

Table of Contents

| | |
|--|----------|
| What's The Difference Between Kanban and Scrum? | 3 |
| How Would You Describe Scrum in a Nutshell? | 3 |
| How Would You Describe Kanban in a Nutshell? | 3 |
| What Are the Similarities Between Kanban and Scrum? | 4 |
| What Are the Differences Between Scrum and Kanban? | 4 |
| Which is Best for My Team? Kanban, Scrum or ...? | 5 |
| Want to Know More? | 7 |

What's The Difference Between Kanban and Scrum?

Or “When Should We Use Kanban Instead of Scrum?”

Or “When Should We Use Scrum Instead of Kanban?”

The two most popular Team practices are Scrum and Kanban. People often wonder whether one is better than the other, or whether one should be applied over another. This page will help you think through the ideas.

How Would You Describe Scrum in a Nutshell?

- Split your organization
 - Small, cross-functional, self-organizing teams
 - Split your work
 - Small, concrete deliverables
 - Assign someone to be responsible for the work item list and to sort the list by priority
 - Assign someone to be responsible for coaching and facilitating the team
 - Implementation team estimates the relative size of each work item
- Split time
 - Short fixed-length iterations (Usually 2 Weeks long, but Between 1 - 4 weeks)
 - Potentially shippable solutions demonstrated after each iteration
- Optimize the release plan
 - Update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration
- Optimize the process
 - Retrospective after each iteration

How Would You Describe Kanban in a Nutshell?

- Visualize the workflow
 - Split the work into pieces, write each item on a card and put on the wall
 - Use named columns to illustrate where each item is in the workflow
- Limit WIP (work in progress)
 - Assign explicit limits to how many items may be in progress at each workflow state
- Measure the lead time

- Lead Time (sometimes called “cycle time”) = average time to complete one item
- Optimize the process to make lead time as small and predictable as possible

As you can see, Kanban is very lightweight and requires a lot of (self-)discipline to implement on a team. The specific practices that Kanban teams might use include:

- Kanban board shows all work of the team
- Kanban board shows flow of work of the team so we can find and work bottlenecks
- Different service levels (classes of service) for distinct types of work (e.g., expedite, fixed date, standard) are identified and supported
- WIP limits in place and enforced
- Kanban boards are processed “Right to left, top to bottom”
- Continuous improvement occurring
- Cycle time and throughput tracked, and improvements focused on improving these numbers

What Are the Similarities Between Kanban and Scrum?

- Both are Lean and Agile
- Both use pull scheduling
- Both limit Work-in-progress (WIP)
- Both use transparency to drive process improvement
- Both focus on delivering releasable solutions early and often
- Both are based on self-organizing (stable) teams
- Both require breaking the work into pieces
- Both optimize the release plan continuously based on empirical data

What Are the Differences Between Scrum and Kanban?

For folks that are into the details of these two methods. They clearly come at the problem of managing knowledge work from different perspectives. The following table provides a clear breakdown of these differences:

| Scrum | Kanban |
|---|---|
| Time-boxed iterations prescribed | Time-boxed iterations optional. * Can have separate cadences for planning, release, and process improvement. * Can be event driven instead of time-boxed. |
| Team commits to a specific amount of work for this iteration (Sprint) | Team commits to work when it is brought to the board for execution |
| Uses velocity as default metric for planning and process improvements | Uses lead time as default metric for planning and process improvements |
| Cross functional teams prescribed | Cross functional teams optional * Specialist teams allowed |
| Items must be broken down so they can be completed in an iteration | No item size is prescribed although small(-er) sized work is highly recommended |
| Burn down chart is prescribed | No diagram type is prescribed |
| "Change agent" is "commitment" | "Change agent" is "WIP limits" |
| WIP limited indirectly * Per iteration | WIP limited directly * Per workflow state |
| Estimation prescribed | Estimation optional |
| Cannot add items to ongoing iteration | Can add items whenever capacity is available |
| Prescribe 3 roles * Product Owner * Scrum Master * Team | No prescribed roles |
| Prescribes 4 events * Planning * Daily Scrum * Review * Retrospective | No prescribed events |
| Scrum board is reset between each iteration | Kanban board is persistent |
| Prioritized backlog is prescribed | Prioritization is optional |

Which is Best for My Team? Kanban, Scrum or ...?

If you treat Kanban and Scrum as if they are competing approaches, or adopt a textbook approach to their implementation, then the advice to choose one approach over the other usually breaks down as follows:

| Question | Kanban | Scrum |
|--|--------|-------|
| Primary Consideration | | |
| * Work for the team is more than 50% demand driven (team's priority is responsiveness) | X | |
| * Work for the team is most project driven (team's responsibility is predictability, forecasting and productivity) | | X |

| | | |
|--|---|---|
| Secondary Consideration | | |
| * Team questions value of estimation and planning | X | |
| * Team struggles to break items into small pieces | | X |
| * No or low appetite for significant process change | X | |
| * Some or high appetite for significant process change | | X |
| * Team members have significant self-discipline | X | |
| * Team members have limited self-discipline | | X |

Some people feel like Kanban is easier to implement and so opt for that. The reality is that the **WRONG** reason to implement Scrum is because the Team is simply unwilling to do Scrum-like events or find that they cannot finish a Story in one Iteration.

A notes on this criterion. Many teams whose work is the result of a ticketing system assume that all their work is demand driven, and that Kanban is therefore their only option. This is a problem in that if the team can plan work, they usually are also able to deliver better and faster over time.

Experience has shown that work coming from the ticketing system does not mean that we must treat every work item as interrupt-driven and therefore you are unable to plan it. For example, teams that are pure production support have different classes of incoming problems and so can vary how quickly you would need to respond to that problem. A production site is down request is going to need a more urgent response than say a minor defect because of a workflow problem. This gives as an option to plan more items and would mean that Scrum practices are useful. For example, if we can suggest to the customer “that they wait for (on average for a 2 week Iteration) 1 week for us to work this defect” then we can treat this as an input into the next planned Sprint (iteration) in a Scrum implementation.

In some instances up to 65% of the incoming work for a Team can be treated this way.

In some ways the discussion about whether you use EITHER Scrum OR Kanban is misleading. Like all agile approaches the best idea is to leverage Kanban and Scrum to improve how you deliver value. Scrum teams can benefit from ideas like understanding the workflow, WIP limits, classes of service. Kanban teams benefit from an understanding of the purpose of Scrum events, roles, and artifacts and incorporate some or all of them into their practice.

Here are some more specific approaches people have tried:

- Team Examples
 - Kanban Teams are set up as cross-functional
 - Scrum Teams set WIP Limits within a Sprint
 - Kanban Teams Setup cadence of retrospectives, daily meetings, etc
 - Scrum Teams establish (capacity-based) rules for Break-fix or other “interruptions”
- Team-of-Teams (Scaling) Examples
 - Kanban Teams commit to work on Sprint Schedule to aid in synchronization and Alignment
 - Portfolio and Program level work is set up as Kanban
- Retrospectives drive improvement
 - Scrum and Kanban as a toolbox of ideas to Leverage
 - Use whatever helps you improve

Want to Know More?

- [Should We Do WIP Limits Even Though We Are Doing Scrum?](#)
- ["Kanban vs Scrum" by Henrik Kniberg](#) - The source of a lot of the ideas in the presentation. As always, Henrik is brilliant.
- [Scrum and XP from the Trenches](#) by Henrik Kniberg
- [Kanban Flow and Cadence](#) by Karl Scotland
 - Comparison of Kanban and Scrum
- I have seen this type of deck in other places, but cannot find it easily so I've created something that can be used to help people understand. This page is the "Wiki-fied" version of this content.
- Series of posts showing how Scrum and Kanban can effectively work together to achieve even better results than one method alone:
 - <https://www.scrum.org/resources/blog/scrum-and-kanban-stronger-together>
 - How Scrum teams can benefit from Kanban practices:
<https://www.scrum.org/resources/blog/scrum-primer-kanban-teams>
 - How Kanban teams can benefit from Scrum practices:
<https://www.scrum.org/resources/blog/kanban-primer-scrum-teams>
- Introduction to Kanban cadences: <http://blog.kanbanery.com/the-seven-kanban-cadences/>

[FAQ](#), [Kanban](#), [Scrum](#), [comparison](#), [agile](#), [SubjectSpecificArticle](#)

From:

<https://www.hanssamios.com/dokuwiki/> - Hans Samios' Personal Lean-Agile Knowledge Base

Permanent link:

https://www.hanssamios.com/dokuwiki/whats_the_difference_between_kanban_and_scrum_ssa?rev=1678454955

Last update: **2023/03/10 05:29**

