

# Capacity-Driven Sprint Planning

by Mike Cohn • 77 Comments

## Capacity-Driven Sprint Planning

There are two general ways to plan a sprint: [velocity-driven sprint planning](#) and capacity-driven sprint planning. In last week's post, I described velocity-driven planning; so in this week's, we turn our attention to capacity-driven sprint planning.

A capacity-driven sprint planning meeting involves the product owner, Scrum Master and all development team members. The product owner brings the top-priority product backlog items into the meeting and describes them to the team, usually starting with an overview of the set of high-priority items.

## Select a Product Backlog

Item Following that, team members select a first item to bring into the sprint. This will almost always be the product owner's top-priority item, but it is possible that the product owner's top priority has too many open issues.

Ideally, a team should be able to still bring that item into the sprint and resolve the issues early enough in the sprint to complete the item. But, it's possible that there are so many issues, that the issues are so significant, or that resolving the issues would take so much time (for example, the need to convene a meeting with 25 user representatives) that the product owner's top priority is skipped.

## Discuss that Product Backlog

Item Having selected a high-priority item, team members discuss the work involved, and identify the tasks that will be necessary to deliver the product backlog item.

Most teams will also *roughly* estimate the hours to do each task. These estimates are rough because the estimates will be used only to influence how many and which product backlog items are brought into the sprint. To do that, estimates do not need to be precise.

Do not ask or expect a team to think of every task that will be done during the sprint. Not only is that impossible, it is also unnecessary.

Teams should think of enough of the tasks that they feel they have thought through the work—but it is important to realize that thinking through the work is the real goal of this meeting. Identifying tasks and hours is secondary. For most teams, though, it's the best tool to know how many product backlog items to bring into a sprint.

## Decide if the Item Can Be Completed Within the Sprint

After they've identified tasks and roughly estimated the hours for that one product backlog item, the team members ask themselves, "Can we commit to this?"

I find it very important that the team members ask this collectively of themselves rather than having a ScrumMaster ask, "Can you commit to this?" When team members ask, "Can we commit?" they are committing to each other rather than to the ScrumMaster.

I don't know about you, but my early, pre-Scrum career is littered with broken "commitments" to bosses who asked if I could deliver something while making it clear my answer better be yes.

The ScrumMaster isn't a boss and shouldn't create that type of feeling among team members, but the person is called "master" – and it's better not to risk being perceived as a boss insisting on a commitment.

Coach a team to ask, "Can we commit?" and it's clear that they are committing to one another, which will likely be a stronger commitment.

Further, by having the team ask themselves, "Can we commit?" it is clear that the answer should be, "Yes we can" or "No we can't." When a ScrumMaster asks, "Can you commit?" some team

members will properly answer with “we” but others will answer with “I.”

Scrum demands a full-team commitment: If you’re behind, I’ll help, and I know you’ll do the same for me. It’s not “these are my tasks” and “those are yours.”

## Repeat with More Product Backlog Items

If the team agrees they can deliver a product backlog item, they select another item and repeat the process. And they continue repeating it until someone says they cannot commit to the selected product backlog item.

If someone cannot complete the item within the sprint, team members will generally discuss the situation and see if someone else is available to help—perhaps a DBA with rudimentary JavaScript skills can help an overwhelmed JavaScript developer.

If not, perhaps that item can be put back on the product backlog but a smaller item can be brought in, or an item that needs less of the skills possessed by the person who could not complete the work.

## What About Story Points and Velocity?

You may have noticed that in the process so far, there has been no role for story points or velocity. Although I still recommend that product backlog items be given quick, high-level estimates in story points, neither story points nor velocity play a role in capacity-driven sprint planning as described so far.

They do, however, play an important role in the final step of a sprint planning meeting.

## Sanity Checking the Commitment

Once team members have filled their available time in the sprint, the Scrum Master can look at the selected product backlog items, sum the story points assigned to each, and share that sum with the team. Team members can then compare it to the average or recent velocity.

Suppose a team with an average velocity of 20 conducts a capacity-driven sprint planning meeting and selects 19 points of work. They've done this without knowing the story point values on any of the selected product backlog items.

When their Scrum Master tells them they've just selected 19 points of work and have an average velocity of 19, that team should feel very confident they've selected an appropriate amount of work for the sprint.

Suppose instead, though, that the ScrumMaster for this team announces they've selected only 11 points of work. They might in that case ask themselves why they were making the work so hard during sprint planning as compared to when they'd earlier estimated the same items in story points.

For example, this may reveal that during sprint planning they'd identified work they'd earlier not thought about, or perhaps had explicitly assumed would not be part of a given story. Or they may discover that the story really is harder than they'd thought when assigning points to it.

Either way, knowing they'd selected 11 yet averaged 20 will help the team know they've selected an appropriate amount of work or perhaps make a change to bring more.

Similarly, if the ScrumMaster announces that the team has selected 30 points, 10 more than their average velocity, team members may wonder what they are forgetting to consider. "Why," they would discuss, "does this work seem so much easier after sprint planning than it did while estimating story points?"

So: story points and velocity do not play a role during the main portion of a capacity-driven sprint planning meeting. But they play the vital role of acting as a sanity check and confirmation of the plan.

## It's a Commitment, Not a Guarantee

It is important that the team's commitment not be viewed as a guarantee. As Clint Eastwood said in one of his movies, "If you want a guarantee, buy a toaster."

The team's commitment is to do its best. I'd like to see them make their commitment perhaps 80 percent of the time. It should be something they take seriously and should make most of that time. That's needed for the business to gain confidence in what a team says it can deliver.

However, finishing everything they say they will 100 percent of the time should not be the goal. A team forced to finish everything every time will do so—but by reducing what they commit to.

## Read the Rest of This Series on Sprint Planning

This post was the second in a three-part series on sprint planning. Part one covered [velocity-driven sprint planning](#), an entirely different approach. In part three, I'll share my [preference between velocity-driven and capacity-driven sprint planning](#).

### Note

This post was originally titled "Commitment-Driven Sprint Planning." I've re-titled to the more appropriate term "Capacity-Driven Sprint Planning" since the original term led to frequent misunderstandings that a commitment was a guarantee. Minor edits for consistency were made to the rest of the post.

---

Posted: October 28, 2014

**Tagged:** product owner, estimating, sprint planning, commitment, commitment driven

---

