```
Errno::ENOENT: No such file or directory @ rb_sysopen - /Users/rpowell/Sites/ee.
```

Three Mistakes Scrum Masters Make and How to Correct Them

by Mike Cohn • 82 Comments

Image not found or type unknown

Three Mistakes Scrum Masters Make and How to Correct Them

Being a Scrum Master can be tough.

Give a team too much advice and they get into the habit of letting the Scrum Master tell them what to do. But don't give enough advice and the team improves more slowly than it could. Solve some problems for the team and soon the Scrum Master is expected to solve every issue. Don't solve other problems, and the team stops even telling the Scrum Master about impediments.

Scrum Masters must walk a fine line. And it's easy to make mistakes. In this blog post, I want to help by describing three common mistakes I see Scrum Masters make. And for each I'll offer some brief advice on how to overcome the mistake.

Letting Work Slop Over into the Next Sprint

First is something I consider one of the worst habits a Scrum team can develop: allowing work to spill over from one sprint to the next. This happens when a team does not finish all of the product backlog items they've planned into a sprint and simply carry the work over into the next sprint.

To be clear: It will happen occasionally that a team doesn't finish everything. In fact, it's a good sign that it does happen sometimes because that means a team is being aggressive in what they plan rather than under-committing each time.

However, a Scrum Master and team should not take a cavalier attitude toward failing to finish. When they do, sprints become artificial and meaningless boundaries. Teams should feel a *slight* bit of pressure as the end of a sprint nears. I want the team thinking, “Uh-oh, the sprint ends tomorrow. I better stay focused today and really wrap things up.”

If spillover is a problem for your team, there are a few things you should consider doing.

First, you need to break the habit. Encourage the team to plan its next sprint such that they can definitely finish everything. That is, go light and plan the next sprint conservatively. Then, if things are going well, add more to the sprint.

Next, make sure the team feels just a little bit guilty when they don’t finish everything. I don’t want them to feel horrible. That’s not the point. I want them to feel about as bad as I do right now. I’m trying to cut down on the soda I drink. I’m making progress and haven’t had any for a week. But today I just felt like having a soda while writing this. So I did. I don’t really feel bad, but I’ll just make sure I definitely don’t have another tomorrow as I don’t want to get back in that habit.

Running the Daily Scrum

A second mistake I see Scrum Masters make frequently is running the daily scrum. I do think the Scrum Master should participate in the daily scrum and should give an update. But the Scrum Master does not need to conduct the meeting by calling on people to speak and so on.

The Scrum Master should explain the rules and purpose of the daily scrum, conduct the first two or three by calling on people, and then let the team conduct the meetings themselves.

The activities of a daily scrum are not difficult to master. The meeting does not need a traffic cop calling on people. When a Scrum Master runs the meeting that way, the discussion becomes too directed at the Scrum Master.

I like to start a new team like this: I’ll definitely run the first two or three meetings. Then I shift to simply announcing, “OK, everyone, it’s time to start.” I might say, “Who wants to go first?” But I’ll soon stop saying even that.

After a few meetings like that, I’ll switch to visibly looking at my watch when the meeting is ready to start and, if necessary, I'll clear my throat loudly enough that people get the point. I'll then stand silently until someone says, "Oh, I guess we're supposed to start." After a few days of looking at my watch and clearing my throat, I'll simplify and just look at my watch at the designated start time. After a couple of days of that, I'll stop even looking at my watch. I'll just stand there and wait for the team to decide to get started.

I definitely don't stay silent during the entire meeting. I may need to coach a person into providing more detail, or I may politely interrupt someone ("Maybe we should go into the details after the daily scrum").

Here's a challenge I'll give you: Imagine me (or any outsider) at your daily scrum. I should not be able to identify who the Scrum Master is just by watching that meeting. There will be times when a Scrum Master says something that only a Scrum Master would say, but that shouldn't happen in most meetings.

Allowing a Team to Burn Out

I'm not a fan of much of the Scrum vocabulary but the term *sprint* seems especially troublesome. When I sprint while running, I tire quickly and often walk to recover.

That's a horrible metaphor for how we'd like the team to work. One sprint ends and the next begins. Teams shouldn't begin their next sprint still trying to recover from their last. Instead, teams should work at what Kent Beck terms a *sustainable* pace.

And so the third mistake I see Scrum Masters make is that they allow teams to go beyond this sustainable pace and burn out. A good Scrum Master will be on guard to prevent team burn out.

Many teams are optimistic and that optimism carries over into the amount of work they pull into a sprint. Scrum Masters should be on the lookout for this behavior and be ready to caution teams when they seem to be committing to more work during sprint planning than they have historically completed inside a sprint. Why? Because even if a team manages to complete the work as planned during the sprint, that team runs the risk of entering the next sprint both tired and also overly optimistic about the amount of work it can complete. Such a team might then once again commit to slightly more work than it can comfortably complete. Eventually, the team will burn out trying to work at a pace that is unsustainable.Another great way for a Scrum Master to guard against burnout is by giving a team time to work on things of their own choosing.

I like to do this by introducing a cycle I call 6 x 2 + 1 (“six times two plus one”). This refers to six two-week sprints followed by a one-week sprint. What the team works on in the one-week sprint is entirely up to them. Some people will use it for personal development (reading, video training, etc.). Others might use it to refactor some ugly code that the product owner hasn’t prioritized. Others might try an experiment that they believe could lead to a great new feature. But it’s entirely up to the team.

Introducing a cycle like this actually benefits more than just the team. The product owner now has a week without any “distractions” from the team. This is often the selling point that gets a product owner on board with this.

This cycle can also benefit the organization if it is one that needs to make commitments like, “We’ll deliver this set of features in 3 months.” The 13th week of the 6 x 2 + 1 cycle can be used for a normal sprint if the team gets behind on a deadline. The team can then be given a week for their own work a sprint or two later after the deadline has been met.

What Do You Think?

Have you seen any of these problems on your team? How have you addressed them? Please share your thoughts and experiences in the comments section below.

Posted: August 1, 2017

Tagged: scrummaster, teams, leadership, sustainable pace
