

Demystifying Sprint Planning

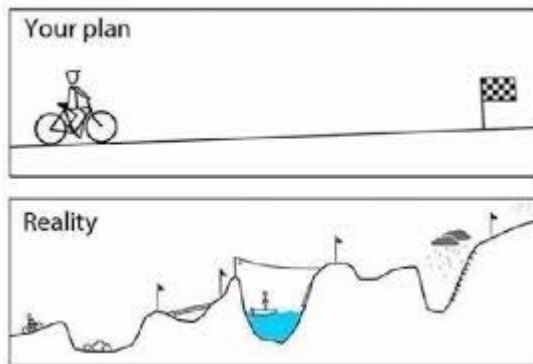
10 October 2017



Hemant Kothiyal

United Health Group

Becoming proficient at planning is a critical skill for any Scrum team. Given the Agile Manifesto's emphasis on "Responding to change over following a plan," it is clear that the perfect plan will never exist in the real world. Hence we have seen that many developers hate plans. Sometimes a plan is used as a weapon against them: "You said it will be done by this week, so make it happen." It is wrong to use a plan as a weapon — but it is equally wrong to not do any planning at all.



In this article I am going to cover some of the more common planning challenges and how to handle them. I have divided this article into three parts: The first covers confusion around sprint planning, in the second I will recap the definition and purpose of sprint planning, and in the last I will share how I have handled common challenges in these areas.

Confusion regarding sprint planning

Based on my experiences with various Agile Teams' planning meetings, I am able to list the following types of confusion as common.



According to the Scrum Guide, the sprint planning meeting is timeboxed to two hours for a weeklong sprint. The ScrumMaster has to ensure that the event happens and also educate the team to keep it within the time limit.

The purpose of sprint planning

Sprint planning should answer the following questions:

- What can be done in this sprint?
- How will the sprint goal be met?

Backlog refinement (What can be done in the sprint?)

Product Owner – Discusses the objective that sprint should achieve

Development Team – Understands and sizes the stories to be delivered.

Usually the team handles these activities a couple of days before sprint planning day (also known as the backlog refinement meeting). Backlog refinement sessions are not only for breaking stories into smaller pieces. Other activities can include:

- Writing stories if time permits
- Sizing
- Asking questions about a particular story
- Establishing acceptance criteria for a story
- Prioritization
- Eliminating waste: Deleting stories/features/epics that are no longer needed

How will the sprint goal be met?

It is purely up to the development team to decide how they will build this functionality into a “Done” product increment during the sprint.

Typical outcomes of this meeting are:

- The sprint goal
- A critical tasks list
- A plan for a self-organizing team
- Acknowledgment of team capacity, taking holidays and trainings into consideration if applicable

Handling the challenges

Now let's look into some of these challenges that we face every day, and examine ways to tackle them.

Scenario 1: Utilization challenge



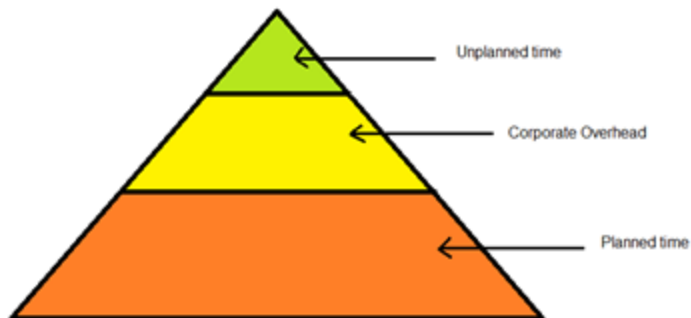
Don't focus too much on 100% utilization of every team member from day 1

Prefer Team commitment and team velocity over Individual capacity.

Planning should be around discussions to meet sprint goals rather than capacity planning.

Typically, capacity planning is a tough task. As Mike Cohn mentioned notes on his site at <https://www.mountingoatsoftware.com/>, there are three possibilities of capacity planning: planned time, unplanned time, and corporate overhead. Try to help the Scrum Team plan with the following possibilities in mind:

Capacity Planning



Unplanned time:

- Tasks discovered post sprint planning
- Tasks bigger than original estimate
- Unforeseen situations

Corporate Overhead:

- All company meetings
- Trainings
- Answering email from your past projects
- HR sensitivity trainings

Plannable time:

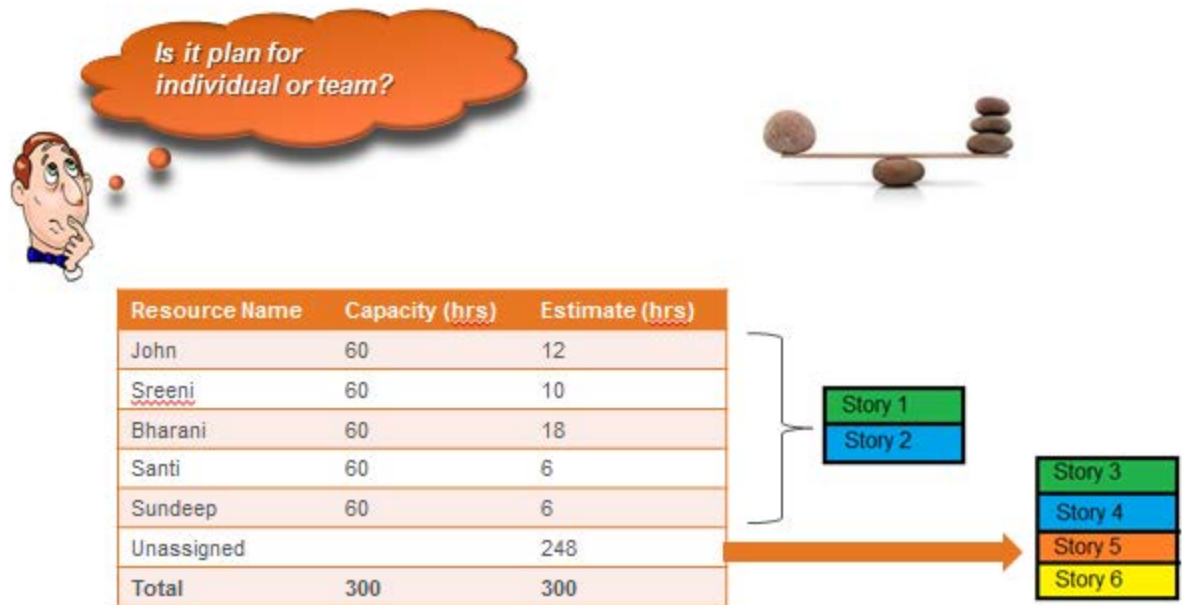
- Time goes into sprint
- Scrum meetings

Now let me share a scenario from my Scrum Team. We end sprint planning having identified critical and important tasks for all stories. The two main story developers have picked some tasks. The team wants to "start finishing and stop starting," so you can see that Stories 3 through 6 are not yet assigned to anyone. We decided to assign those tasks on the next day of the sprint, based on the progress of the top two stories.

If anyone is thinking that we are underallocating team members, we can see here that the total available capacity for the team is 300 hours, and the total of all task estimated hours is also 300 hours. Hence this team is well occupied.

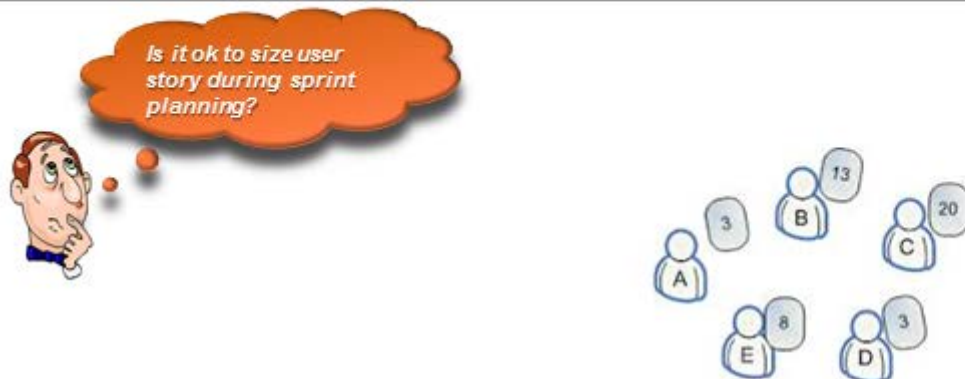
My takeaway is to always provide an opportunity for the entire team to collaborate and contribute to each story before assigning any story to one developer alone.

Scenario 2: Commitment challenge



Encourage team collaboration and contribution on user story based on priority.

Scenario 3: Timing challenge



Don't encourage user story sizing during sprint planning. It consume time and deviate team's focus.

Take this activity in Backlog refinement, few days prior to sprint planning.

Encourage sizing to be complete prior to sprint planning.

Scenario 4: Sprint goal challenge



According to Scrum Guide, we can change user stories during sprint as long as:

- No changes are made that would endanger the Sprint Goal.
- Quality goals do not decrease; and,
- Team collaborate with Product Owner to negotiate the scope of Sprint Backlog within the Sprint.

Changes are allowed only if sprint goals are not compromised

Scenario 5: Participation challenge

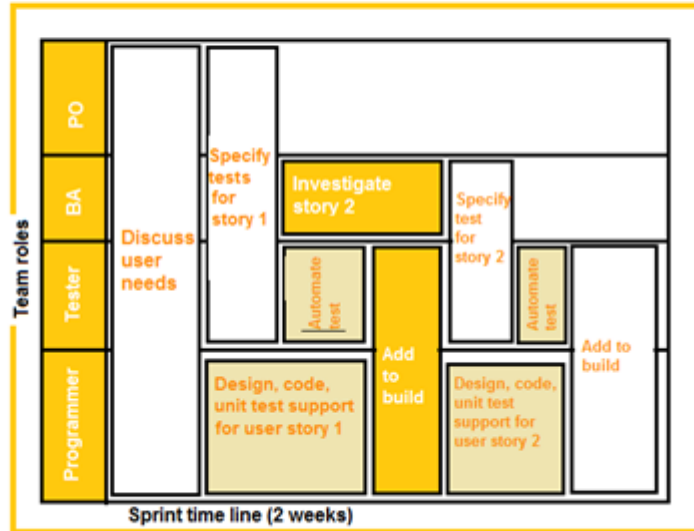


- Its mandatory to have everyone in sprint planning.
- Keep one goal- How to achieve early acceptance with meeting DOD
- Pick first story - Ask entire team to create their self organizing plan as an output pf their tasks.
- They will come up with a plan whether they need swarming to achieve this or not.
- Ask them to use white board etc. to have some draft plan ready with initial tasks.
- Repeat same steps for other stories.

Goal is to have entire team participate in creating sprint execution plan.

Following are steps I always use in sprint planning to create opportunities for the team to value collaboration rather than allocating individuals to each story. I have been inspired by Mike Cohn's book *Succeeding with Agile*. He clearly explains that you should try to discover how can different roles within the team see opportunities to become involved in the user story right from Day 1. It is always possible to start several tasks in parallel and have a "handshake" at the appropriate time, as shown below.

Planning steps



With one team, we got outstanding results. We had a story total estimated at 60 hours but it was delivered in three working days, because when one developer was writing code another was writing unit test and integration test cases. Meanwhile, BA were creating test data. On the third day, they all delivered the user story with quality code and proper automation testing.

I have previously written about identifying tasks during sprint planning, at <https://www.scrumalliance.org/community/articles/2017/september/should-teams-identify-tasks-during-sprint-planning>.

I am sure you all are also facing such challenges and creating interesting and innovative solutions. I would like to hear from anyone who would like to share real-life experiences from your projects.