

# How Full to Fill a Sprint

by Mike Cohn •

10 Comments

Image not readable or empty  
Gas Gauge at Almost Full\_1790060Large.jpg

An important consideration in [commitment-driven planning](#) is how full to fill the sprint. To answer that, we need to understand that a sprint includes three types of time.

The first type of time is corporate overhead. Corporate overhead includes everything required to be a good citizen in today's corporate world. That includes time spent in all-company meetings, reading emails, going to HR sensitivity training classes, and even attending the various meetings for a team's agile process.

The second type of time is plannable time. This is the time that team members can plan to use on the real work of the sprint. A team member does not, however, want to fill the sprint too full. Team members need to leave time room for some unplanned time, which is the third type of time.

Unplanned time goes toward three things:

1. Emergencies. This is something like the server going down and your team being asked to fix it.
2. Tasks that get bigger than expected. For example, in sprint planning you think a task will take 8 hours. Later, after spending 6 hours on it, you realize you still have 6 to go.
3. Tasks that are not identified in sprint planning. No matter how hard you try during sprint planning, you aren't going to think of everything.

Graphically, I think of the sprint as a rectangle, or anything with area as you can see in the figure here.

Image not readable or empty  
Chart of Unplanned Time, Plannable Time and Corporate Overhead in a Sprint

The first thing we put into our sprint is corporate overhead. Next, the team loads the sprint with plannable time—but not so full that the slightest problem causes the sprint to overflow. They leave room at the top of the rectangle for unplanned time—emergencies, tasks that get bigger, and tasks they don't think of.

Determining how full a team should fill a sprint is really a question of how much plannable time they have. Consider two teams. The first is part of a large organization with lots of corporate overhead.

This team is always responsible for second-tier tech support on the existing application while trying to move forward on a next-generation product. This team may not have much plannable time at all.

The second team is two programmers in a garage working on a startup's first product. Corporate overhead for them is who is going to order lunch for the day. They might have much more plannable time than the first team.

Similarly, consider an amazingly agile team that sucks at estimating their work. They are horrible at it. They think something will take two hours and it takes 10. And they rush through planning, so they fail to identify many tasks they could have identified if they'd tried harder.

Because they're so bad at estimating, this team might need to leave a lot of unplanned time in their sprint. Their plannable time would be less than that of an equivalent team that could estimate better.

This means a team's amount of plannable time cannot really be used as a measure of how hard

that team works or of how good they are.

If you are an experienced agile team, you'll find that your amount of plannable time is also the vertical intercept on your sprint burndown chart. This is sometimes called a team's *capacity*. This is distinct from *velocity*, which should always be measured in the units used on a team's product backlog items, usually story points.

---

Posted: March 10, 2015

**Tagged:** scrum teams, velocity, sprint planning, commitment, commitment driven, time

---

## About the Author

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* as well as the [Better User Stories](#) video course. Mike is a founding member of the Agile Alliance and Scrum Alliance and can be reached at [hello@mountaingoatsoftware.com](mailto:hello@mountaingoatsoftware.com). If you want to succeed with agile, you can also have Mike email you a short tip each week.

---