

The Kanban Guide for Scrum Teams

April 2018



Developed and sustained by Scrum.org and Daniel Vacanti

Table of Contents

Purpose	3
Relation to the Scrum Guide	3
Definition of Kanban	3
Kanban with Scrum Theory	3
Definition of Workflow	4
Service Level Expectation	5
Kanban Practices	5
Visualization of the Workflow - the Kanban Board	5
Limiting WIP	5
Active Management of Work Items in Progress	6
Inspect and Adapt Workflow	6
Flow Metrics and Analytics	6
The Basic Metrics of Flow	7
Flow-Based Events	7
The Sprint	7
Flow-Based Sprint Planning	7
Flow-Based Daily Scrums	8
Flow-Based Sprint Review	8
Flow-Based Sprint Retrospectives	8
Endnote	8
History and Acknowledgments	9

© 2018 Scrum.org. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this *Kanban Guide for Scrum Teams*, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

Purpose

The flow-based perspective of Kanban can enhance and complement the Scrum framework and its implementation. Teams can apply Kanban whether they are just starting to use Scrum or have been using it all along.

The Kanban Guide for Scrum Teams is the result of a collaboration between members of the Scrum.org community and leaders of the Kanban community. Together, they stand behind *The Kanban Guide for Scrum Teams*. It is their shared belief that professional software practitioners can benefit from the application of Kanban together with Scrum.

Relation to the Scrum Guide

This guide is not meant to replace or discount any part of *The Scrum Guide*. It is designed to enhance and expand the practices of the Scrum framework. This guide assumes the reader is operating a process using the Scrum framework. Therefore, *The Scrum Guide* applies in its entirety.

Definition of Kanban

Kanban (n): a strategy for optimizing the flow of stakeholder value through a process that uses a visual, work-in-progress limited pull system.

Central to the definition of Kanban is the concept of "flow." Flow is the movement of customer value throughout the product development system. Kanban optimizes flow by improving the overall efficiency, effectiveness, and predictability of a process.

Kanban with Scrum Theory

First, a quick review of a key tenet of *The Scrum Guide*:

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

Scrum mandates that the Sprint Backlog be transparent, but it provides limited guidance on how to accomplish this. Nor does it define how to achieve explicit transparency to the flow of work into the Product Backlog, from the Product Backlog into the Sprint Backlog, and whatever happens to the work after it makes it into a "Done" increment. This is where Kanban can help. By visualizing work in new ways, a Scrum Team can apply the set of practices laid out in this guide to more effectively optimize value delivery. These practices borrow from and build upon the principles of lean thinking, product development flow, and queuing theory.

© 2018 Scrum.org. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this *Kanban Guide for Scrum Teams*, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

A beneficial side effect of optimizing value delivery is that it provides many more opportunities to inspect and adapt process and product. Tightening the customer feedback loop with Kanban is a proven strategy for empirically improving a process.

Kanban's hyperfocus on transparency, visualization, and flow, combined with the Scrum framework, form a powerful foundation on which to design a process to deliver optimal customer value.

Definition of Workflow

Optimizing flow requires defining what flow means in a Scrum context. Each Scrum Team must create its definition of "Workflow" containing the following elements:

- Defined points at which the Scrum Team considers work to have started and to have finished.
- A definition of the individual units of customer value that are flowing through the Scrum Team's system (most likely Product Backlog Items (PBIs)).
- A definition of the workflow states that the PBIs flow through from start to finish (of which there must be at least one active state).
- Explicit policies about how work flows through each state (which may include items from a Scrum Team's definition of "Done" and pull policies between stages).
- A definition of how Work in Progress (WIP) will be limited.
- A set Service Level Expectation (SLE) that communicates a forecast of how long it should take to complete work items.

While it is up to the Scrum Team to define its workflow, there are a few elements it must include:

- Identifying work items that are not yet in an active state as "not started."
- Identifying work items entering the active state ("started") as Work in Progress (WIP).
- Identifying work items that have passed through all of the active states planned for that item as "finished."

In summary, the definition of "Workflow" includes a shared understanding within the Scrum Team of how work is defined (work items), the start state of the process, the active states for the work items, and the finished state of the process.

Note that the states in the definition of "Workflow" may not coincide with the states as defined by a Sprint Backlog. For instance, a Scrum Team's definition of "Workflow" may include states that are upstream, downstream, inside, or outside of the Sprint Backlog. Similarly, the work items moving through the workflow may not correspond to Product Backlog Items or other parts of the Sprint Backlog or Scrum. Finally, a particular work item might not flow through all of the active states, and a work item might not even flow sequentially through the active states.

The creation and adaptation of the definition of "Workflow" may impact or be impacted by existing artifacts. The Product Owner's and Development Team's accountabilities and responsibilities over those artifacts remain as described in the Scrum Guide.

Service Level Expectation

An SLE forecasts how long it should take a given item to flow from start to finish within your workflow. The SLE itself has two parts: a period of elapsed days and a probability associated with that period (e.g., "85% of work items will be finished in eight days or less"). The SLE is based on a Scrum Team's historical cycle time, and once calculated, should be posted on the Kanban board. If no historical cycle time data exists, the Scrum Team should make its best guess and then replace that guess once there is enough historical data to do a proper SLE calculation.

Regardless of how a Scrum Team chooses to craft its definition of "Workflow", it is the central concept of this guide. All other elements of this guide depend heavily on exactly how a Scrum Team specifies the definition of "Workflow." It can and should change as Scrum Team's empirically discover better ways of flowing work. Similarly, the consistent use of the definition of "Workflow" is necessary when used in conjunction with all of the other elements in this guide (e.g., flow metrics). Consistency ensures the integrity of Kanban's hyperfocus on transparency.

Kanban Practices

Scrum Teams achieve flow optimization by using the following four practices:

- Visualization of the workflow
- Limiting WIP
- Active management of work items in progress
- Inspecting and adapting their definition of "Workflow"

Visualization of the Workflow - the Kanban Board

Visualization using the Kanban board is the way the Scrum Team makes its workflow transparent. The board's presentation should prompt the right conversations at the right time and proactively suggest opportunities for improvement.

Limiting WIP

Work in Progress (WIP) refers to the work items the Scrum Team has started but has not yet finished. Scrum Teams using Kanban must explicitly control these in-progress work items from the time they consider them "started" until the time they consider them "finished." That control is usually represented as a number or numbers on a Kanban board. Those numbers are called "WIP Limits." A WIP Limit can include work items in a single column, several grouped columns, or a whole board. Once the Scrum Team has established a WIP Limit, it refrains from pulling more than that number of work items into a given part of the workflow. The Scrum Team controls what the limits are and how it will apply them.

The primary side effect of limiting WIP is that it creates a "pull system." It is called a pull system because the Scrum Team starts work on an item (i.e. pulls) only when there is a clear signal that it is time to do so (note this is different from a "push" system, which demands that work start on

an item whenever it is requested). When the WIP drops below a defined limit, that is the signal to start new work.

The Sprint is itself a form of limiting WIP. By definition, a Sprint is a way of controlling how much work a Development Team is going to attempt during a specified period. Work is only pulled into a Sprint Backlog when the Development Team chooses to do so (usually, but not always, during Sprint Planning). Whether intended or not, Scrum has embraced this fundamental practice of flow from its beginning. Kanban's more granular and explicit WIP Limit not only helps workflow but can improve the Scrum Team's focus, commitment, and collaboration even further.

Active Management of Work Items in Progress

Limiting WIP is a necessary component to achieve flow, but it alone is not sufficient. The third practice to establish flow is the active management of work items in progress. Active management can take several forms, including but not limited to the following:

- Responding quickly to blocked work items.
- Making sure that work items are only pulled into the workflow at about the same rate that they leave the workflow.
- Ensuring work items aren't left to age unnecessarily and are completed according to an established SLE.
- Unclogging work that piles up in a column or columns.

While many of these topics may be covered during the Daily Scrum (see the *Flow-Based Daily Scrum* section below) it is the Scrum Team's responsibility to ensure the continuous proactive, active, and reactive management of work items in progress.

Inspect and Adapt Workflow

Think of policies as the "rules of the game" for the Scrum Team's "Workflow". Small changes to those policies can have a material impact on how the Scrum Team performs overall. *The Scrum Guide* provides a minimum set of explicit policies as well as instructions for how to figure out some context-specific policies (e.g. Scrum Team's definition of "Done"). When applying Kanban, *The Scrum Team* will want to supplement *The Scrum Guide* with more explicit policies for its process. Those policies will be captured in a Scrum Team's definition of "Workflow".

Explicit means that these policies are written down or visualized somehow and that the whole Scrum Team understands them. The Kanban board should display all relevant policies or direct Scrum Team members where to find them.

A good example of a policy that needs to be explicit is how the Scrum Team defines the moment when items go from unstarted to started and from in-progress to finished. No doubt the Scrum Team will add other policies on top of the specific ones called for by this guide.

Flow Metrics and Analytics

The application of Kanban within a Scrum context requires the collection and analysis of a minimum set of flow metrics. These metrics are necessary for the practice of active management

© 2018 Scrum.org. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this *Kanban Guide for Scrum Teams*, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

of work items in progress. They also make the flow transparent and enable flow-oriented inspection and adaptation. These metrics are a reflection of the current health and performance of the Scrum Team's approach. They will also point to interventions that can improve Scrum Team function and the value it delivers.

The Basic Metrics of Flow

The four basic metrics of flow that Scrum Teams using Kanban will need to track are as follows:

- Work in Progress (WIP): The number of work items started but not finished (according to the Scrum Team's definition of "Workflow").
- Cycle Time: The amount of elapsed time between when a work item "starts" and when a work item "finishes."
- Work Item Age: The amount of elapsed time between when a work item "started" and the current time.
- Throughput: The number of work items "finished" per unit of time. Note the measurement of throughput is the exact count of work items.

Calculating cycle times and work item age requires the Scrum Team to (at a minimum) track the start date and finished date of each item.

These metrics should be monitored throughout the Sprint – specifically in the Scrum Events (see the *Flow-Based Events* section below). As always, there are other flow metrics that the Scrum Team may want to examine, but these are the minimum requirement.

Flow-Based Events

Kanban in a Scrum context does not require any additional events to those outlined in *The Scrum Guide*. However, using a flow-based perspective can enhance Scrum events.

The Sprint

A Sprint is a cadence or a regular "heartbeat" for inspection and adaptation. The Sprint's regular cycle and elements are naturally suited to managing flow. A Sprint contains the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective. The events in a Sprint can work as a feedback loop for inspecting Kanban flow metrics, and the Kanban approach can also act as a feedback loop for inspecting the Scrum implementation.

Flow-Based Sprint Planning

A flow-based Sprint Planning meeting uses flow metrics as an aid for developing the Sprint Backlog. For example, using historical Throughput to understand the Scrum Team's capacity for the next Sprint. A Scrum Team's SLE might influence the work planned for the first days of the Sprint.

Flow-Based Daily Scrums

A flow-based Daily Scrum focuses on ensuring the Scrum Team is doing everything it can to maintain flow every day. While the goal of the Daily Scrum remains the same as in *The Scrum Guide*, the meeting itself takes place around the Kanban board and focuses on where flow is lacking and on what actions the Scrum Team can take to get work flowing again.

Some additional things to consider during a flow-based Daily Scrum are as follows:

- What work items are blocked and what can the Scrum Team do to get them unblocked?
- What is the Work Item Age of each item in progress? What work items have violated or are about to violate their SLE and what can the Scrum Team do to get that work completed?
- Are there any things that may impact the Scrum Team's ability to complete work today that is not represented on the board?

Flow-Based Sprint Review

The Scrum Guide provides a detailed outline of the Sprint Review process. In addition to these activities, inspecting Kanban flow metrics as part of the Sprint Review creates opportunities for new conversations about monitoring progress towards a goal. Reviewing Throughput might provide some additional information when the Product Owner discusses likely target and delivery dates. Reviewing a Scrum Team's SLE may cause the Product Owner to revise the Product Backlog.

Flow-Based Sprint Retrospectives

A flow-based Sprint Retrospective adds the inspection of flow metrics and analytics to help determine what improvements the Scrum Team can make to its processes, including the Sprint Retrospective itself. The Scrum Team using Kanban also inspects and adapts the definition of "Workflow" to optimize the flow in the next Sprint. Using a cumulative flow diagram to visualize a Scrum Team's Work in Progress, average approximate Cycle Time and average Throughput may be valuable.

The Scrum Guide dictates that the Sprint Retrospective take place after the Sprint Review and before the next Sprint Planning. This does not change when using Kanban. However, flow-based retrospective opportunities need not coincide within the boundaries of a Sprint. They can occur "just in time". Correspondingly, changes to a team's definition of "Workflow" may happen at any time, however, as these changes will have a material impact on how the Scrum Team performs, changes made during the regular cadence provided by the Sprint Retrospective event will reduce complexity and improve transparency.

Endnote

Scrum is not a process or technique. It is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. As *The Scrum Guide* points out, it functions well as a container for other techniques, methodologies, and practices.

© 2018 Scrum.org. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this *Kanban Guide for Scrum Teams*, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

The flow optimization practices of Kanban provide Scrum Teams with additional opportunities to inspect the right thing, at the right time, and then based on that inspection, adapt as needed. Kanban's hyperfocus on transparency, visualization, and flow maximizes feedback, empiricism, and ultimately the delivery of customer value.

History and Acknowledgments

The set of practices commonly referred to as Kanban mostly originated on a team at Corbis in 2006. Those practices quickly spread to encompass a large and diverse international community who over the years continued to enhance and evolve the approach.

This guide was developed collaboratively by Scrum.org, our Professional Scrum Trainer Community, Steve Porter, Yuval Yeret, and Daniel Vacanti.

A special thank you to Louis-Philippe Carignan and Charles Bradley for their contributions in this effort. We also owe a debt of gratitude to all those practitioners who have in the past contributed to make Kanban a viable and successful lean-agile strategy.

© 2018 Scrum.org. Offered for license under the Attribution Share-Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this *Kanban Guide for Scrum Teams*, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.