How to Estimate Velocity As an Agile Consultant

by Mike Cohn • 31 Comments

Many of you work in a dedicated in-house team, but some of you contract with companies for Scrum and agile consulting. One problem that sometimes arises as an agile contractor is when the prospective client wants an upfront commitment on the scope of the project, but the Scrum consultant feels there isn’t enough info to estimate a backlog for the RFP phase. Add to that the fact that you are either estimating on behalf of other teams, or your own without prior history of the project, and it can be challenging.

So what then?
This challenge really comes down to being able to estimate velocity even in situations that aren’t cut and dry. Chapter 16 in the *Agile Estimating and Planning* book talks about this.

To be clear, points are relative, and it's hard to compare teams, but if a team builds project A, then is about to start project B, we can use their A velocities to predict B as long as the points are consistent across projects. If a 5-point story on A is expected to take as long as a 5-point story on B, use the historical velocity.

The problems enter when a team has no velocity data.

**What to Do if You Already Have a Team Assembled for the Project**

Let's assume you have the team – they just don’t have any velocity data. Have them estimate the required stories in story points the normal way, even if they have no idea what their velocity will be. They do know, however, that the sum of the stories is, say, 300 points.

Now, have them plan a sprint.

My preferred way to plan a sprint is to grab one story, split it into tasks, estimate the tasks in hours and ask the team if they can commit to finishing it. Then repeat with additional user stories until the sprint is full.

Suppose a team does this and commits to stories X, Y and Z. They couldn't have discussed velocity, as they have no data. But, they do have points on all stories including X, Y and Z.

So, look at the points on those three stories, add them up, and that’s a starting point for velocity. If you can, get the team to do this again by planning another sprint that way and commit to M, N, O and P, which is probably a different number of points.
Use the two values as a range. Say 18 to 22 points. Maybe adjust those based on your intuition. If you think the team, like most teams, is about to overcommit in their early sprints, use estimates a few (or many) points lower based on your judgment, but based on what they came up with.

What to Do if You Don’t Have a Team Ready to Go

Let’s make the situation worse. Let’s assume you don’t even have the team yet. You’ve got plenty of employees, but you don’t know who will do this project. Get some representative people together and have them pretend to be the team and do the steps above.

Then you can tell a boss or customer "a project with three senior programmers, a senior tester, a junior tester, a good designer and a great database person can do this project in X sprints."

Or make some adjustments based on differences in how you think you'll really staff it; change
the three senior programmers to be two or more junior programmers and lower the velocity based on your expert opinion.

Using data from other teams can help. I like to make these types of predictions with velocity ranges. Suppose you have other agile teams. Calculate the relative standard deviation in velocity for those teams. (Relative StDev is just StDev but expressed as a percentage.)

Average those relative standard deviations over all teams. Say you have teams with 15 percent, 20 percent and 25 percent. That's an average of +/- 20 percent. Use that for the new team. If they estimate velocity as 20, you could add + and – 20 percent.

Often though, I'll take the estimated velocity as the high value and go down by 40 percent, just given the likelihood of teams overcommitting.

Posted: July 25, 2013

Tagged: user stories, sprinting, estimating, story points, velocity