

A Simple Way to Run a Sprint Retrospective

by Mike Cohn • 79 Comments

A Simple Way to Run a Sprint Retrospective

There are perhaps as many ways to run a retrospective as there are teams to conduct them. I want to describe my favorite way, especially because it's an approach that has stood the test of time, having worked for years with many, many teams.

The Start, Stop and Continue Retrospective

I like to conduct a sprint retrospective by asking team members what they would start, stop and continue doing. This type of meeting becomes known as a “start, stop and continue” meeting.

The *start* items are things a team member thinks the team should add to its process. Some examples would be:

- Showing the software to customers early
- Specifying acceptance tests early and with customers
- Doing code inspections
- Being on time for daily standups
- Finishing one story before starting the next

Items on the *stop* list are things that someone on the team thinks are inefficient or are wasting time. The team should stop doing these. Examples from past retrospectives include:

- Checking in code without being sure all tests will pass
- Taking more than 15 minutes for daily scrum meetings
- Skipping [product backlog refinement](#) meetings when we're feeling behind at the end of a sprint

The *continue* list contains items the team would like to continue to emphasize but that are not

yet habits. So any of the start or stop items above could go onto the *continue* list and stay there for a few sprints.

Eventually--once the item became a habit--it would be removed from the continue list. Otherwise, the continue would become tremendously long.

Ask for Items in Different Ways

A ScrumMaster can ask team members for items in different ways. The easiest is just to say, "Yell them out," and team members are free to intersperse start items with stops and continues. This is my default mode.

But, it can get repetitious sprint after sprint. So, I'll mix things up and sometimes I'll go around the room asking each person to give me one item, perhaps making two passes around the room before opening it up for additional items.

Other times, I'll want to emphasize a specific type of item--often the stops. So I'll ask all team members to yell out nothing but things to stop doing. Or, I'll combine approaches and go person-by-person around the room asking each to identify one thing to stop in the team's current process.

There are plenty of ways to mix up the idea generation in a start-stop-continue retrospective so that it will take a long time before it gets boring or repetitious.

Vote

After enough ideas have been generated, have team members vote for the most important item or items. It's often obvious when it's time to do this because the creativity has died down and new ideas are not coming very quickly.

The ScrumMaster can have each team member vote for the one, most important idea or can use any typical multi-voting approach. For example, give each team member three votes to allocate as they wish (including all three votes to the same items).

I like multi-voting in a retrospective. The nature of most retrospective items is that many do not really take time to do it. Many are more behavioral. Consider being on time for daily standups from the examples above. That doesn't take any time. In fact, perhaps it saves time.

Multi-voting would allow a team to choose to work on that behavior and perhaps another couple of items. Generally, I'd pick no more than three. Even if they don't take any (or much) time, choosing too many items does detract from the importance of those selected.

In addition to voting for new items to pursue, discuss whether items on the continue list have been achieved, are no longer important or should be otherwise removed from the list.

The Next Retrospective

In the next retrospective, I suggest the ScrumMaster bring the list of ideas generated at the previous retrospective--both the ideas chosen to be worked on and those not. These can help jump-start discussion for the next retrospective.

I tend to write them on a large sheet of paper and tape it to the wall without any fanfare or discussion. The items are just there if the team needs them or wants to refer to them. I then facilitate a new start, stop, continue discussion.

Benefits of Start, Stop and Continue

I find that conducting retrospectives this way is fast, easy, non-threatening and it works. A start, stop and continue meeting is very action-oriented. No time is spent focused on feelings. We don't ask team members how they felt during a sprint; were they happy or sad, warm or fuzzy.

Each item generated will lead directly to a change in behavior. The team will start doing something, or they will stop doing something, or they will continue doing something until it becomes a habit.

Yes, I'm prepared for many people to leave comments saying it's important to work through people's feelings first. Or that we won't know how to act until we've first dealt with how people

feel. Go ahead. In some cases that may be true. But in plenty of other cases, we can identify what to do (“we need to start testing sooner”) directly.

And that’s the strength of a start, stop, continue approach to sprint retrospectives.

What Do You Think?

What do you think? How do you like to run retrospectives? And specifically, is there anything you’d like to start, stop or continue about your own retrospectives?

Posted: January 26, 2016

Tagged: teams, scrummaster, meetings, retrospectives



About the Author

Mike Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of *User Stories Applied for Agile Software Development*, *Agile Estimating and Planning*, and *Succeeding with Agile* as well as the [Better User Stories](#) video course. Mike is a founding member of the Agile Alliance and Scrum Alliance and can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also have Mike email you a short tip each week.
