When Kanban is the Better Choice

by Brendan Wovchko • 37 Comments

When Kanban is the Better Choice

Scrum's great. And I'm very closely associated with Scrum, since I co-founded the Scrum Alliance, co-taught the very first Certified Scrum Master course, and still teach many Certified Scrum Master and Certified Scrum Product Owner courses.

But, as great as I think Scrum is, I don't think Scrum is right for every team in every situation.

In some situations, other approaches are called for. Usually it will still be some variant of an agile approach. Most commonly that means Kanban.

And so, I'm writing to introduce a guest post on Kanban.

This post is from Brendan Wovchko, whom I'm happy to call a friend. Brendan is an expert at both Kanban and Scrum. He and I have co-trained together, and I always appreciate his openness to considering each team's unique situation when advising them.

He knows that Scrum and Kanban each can be best for different situations. And in the following post he's going to share reasons to favor Kanban in some situations.

--Mike

I'm frequently asked by my workshop participants to weigh-in on their team's debate between Scrum and Kanban. I always find their questions interesting because their assumption is that Scrum and Kanban are somehow enemies.

As an agilist, I value experimentation over prescription. I don't want someone to tell me what's best for a specific team, I want to prove it.

A truly agile team will experiment with a lot of ideas to find which works best, not just adopt the first thing they try. Scrum and Kanban aren't competitors, they are experiments every team should try.

While that is true, I recognize that many teams have limited liberty to experiment. If you work for an organization that won't tolerate small failures in order to discover big productivity gains, I understand! If that's you, I'm willing to bend my rule about prescription and offer some advice on five conditions under which I've found Kanban to be a better fit than Scrum.

1. Low Tolerance for Change

Over the years I've encountered many organizations that are brittle to change. They're comfortable with the way they do things even if they aren't getting results. Their solution isn't to rethink how they work, it's usually to make their teams work longer and harder—which isn't a real solution. Scrum is a threat to brittle, change-adverse organizations primarily because of how quickly it transforms roles and meetings.

Kanban doesn't use transformative change, it embraces evolutionary change. Kanban employs a start with what you do now mindset that introduces teams to the shallow end of the pool before taking them toward the deep end of maturity. Kanban doesn't require any changes to roles or meetings when first getting started.

2. Obvious at All Levels

The primary advantage Kanban has over Scrum is that it is immediately intuitive to anyone. A Kanban board is an instant sense-making device. It requires zero explanation to understand. However, Kanban's primary advantage is also its biggest flaw. It's easy to be fooled by Kanban's hidden sophistication. Kanban is far more sophisticated than a simple tool for visualization. Kanban is built for speed but most teams will never master the behaviors that produce those results because their commitment to learning Kanban stops at visualization.

3. Fluid Priorities

Scrum produces the best results when a team commits to a batch of work and is empowered to remain focused for the duration of their iteration. The success of Scrum requires informed and empathetic stakeholders who are bought-in to being agile.

Kanban is able to survive conditions where agile culture doesn't yet exist because it encourages *optionality*. A Kanban team prefers to not commit to work in batches and they don't commit to work until they start it. This means a Kanban team can be maximally flexible to respond to emergencies or changing priorities without needing to renegotiate commitments.

4. Small Teams

The ideal size of a Scrum team is a two pizza team. This ensures that a team is small enough to be efficient and large enough where the time investment in meetings makes sense. If your team is smaller than a two pizza team, Kanban is the best option.

5. Complex Collaboration

Scrum has changed the way the world thinks about work. My favorite attribute of Scrum is the cross-functional team. The idea that a team is comprised of people from different departments and disciplines is a game changer.

In the case of software development, it's not just engineers and testers anymore. Today, we have user experience, visual design, writing, editing and many other activities.

If your team has a large number of activities, the strategies of sprinting ahead or using a scaling framework may introduce unnecessary complexity and delay.

Kanban doesn't utilize time boxing to create predictability, it uses lead time—so it is capable of sustainably supporting an unlimited number of activities and collaborations.

I'd continue to encourage you to resist the idea that Scrum and Kanban are competitors or enemies. Both are a means to help teams and their stakeholders achieve sustainable success. Don't assume that whichever you best understand or most frequently use is superior. Adopt a true agile mindset and experiment with both!

Will you take 2 minutes to tell us about your experience with kanban?

Trying new methodologies can be key in succeeding with agile.

But that doesn't mean it's not challenging.

Have you tried Kanban with your team?

Have you experienced any challenges using it?

If so I'd love to hear from you.

Would you take two minutes and click the link below to take a short survey?