future comparison. As changes are made to the flow, the team will compare new measures to the previous baseline to assess improvements (or degradation) in a quantitative manner.

Step 2 – Identify Bottleneck

Through the natural process of doing work and delivering output, the team will likely discover a bottleneck in one of the steps. A "bottleneck" is a condition where the WIP limit has been reached for a process step, and the team is struggling to move the work to the next step. Within the Kanban board, this condition is identifiable from a process step being fully-occupied, while the following step exhibits excess capacity for work. This occurrence should alert the team to consider changes to improve the end-to-end flow. If there is more than one bottleneck on the board, it is beneficial to focus on the step that has a larger impact to the team; optimizing the biggest inefficiency will provide the biggest benefit.

Step 3 – Apply swarming at the bottleneck

If a high volume of work has accumulated at the bottleneck, the team needs to smooth out the flow by reducing that work at the point of the bottleneck. One technique for achieving this is to focus team resources specifically at this part of the process flow. This means that anyone who is working on work items that are in other steps should try to refocus on the tasks/work items that are stopped at the bottleneck.

Step 4 – Reduce WIP limit at preceding step

To streamline the overall flow of value, the bottleneck needs to be reduced. There are multiple ways to accomplish this. The initial change that the team should consider is to control the flow of work into this step. Similar to a freeway onramp that is throttled by a stoplight to prevent too many automobiles from entering a crowded area of the highway, controlling the amount of work entering a specific process step will help the work within this step move through to the end more quickly.

Step 5 – Increase WIP limit at bottleneck

It is possible that the reason that work has accumulated at the bottleneck is a result of not having sufficient WIP. Hence, the team may wish to increase this WIP to prevent this bottleneck from resurfacing in the future. This is generally done after the accumulation of work at bottleneck has been resolved.

A word of caution: By increasing WIP limit at this point, there's a possibility that this may will increase the risk of creating a new bottleneck at another step downstream of the process. The team is encouraged to explore this prior to making a decision to increase WIP at a specific process step.

Step 6 – Increase the number of process steps

Depending on the number of steps the team decided to implement for the workflow, it may be advantageous to eliminate or add process steps to the flow. As discussed previously, the higher number of steps there are, the more administrative overhead is required to maintain the flow. However, this will likely provide increased level of visibility that may help the team determine the "real" bottleneck.

One important note for the team to consider is to make only one change at a time when working to maximize the overall efficiency of the process. For example, only change the WIP limit for one process step, then measure the average cycle time to evaluate the impact of that change. If more than one change is implemented between metrics are taken, it will be much more difficult to identify which change actually led to the change in throughput.

	Request Accepted	Analysis	Implementation	Review	Delivered
Scenario 1: Efficient Flow	[N/A] {10}	[5] {2}	[5] {3}	[5] {2}	[N/A] {N/A}
Scenario 2: Bottleneck at "Analysis"	[N/A] {10}	[5] {5}	[5] {1}	[5] {2}	[N/A] {N/A}
Scenarior 3: Bottleneck at "Implementation"	[N/A] {10}	[5] {3}	[5] {5}	[5] {1}	[N/A] {N/A}
Scenario 4: 2 Bottlenecks at "Implementation & Review"	[N/A] {10}	[5] {4}	[5] {5}	[5] {5}	[N/A] {N/A}

Table 2: Optimizing the Flow of Value

Legend: [WIP LIMIT]. {# Active Requests}

Table 2 shows four examples of likely scenarios for a typical Kanban flow. The data from the board allows teams to evaluate the effectiveness of the flow which enables further optimization through an iterative approach. For simplicity, this example assumes there are no WIP limits for the first state of the flow ("Request Accepted"). In real-world application, there may be situations where it is beneficial to impose a WIP limit for the initial state of the flow.

Scenario 1: Efficient Flow

This is an example of an efficient flow in which none of the workflow states are fully-utilized; this condition is ideal in that it provides a smooth flow of work through the system. There's a general misconception that 100% utilization is a positive attribute of a team or an individual contributor. Many studies have demonstrated that having some type of buffer actually facilitates flow more efficiently than having zero buffer/slack. In order to maximize the flow of work, effective management of the queue is essential; if the queue (i.e. backlog of work items) builds up excessively, the flow will be disrupted and wait times will increase, resulting in increased Lead Time.

Scenario 2: Bottleneck at "Analysis"

The amount of in-progress work has reached its maximum limit at the "Analysis" step of the process. This means that no more work can enter this state, so a bottleneck has been formed. In addition, the amount of work that is currently in the "Implementation" step is less than optimal (only at 20%). This is a clear indicator of an inefficient flow. The team may consider focusing their resources and attention to execute the tasks within the "Analysis" step in order to clear the path for downstream activities.

Scenario 3: Bottleneck at "Implementation"

Similar to Scenario 2, the mount of work is aggregated at one point of the flow, and must be addressed in order to alleviate the bottleneck.

Scenario 4: Two Bottlenecks at "Implementation" and "Review"

This is the most complex scenario within the list of examples, given that the flow has two distinct bottlenecks. The team may choose to address either of the two bottlenecks one at a time. It may be advantageous to address the latter step in the flow in order to move those to closure and deliver them to the customer. To accomplish this, the team may choose to halt work on some of the items within the "Implementation" step in order to swarm around the items within the "Review" step.

In conclusion, implementing an effective Kanban team is not an exact science; some experimentation and refinement is expected for teams that choose to adopt this model. Provided that the team applies a methodical approach to adapt to new learnings, the delivery of value will be optimized quickly.



About the Author

Mr. Eugene Lai is a seasoned innovator in Information Technology and process engineering with over 20 years of proven track record delivering high-impact solutions within PMOs and Agile teams. In previous roles as Lead Software Engineer, Chief Scrum Master, Principal Program Manager, VP of Project Management, PMO Director, Technical Consultant and Agile Advisor, Mr. Lai has successfully applied process lifecycle frameworks and delivered complex solutions to various Fortune 500 financial services companies and U.S. Department of Defense customers across the globe.

About Cprime

Cprime is an industry-leading, full-service global consulting firm with a focus on providing integrated and innovative solutions around digital transformation, product, cloud, and technology. With over 20 years' experience, we provide strategic and technical expertise to businesses across more than 50 industries. Our team of advisors and technical experts have the know-how to meet organizations where they are to develop actionable solutions and solve business challenges. We also collaborate with our expansive network of partners to design, deploy, and harmonize technology stacks across organizations. Our mission is to empower visionary business leaders and teams to reimagine the future of work to achieve better outcomes.

Visit us at **www.cprime.com**