

Overheard During a Customer Conversation About Estimates

by Mike Cohn •

43 Comments

I ended up in a Twitter-storm lately defending the idea that all estimation is not waste. It truly concerns me that more and more agilists seem to be saying this. I will be the first to admit that not every project benefits from estimation. But I suspect that it's a pretty rare project where no estimation at all needs to be done.

Estimating is a way of buying knowledge. If having the additional knowledge will lead to different decisions, then acquiring that knowledge might be a good thing to do. (I can't say it will always be a good thing to do because of the cost of acquiring could exceed the benefit.) Anyone who has been in one of my Certified ScrumMaster courses will recall that during the section on sprint planning I make a point of saying that most teams spend too long estimating.

But that doesn't mean we should go to the opposite extreme and abandon estimating entirely. Whether we estimate, and how much time to spend on it, is a function of the context of the system being built. If your context doesn't require you to invest in estimates, by all means, don't estimate. But don't advise the rest of the world that they never should just because you have one example where estimating is a waste. I can give plenty of examples where estimating is wasteful. I can give more where it's vital.

The recent Twitter-storm on this, led a friend to email me snippets of a conversation he'd participated in. I asked him for permission to share those snippets and he granted it so they will form the bulk of this post. In this situation my friend ("Me" in the following) is a leader in a company doing contract development. They are in the midst of a large project and had something along the lines of the following conversation that included the ScrumMaster, two programmers who wanted to stop estimating as they found it wasteful, and their Customer, Bob.

- **Me:** Customer Bob – The team here is thinking about doing away with estimates and focusing on building high-quality software.
- **Customer Bob:** Don't you focus on high-quality software today? I'm confused.
- **Me:** Well, yes, but these guys believe that we can offer you more value and you can get

more functionality if we don't estimate.

- **Customer Bob:** I see. So let me ask a few questions. Are we still on track to complete the first half of the features we identified upfront on the contract?
- **Programmers Pete/Patti:** I'm not really sure. ScrumMaster Sam would be able to answer that.
- **Customer Bob:** Ok, what say you ScrumMaster Sam?
- **ScrumMaster Sam:** Well – we haven't been able to forecast anything beyond this past release because we haven't planned for the future release yet.
- **Customer Bob:** I understand. So I'm going to need to do a pulse check this next Release to see how we're going to end-up at the end of the next quarter. That means I'm going to need to have estimates (at least gross / high-level estimates) on the Backlog.
- **Programmers Pete/Patti:** Why don't we just follow your priorities and focus on development – they're in priority order right?
- **Customer Bob:** Well yes. When we funded this initial increment it was based on your initial estimates. I realize they can change and I get the agile thing, which is why we've written the contract to allow you to focus on my prioritized backlog. That said, we had to do our due diligence because we've collected funds from multiple stakeholders to fund this program. They want to know what they're going to get for their money, and when they're going to get it. Regarding follow-on funding – if we estimate and determine that we're nowhere near where we said we would be then I'm going to have to answer to the stakeholders. I've been very focused on not deviating from our (minimal marketable feature) requirements because we have multiple parties to satisfy.
- **Programmers Pete/Patti:** Well if we don't complete it all in this funding increment – can't we just tack it on to the follow on phase?
- **Customer Bob:** *What follow-on phase?*
- **Programmers Pete/Patti:** The next funding increment we were talking about last time.
- **Customer Bob:** There's no assurance of doing that. If we're not able to estimate completion within a reasonable level of confidence I may have to put any subsequent work up for competitive bids.
- **Customer Bob:** Programmer Pete/Patti - First of all, in order for me to even consider giving future funding I need to know how close our estimating process got us this go-round. When I ask the stakeholders for more money they're going to think it's going into a black hole if we only accomplish ½ of what we said we would. I will manage this but I need to know we have a reasonable approach for coming up with high level estimates. Second – to get follow on funding, I'm going to need to know roughly how much we need to collect (this is a long lead time activity). We'll have to improve our estimates if they're

nowhere near accurate and I'll beat the pavement to get the funding. That said, money doesn't come for free. I have to have some level of metrics that tell me throughout the project that we're tracking to completion. In other words – I'll look at the project burndown and watch our costs throughout the project.

- **Programmers Pete/Patti:** Ok, I get it that we need to do project (high-level) estimates so you know how much money to collect. Can we maybe get rid of the estimates on the Sprint Backlog (hours) and just follow our velocity. We were thinking of going to a pull-based system.
- **Customer Bob:** Well – I'm glad you'll have a velocity now based on a gross estimating approach (story points). A pull-based system sounds nice, but it doesn't sound like it's going to give me a better sense of what we can accomplish in a sprint. Your velocity is somewhat stabilizing, but you're still committing to more work in Sprint Planning than you can accomplish. It's a false sense of security – let's bring it down and be realistic. So far, your team has done a pretty poor job of estimating during Sprint Planning. I say this because the last 4 sprints you've had to drop functionality and I haven't heard you talk about improving your estimates during the Retrospectives at all. Why do I care? I care because the projected velocity you have to meet the backlog was determined based on our MMF's. If every sprint you're coming in at 20% below that rate – guess what – by the end of the project we will not meet our MMF needs. The only way I have to know we're getting there is to track our progress and work with the SM's when I see a problem. Right now – your team is estimating during Sprints that you'll have Velocity of X and you're consistently coming in much lower. Now you want to throw that away altogether (I suppose if it's broke – throw it away instead of fix it?). I'm ok with coming in lower, but then we'll have to add more scrum teams, funding or resources to meet our MMF.
- **Customer Bob:** Bottom line, money doesn't grow on trees. I'm willing to be flexible and accommodate change on the project – however, I need you to take estimating seriously. Being a good steward of other peoples money means we have to perform according to some reasonable set of expectations. We can reset those expectations, but if you don't take estimating seriously we'll never be able to give predictions to the funding community with any level of confidence. Since features do change, that means we need to estimate on this project.
- **Programmers Pete/Patti:** Alright – I guess we're estimating. I didn't realize all the complexities of the “bottom line” and collecting funds from multiple parties. We'll get started on preparing the backlog to estimate this last release. Most importantly, we know what to talk about in this next retrospective.

I got permission to share this because I think it points out a variety of examples of why estimating *may* be needed on some (many, I think) projects.

In follow-up discussions with the friend who shared this, he made a very good observation: Notice that the need for estimates above is present regardless of which agile development process the team is using. It doesn't matter if they are using Scrum, XP, Kanban, etc., their customer needs to know about more than the next week or two.

So: I don't always estimate. But it's a very helpful skill to become proficient at and is useful on many, perhaps most, projects.

Posted: November 30, 2012

Tagged: story points, scrum teams, product backlog, sprint planning, estimating, velocity, metrics

About the Author

As the founder of Mountain Goat Software, 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*. Mike is a founding member of the Agile Alliance and Scrum Alliance. Mike 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.
