



Michael James is a software process mentor and Certified Scrum Trainer, focusing on the engineering practices that enable Agile project management. Having worked in the software industry for more than 20 years as a software developer (formerly “architect”), he has experience in automated testing that predates the Extreme Programming movement; formal, phased, high-ceremony processes based on DOD-STD-2167A; chaotic non-processes of the dot-com era; and Agile processes including Scrum and XP.

## Should Scrum Teams Re-estimate Work in Progress?

Posted by Michael James on July 1, 2008 at 11:45 p.m.

*I attended your training class a couple of months ago. We are continuing along with our scrum process, and we are learning a lot. I have one question - hope it is ok to ask. We did story point estimation as part of our team's Sprint planning, and came up with the total story points that we were going to tackle for the sprint. As we started working on some of the tasks, we realized that they were more difficult than we had originally estimated. Also, some people finished their tasks early and picked extra stuff from the backlog.*

*My question is as follows: Do you ever change the total story point value in the middle of the sprint (for either of the reasons that I mentioned above), and if so, how would that work if your burn down is based on story points vs time. I know we talked about it in class, but I can't remember the answer.*

I remember asking my own Scrum coach the same question a few years ago.

### Background

Scrum teams often estimate User Stories (or other Product Backlog Items) in “story points” using a “planning poker” estimation game<sup>1</sup> to get the whole team's input. This first pass rough estimation is done at the Product Backlog Item (PBI) level, to help keep the Product Owner aware of return on investment (ROI) when prioritizing work. The estimates themselves are notoriously inaccurate. But getting everyone into one room to do the estimation forces us to have conversations that lead to clarifying the items and splitting them into thinner vertical slices of product.

During the Sprint Planning Meeting, the team looks more closely at the PBIs which are being considered for one fixed-length, fixed-scope Sprint. They rewrite them a bit, break them down into Sprint Tasks, and negotiate how much they can do in one Sprint. I suggest estimating Sprint Tasks in hours, and breaking them down when they're bigger than 8-12 hours<sup>2</sup>.

**danube**

Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801

<sup>1</sup> [http://danube.com/blog/michaeljames/estimation\\_game](http://danube.com/blog/michaeljames/estimation_game)

<sup>2</sup> [http://danube.com/blog/michaeljames/task\\_granularity](http://danube.com/blog/michaeljames/task_granularity)

How does the team know how much to commit to? The PBI story points and the Sprint Task hours help. But really it comes down to gut feel and learning from the previous Sprint. I never expect teams to exactly nail the first couple Sprints. I personally prefer to keep Sprints short (like two weeks), so we learn faster.

Once the PBIs are committed to a Sprint, their story point estimates have served 90 percent of their purpose. Their only remaining purpose is to measure velocity to help inform release planning<sup>3</sup>.

### Should we re-estimate?

What should we do when we've committed work to a Sprint, and now we're in the Sprint and it's harder than we expected?

1. Increase the appropriate Sprint Task estimates, or add new Sprint Tasks to the taskboard (or tool).
2. Leave the PBIs story point estimate the same!
3. Relax. It's normal. This is why the Sprint Burndown Line often goes up before it goes down, and why trying to draw an ideal line probably won't serve you<sup>4</sup>. If the line's still trending up halfway through the Sprint, the team may want to renegotiate the commitment with the Product Owner.

If we committed to a small-looking PBI which turned out to be large, why shouldn't we give ourselves velocity credit for a large PBI? Because this would skew our velocity too optimistically for realistic release planning. There are other small-looking PBIs in your release backlog which will also turn out to be large. Icebergs look small from a distance.

So the answer to your question is "no."

### Is collaboration occurring?

I have a second concern about your team.

If I understand your email, some members fell behind, while others got "their tasks" done early and picked up new work from the Product Backlog, beyond the original commitment. I'm not yet inspired by your team's collaboration.

For the simple, low-risk work of Frederick Taylor's era, individuals (such as bricklayers) working on tasks in parallel seemed more

<sup>3</sup> <http://danube.com/whitepaper/macro-measurements>

<sup>4</sup> [http://danube.com/blog/michaeljames/tool\\_usage\\_recommendations](http://danube.com/blog/michaeljames/tool_usage_recommendations)

“efficient.” For the complex, high risk work that Scrum’s intended for, I’ll bet you’ll get better results by helping each other out more<sup>5</sup>. I wouldn’t recommend individuals work ahead of the Sprint commitment until the team’s committed work meets its acceptance criteria.

Yeah, collaboration entails learning new skills. But it brings a great opportunity for higher performance. The Sprint Retrospective is one opportunity to address this.

Hope that helps!

--mj

**danube**

Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801

---

<sup>5</sup> [http://danube.com/blog/victorszalvay/famous\\_last\\_words](http://danube.com/blog/victorszalvay/famous_last_words)