

A Change Would Do You Good

by Jonathan Kohl

I just visited a local bookstore where the shelves are lined with books like *Spice Up Your Diet*, *Exercise That Can Change Your Life*, and *Spice Up Your Love Life*. It stands to reason that if I get bored with these basic needs, my more complex needs, such as solving technical problems, are also going to be affected over time. If a change in my diet and exercise regimen can be beneficial, wouldn't a change to my favorite software processes do me good as well?

A senior manager and friend approached me with a set of communication problems his software development team was experiencing. He described situations where important information was missed, lost, or not made available to the right people, which led to duplication of effort, software errors, and employee frustration. To help deal with these issues, I suggested that he implement daily Scrum standup meetings. This is a meeting style where each team member talks about three things: what he has been working on since the last meeting, what he plans to work on, and any roadblocks he has to completing his work.

I checked back after a few months, and my friend was excited with the results. He said that the team was working together in ways he hadn't expected. There was less duplication of work and the shy members were much more assertive. We spoke again, months later. "You're

not going to be very happy with me," he commented. He told me his team had stopped doing daily standups because after a while, they weren't productive. They had become a hollow routine, and while people were speaking to each other, they weren't really communicating anymore.

As a response, he had changed the standup schedule. During slower times in a project, they might have one or two standups a week. When they were busier, they would increase the frequency. Closer to a software release, when team members needed more communication, they went back to a daily schedule. He explained that this change had returned them to their former level of success. Instead of being upset with him, I was overjoyed. When he recognized the tool was no longer contributing to the success of the team, he changed it.

When I was in university, one lesson I learned in a business class was that new teams have trouble being productive at first because they lack *cohesion* (uniting for a common purpose). With time, organizational support, and a healthy environment, they become cohesive and are able to be productive. However, after a while, teams can become *too* cohesive and productivity suffers. In some cases, team members may become bored and spend time engaging in distracting behavior instead of working. As managers, our job is to monitor the level of group cohesion over time and

“As managers, our job is to monitor the level of group cohesion over time and help the team introduce just enough positive change to keep it engaged in a productivity sweet spot.”

help the team introduce just enough positive change to keep it engaged in a productivity sweet spot. The lesson is: If you don't adapt your processes and practices over time, you will see early success, reach a plateau, and then regress.

We see this in other disciplines as well. On sports teams, like hockey teams, coaches understand the need for continuous change. Hockey coaches “juggle lines,” which means they change who is playing with whom as they try to win games. They don't change for change's sake, but they understand that mixing things up helps productivity. Athletes also change their training regimens, because they find that they plateau and stagnate if they do the same kind of training year after year. Moving back to software, in *Crystal Clear Applied: The Seven*

Properties of Running an Agile Project, Alistair Cockburn points out: “The people, the technology, and the assignment change over the course of a project. The conventions the team uses need to change to match.” Alistair has even recommended software teams try a different software development methodology on each project.

Methodology change is challenging. Sometimes, even “change-embracing” agilists can be surprisingly rigid when it comes to changing processes that aren't working anymore. We often put process on a pedestal and believe that merely following it will take care of everything else. We need to put process in its place. A software development process is merely a tool that we can use to help reach our goals.

One important goal is to provide value to our customers. Merely following a process and completing tasks is not always enough to satisfy and impress them. Adherence to a process shouldn't be the end goal; we should be more concerned with how our process helps us create value. The more processes we are fluent in, the more tools we have to help create value.

One interpretation of the Sheryl Crow song “A Change Would Do You Good” is that it describes a break up with a lover. Is it also time to “break up” with some of your old, ineffective software development practices and try something new? **{end}**