

# Placing Rules on Self-Organizing Teams

by Mike Cohn • 32 Comments

This article has an audio version: [Download Audio Version](#)

## Placing Rules on Self-Organizing Teams

Many of the challenges in agile and Scrum stem from the idea of the self-organizing team. Of course, many (perhaps most) of the benefits are also the result of self-organizing teams.

One of the questions I get from many leaders is whether it's OK to mandate the team do something like use a particular tool, comply with a coding standard, or such.

Absolutely. A leader in the organization has the right to mandate anything like this. I've even seen a CEO who couldn't tell you a single difference between Java and COBOL who insisted her team use Java. And I supported her in that decision. This was back in 2000 when Sun Microsystems had announced their \$100 million "Java Fund" of VC money to companies if they used Java. So this CEO had a reason for her mandate.

So, if you're a leader in your company and have the organizational clout to make dictates: Go ahead.

But, be careful. You can give the team any rule you want, but if you give them one rule too many, they'll shut down. They will go from self-organizing to feeling, "Management just tells us what to do." And there is a very fine line here--quite literally one rule too many will push a team over the precipice.

Choose your rules wisely. Mandate that the team follow some coding standard that they determine? I've asked for that. Insist that all five teams in the company figure out and use a common testing tool. I've asked for that. Both of the applications the company produce must use the same main JavaScript library so programmers can help those on other teams. Yep, I've

asked for that, too.

But, I've also seen management mandate things that didn't make sense when the risk of pushing the team out of the realm of self-organization was considered. One PMO insisted all daily scrums occur between 9 AM and noon. This was so the PMO could prepare reports based on the results of the daily scrums. (And, yes, that was a bad idea, too.)

So, when placing a rule on a team, consider it carefully. Any one rule could push them over the edge. It won't necessarily be that rule--in fact, the rule that pushes them over could be a worthwhile rule. It will generally be the overall quantity of the rules that creates the problem.

For each rule, consider whether that rule alone is worth the risk. If it's not, don't put the rule in place. Also, any time you consider adding a rule to a team, see if there's another rule (or constraint on how they work) that you can remove. Otherwise rules build up over time. It's good to periodically review all rules you've placed on teams and see if any have outlived their usefulness.

Choose wisely.

---

Posted: January 8, 2014

**Tagged:** sprinting, management, daily scrum, testing, audio, self organizing teams

---