

SCRUM VS KANBAN: A COMPARISON OF AGILE METHODOLOGIES



Companies must be agile. Companies must deliver digital services quickly; they must align with customers' every-changing needs. [Lean startups](#) are all the rage in the age of [VUCA](#). Pivoting in response to consumer trends can no longer wait for monolithic and bureaucratic change procedures. Practitioners in the tech field have prioritized two approaches, Scrum and Kanban, in pursuit of this agility.

Let's look at the main differences between Scrum and Kanban. Note that they are not in opposition—it's not one or the other. In fact, you may get the most benefit when using them in tandem.

Scrum and Kanban

A comparison of Agile methodologies

	Scrum	Kanban
Origin	Software development	Lean manufacturing
Ideology	Solve complex problems while delivering valuable products	Use visuals to improve work flows and processes
Practices	Sprint planning Sprint Daily scrum Sprint review Sprint retrospective	Visualize the flow of work Limit work in progress Manage flow Make process policies explicit Implement feedback loops Improve, experiment
Roles	Product Owner Scrum Master Development Team	No formal roles
Metrics	Velocity	Cycle time Throughput

Origin stories

The way Scrum and Kanban originated highlights their differences.

Scrum

Scrum originated with a January 1986 [HBR](#) paper, The New New Product Development Game, written by Hirotaka Takeuchi and Ikujiro Nonaka. The main premise was to move away from a sequential approach to new product development. Instead, developers should embrace a holistic, fast, flexible process. (Takeuchi and Nonaka borrowed from the game of rugby, hence the name Scrum.)

Ken Schwaber and Jeff Sutherland embraced these ideas, presenting Scrum and [its applicability to software development](#) at the 1995 Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) conference. In 2001, Schwaber and Sutherland participated in the famous ski resort meeting where the [Agile Manifesto](#) was crafted. The two later authored the [Scrum Guide](#), in 2010, which is recognized today as the official Scrum Body of Knowledge.

Kanban

Kanban originates decades earlier, when Japanese shop owners used sign boards in crowded streets to advertise their wares and differentiate them from competitors. The name Kanban comes from two [Japanese](#) names, Kan meaning 'sign' and Ban meaning a 'board'.

In 1956, a young Toyota industrial engineer Taiichi Ohno created a system that used paper cards for

signaling and tracking demand in his factory, naming the new system Kanban. Benefits including reduced stockpiles, improved throughput, and provided high visibility into the process propelled this approach to success. The system was incorporated into the entire organization in 1963 and became part of the [Toyota Production System](#).

In 2004, [David J. Anderson](#) was the first to apply Kanban to IT, software development, and knowledge work.

Ideologies

People use the Scrum framework to address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is founded on empirical process control theory, which asserts that knowledge comes from experience and making decisions based on what is known. Three pillars uphold this approach:

- **Transparency.** Significant aspects of the process must be visible for those involved in the outcomes.
- **Inspection.** Those involved must frequently inspect the artifacts and progress towards the goal.
- **Adaptation.** If any aspect of the artifacts or progress is unsatisfactory, adjustments must be made as soon as possible.

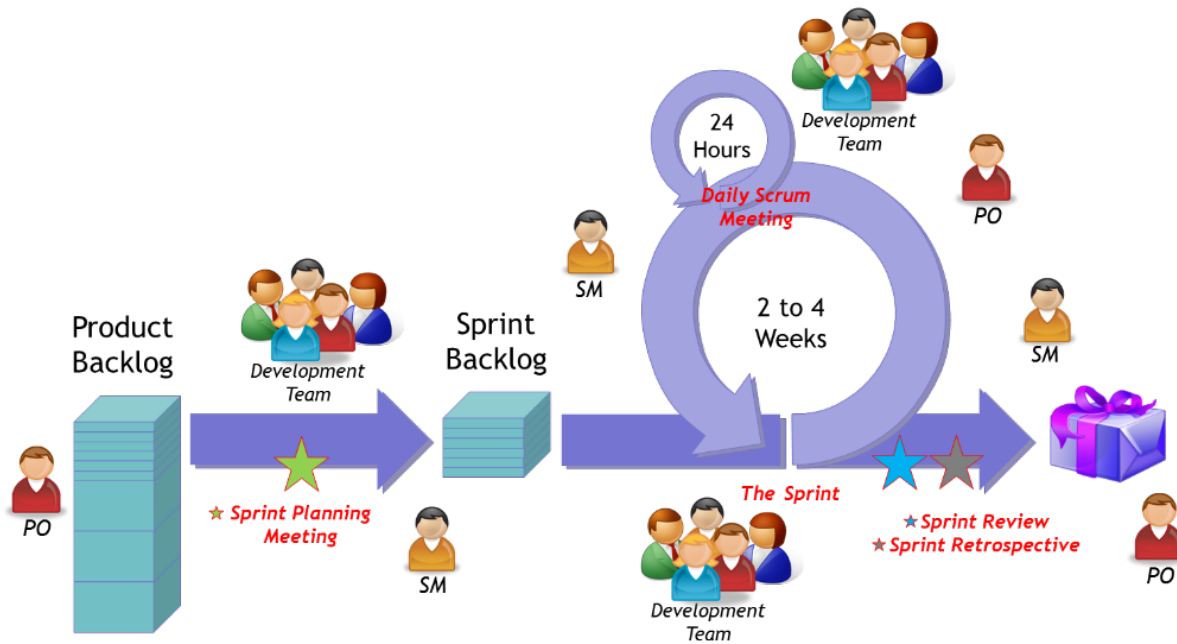
Kanban is a way to improve flow and provoke system improvement through visualization and controlling work in progress. It has four foundational principles:

- Start with what you do now.
- Agree to pursue evolutionary change.
- Initially, respect current roles, responsibilities, and job titles.
- Encourage acts of leadership at all levels.

Practices

Scrum denotes five time-based events for managing product delivery iteratively and incrementally, while maximizing opportunities for feedback. These are:

1. **Sprint Planning.** An eight-hour session where the team decides what to deliver in the coming sprint (from the product backlog) and how to go about it.
2. **Sprint.** A timeframe of a month or less where the team delivers what was agreed in the sprint planning session.
3. **Daily Scrum.** A 15-minute timebox (commonly referred to as daily stand up) where the team meets daily during the sprint to inspect progress and identify blockers.
4. **Sprint Review.** A four-hour timebox event held at the end of the sprint. The team demonstrates the product/changes to customers and gathers feedback on what to incorporate in the product backlog for delivery in subsequent sprints.
5. **Sprint Retrospective.** A three-hour timebox event held after the sprint review (and before the next sprint planning). The team reviews their work, identifying opportunities for improvement work processes in subsequent sprints.

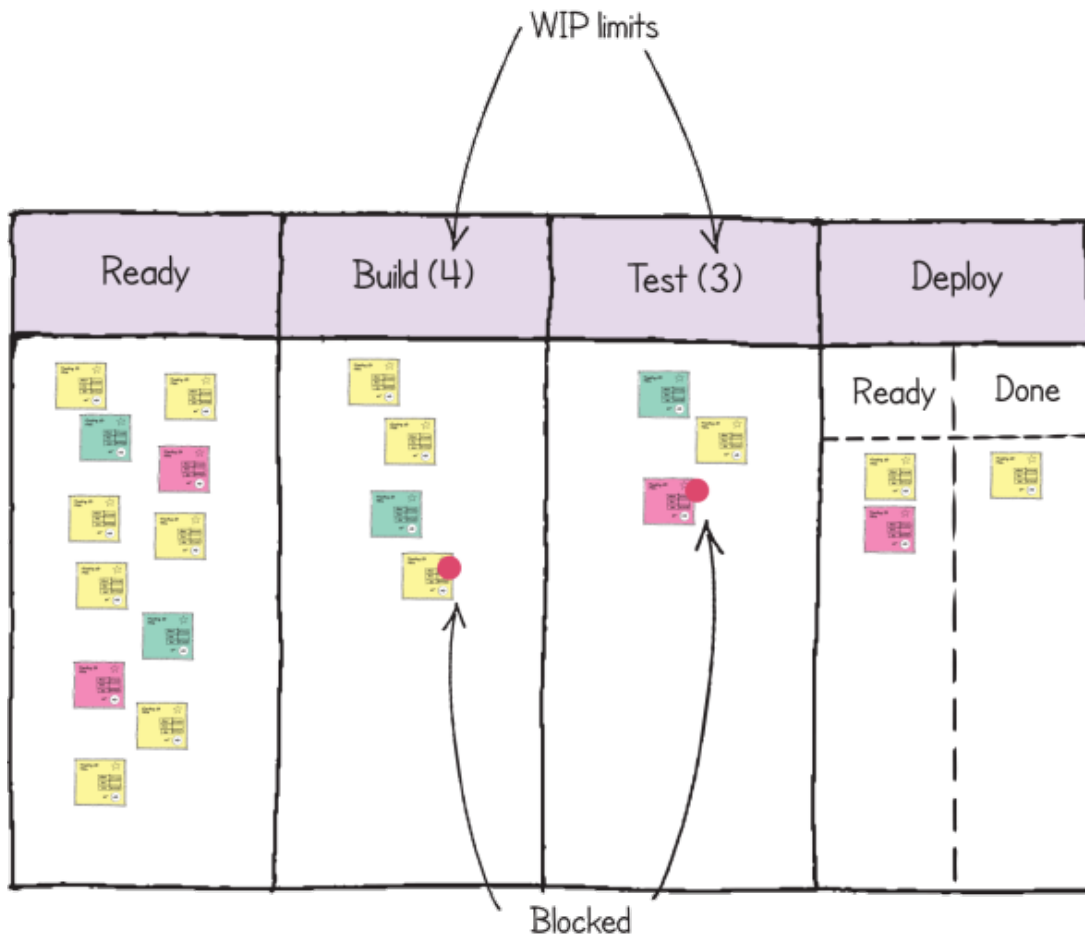


Scrum framework

(Source)

Kanban has six core practices:

1. **Visualize the Flow of Work.** Use cards or software to visualize the process activities on swim lanes.
2. **Limit Work in Progress (WIP).** Encourage your team to complete work at hand first before taking up new work. The team pulls in new work only when they have capacity to handle it.
3. **Manage Flow.** Observe the work as it flows through the swim lanes. Address any bottlenecks.
4. **Make Process Policies Explicit.** Visually diagram the process rules and guidelines for managing the flow of work.
5. **Implement Feedback Loops.** Throughout the work process, incorporate regular reviews with the team and customers to gather and incorporate feedback.
6. **Improve Collaboratively, Evolve Experimentally.** As a team, look for and incorporate improvement initiatives, including through safe-to-fail experiments.



Kanban Board ([Source](#))

Roles

Scrum defines three main roles:

- **Product Owner.** The sole person responsible for managing the Product Backlog.
- **Scrum Master.** A servant leader responsible for helping the team understand Scrum theory, practices, rules, and values.
- **Development Team.** Self-organizing group of 3-9 professionals responsible for delivering the products.

Kanban has no set roles. However, [David Anderson](#) advocates for two roles:

- **Service Delivery Manager (SDM).** The person who ensures work items flow and facilitates change and continuous improvement activities.
- **Service Request Manager (SRM).** The person who orders and prioritizes work items and improves corporate governance with the process.

Metrics

In Scrum, the main production metric is **velocity**. This describes the rate of progress of what the team is delivering, based on the estimation carried out during sprint planning. The scrum team uses velocity as an indicator of how much they are delivering the items in the product backlog at each sprint, based on the effort required.

Kanban uses two main metrics:

- **Cycle time** measures how much time a task spends going through the process (i.e. how long a card stays within the WIP swim lanes).
- **Throughput** measures the total amount of work delivered in a certain time period (i.e. number of cards delivered in each time period on a specific Kanban board).

Using Scrum and Kanban together

Scrum and Kanban can be used together, both in development environments and IT service management. In fact, the [Scrumban framework](#) has emerged. Scrumban leverages both frameworks to better embrace agility and to improve what is lacking in each. Using both frameworks provides many benefits, particularly from a people perspective: fostering collaboration and improvement through feedback and focusing attention on delivering business value.

Additional resources

To learn more about Agile and software development, check out these BMC Blogs:

- [Intro to Agile with Scrum: 4 Tips for Getting Started](#)
- [Agile Roles and Responsibilities](#)
- [Customer User Feedback: The Keystone of the Agile Approach](#)
- [The Software Development Lifecycle \(SDLC\): An Introduction](#)
- [DevOps Feedback Loops: An Introduction](#)