4.1: “Planning” Day

This chapter discusses the “day” that takes place between sprints, which includes the Sprint Review, the Retrospective, and Planning for the next Sprint

Table of Contents

All teams are different, but there are some commonalities beyond just the basics of scrum. For example, many of the teams we coach try to limit the amount of time they aren’t actually “doing work” – that is, not actually working on stories.

In order to “waste” as little time as possible, they try to limit the amount of time they spend between sprints. That is, they try to minimize the time they take on the Sprint Review, the Team Retrospective, and Sprint Planning. In practice, they try to do all of these activities in one day. There are two patterns. The first one does it all in one Calendar day, and is the one we describe in this chapter. The second spreads the day across two calendar days, with the Review the first afternoon, Planning the next morning, and the Retrospective in between – wherever it fits.

In this chapter we describe this day… and we’ll explain it with an artificial timeline just to get our points across. Don’t take this to mean that we think this timeline is the only way to go, or even the best way to go – we don’t – it’s for illustrative purposes only.

Sprint Review with Stakeholders (10am to Noon)

In this example, we hold the review in the morning. We don’t do it too early, because we need the Stakeholders to be there awake and alert. We also want to get prepared, and if we hold it too early in the morning we’ll be tempted to spend the previous day getting ready. This would be *really* bad… so we try to reduce the pressure to do so.

You don’t want to spend *too* much time preparing for the review; you want it to be part of the natural flow of things. But most teams can’t help it; they feel a need to prepare a “dog and pony” show, even if it’s just a small one. So, in the morning before the Review you get ready for it. By limiting it to the early morning, you spend no more than 1-2 hours on it. This limits the damage…

When preparing for the Review it is important to remember that the team can’t demonstrate or review anything that isn’t done; that is, only stories that have met their “doneness” agreement can be used in the Review. We like to say that there is no “partial credit” in scrum.

That being said, let’s get real here. The team may want to review something that’s in progress in order to gather information that will help them finish it. If you feel that you need to do this, be open, honest, and visible about it. Make sure the Stakeholders know that this thing isn’t done; that the team is just taking advantage of the fact that the Stakeholders are there to get information that will help then finish it.

Once we’re prepared for the meeting, it’s time to actually do it. There are a few things to remember. First and foremost, this is the Product Owner’s meeting. The Product Owner is accountable to the Stakeholders for the Team’s performance, and this meeting is for the Product Owner to review with the Stakeholder what the Team has done for them. Of course, most of the meeting actually consists of Team Members working with, and demonstrating things for, the Stakeholders.

What things get demonstrated? Is it a “show and tell” of everything the Team did in the last Sprint? No! The Sprint Review is a highly-focused meeting that provides Stakeholder feedback to the Team about its direction and priorities. In order to do this, the Team must demonstrate two categories of things:

1. The Team must show the Stakeholders what they (the Stakeholders) need to see in order to understand the direction the team is going, and
2. The Team must show the Stakeholders what they (the Team Members) are confused about and need feedback on.

In other words, we must maintain our focus. Our focus in the Sprint Review is to get feedback in order to determine if the Team is going in the right direction, and to change direction if it isn’t. It is not required, but it would be nice if the Team and Stakeholders agreed on a unifying Goal for the next Sprint as part of this Review.

There are many ways to do this and teams will find their own way, but it must be stressed that we don’t want to bore the Stakeholders by telling them *everything* we know. Keep the meeting focused on the two issues given above. Ask the Stakeholders what they want to see next time. Make sure the Team uses this meeting to decrease their uncertainty about what needs to be built.

Keep your eyes on the ball… keep focused on the following three purposes for the Review:

- To have the Stakeholders gain and retain confidence that the Team is going in the right direction – that the Stakeholders are getting what they expect;
- For the Team to maintain confidence that they are going in the right direction – that they are producing what the Stakeholders want and need; and
- To make necessary modifications in the directions and guidance to the team – to make appropriate changes and modifications to the Release Plan/Strategy. Here are a couple of things you can do to help with this, that will help with focus and setting expectations...
    - Have the Stakeholders work with Team to help determine the Team's Sprint Goal(s) for the next Sprint. Having a single goal is ideal, and having fewer than two or three is crucial.
    - Ask the Stakeholders what they want to "see" in the next Review. This will help with the definitions of done, in order that what needs to be presented is actually presentable.

Ultimately, remember that the main purpose for the meeting is more ambitious – you are trying to get the Stakeholders to understand that they are important – that they are actually driving the project. Without their feedback things may fail – they will certainly be worse than they could have been. It is their feedback that gives the Product Owner the information that he/she needs to be strategic agile.

Release Re-Planning (lunch, Noon – 1pm)

After the Team has its review with the Stakeholders, go to lunch and let the comments of the Stakeholders sink in. Maybe the Product Owner needs to take a couple of the key stakeholders to lunch. This is a good time to discuss the overall direction of the project, taking into account the review just completed.

The Product Owner may need to change direction, change the next sprint's goals, or do some significant reprioritization of the backlog prior to the planning meeting coming up. This is the time to iron this out with the stakeholders, if it wasn't done during the review meeting itself.

If you believe that this re-planning will take a long time, then this is a good reason to split the planning day across two days, with the review in the afternoon of the first day, and the retrospective and planning meeting the second day. This allows a good dinner discussion about the redirection of the project.

Sprint Retrospective with the Team (1pm to 3pm)

After lunch it's time for the Team to take ownership of its process with a Sprint Retrospective. There are many techniques for retrospectives, and we won't go into them here[1]. What we will do is make some comments about things that are important to know about Sprint Retrospectives.

First, the Product Owner is a member of the Team, and *must* be present at the Sprint Retrospective. Everybody on the Team must be present at the Sprint Retrospective. The Sprint Retrospective is a formal meeting for the whole Team to address its process – it's not about individuals or sub-teams.

Within the Sprint the ScrumMaster should be having frequent Intraspectives with various subsets of the team in order to address issues as they come up. I use the word "intraspective" to represent retrospective-like discussions that happen *within* the Sprint, while the Retrospective is the formal meeting at the end. Think "retro = look back" and "intra = within" and you'll get the idea. These intraspectives are the proper times to have retrospectives without the Product Owner. Of course, if you don't do them then, you may need to excuse the Product Owner from the retrospective for a few minutes to let an open discussion take place.

Second, the Retrospective is not just a "whine" session. The most important thing to determine at the Retrospective is what worked well. It is very important that we don't stop doing what works for us; and nothing upsets a team more than sliding backwards. After we know what we don't want to change, we can then think about what we'd like to change.

Third, we want everybody on the team to be involved in the Retrospective; we want everybody to be respected during the Retrospective. This is hard to do when we have quiet people and noisy people on the team; if our Retrospective was simply a conversation, the noisy people would dominate.

---

[1] See, for example, the book "Agile Retrospectives" by Esther Derby and Diana Larsen.

So, we usually do our retrospectives in two parts: first we have a data-gathering section (typically by writing, some interactive activity, or game), and then a conversation. The purpose of the data-gathering section is to get everybody’s ideas out on the table where they can be used as an agenda for the conversation. Typically, the ScrumMaster facilitates the Retrospective, and manages the conversations and the data-gathering.

There are many issues to manage, including power dynamics involving the Product Owner, process issues, the definition of “done”, and so on. It is up to the ScrumMaster to facilitate these discussions of these issues amongst the team.

Fourth, when you do decide to change the process, there are three types of changes:

- Decisions the Team can make and implement immediately, such as a change to the Team’s definition of “done” that will be implemented in the Planning Meeting later in the Planning Day. For example, the Team may decide that they need to do a team review of the User Interface at a whiteboard before they start coding;
- Changes that are under the Team’s control, but will take time and effort for the Team to implement. Examples could include improving the build process, beefing up test tools, rearranging the Team room, and so on. These changes must be written as stories and added to the Backlog, so that they can be prioritized and done eventually; and
- Changes that are outside the Team’s control and must be accomplished outside the Scrum Team. In this case the ScrumMaster (often with the Product Owner) goes outside the Team to the Business Owner (and other Stakeholders) to get some help. If the change will take a lot of time and/or effort, and the Business Owner can’t get it done immediately, it is usually managed by the ScrumMaster Community as a whole – the ScrumMaster Community is the primary “agent for change” in an organization using scrum.

Notes on Changing the Team’s Process

There are two metaphors I like to think about when changing a Team’s process. I learned these from an ex-boss of mine, Alan Shalloway of NetObjectives. These two metaphors are the “Pick up Sticks” metaphor and the “Trim Tab” metaphor, which I present to you now.

“Pickup Sticks” is a children’s game that involves dropping a bunch of colored sticks on a table, and then removing sticks one at a time, without moving any other sticks, until the “gold” stick is removed[2]. This metaphor is useful to us because we may want to make a big change (represented by the gold stick), which will require us to make many small changes before we can get there.

As Alan reminded me, adults can only learn something they “almost already know”, and the sticks we remove on the way to the gold stick are small enough changes that the team “almost already knows” them. It takes time to get to the big changes we want, and making big changes is an incremental process. In fact, it’s an agile process. Inspect and adapt along the way, but don’t lose track of what the team wants and needs.

The second metaphor is the “Trim Tab” metaphor. Buckminster Fuller[3] introduced this metaphor in 1972, and it is very relevant to us when considering change. First of all, you need to know what a “trim tab” is, so what is a “trim tab?” Basically, when it comes to ships, a trim tab is a little rudder that is attached to a big rudder. As Fuller put it:

> *“Think of the Queen Mary – the whole ship goes by and then comes the rudder. And there's a tiny thing at the edge of the rudder called a trim tab. It's a miniature rudder. Just moving the little trim tab builds a low pressure that pulls the rudder around. Takes almost no effort at all.”*

In other words, a trim tab is something that does something small that makes something big happen as a natural by-product. When it comes to having teams change behaviors, we should be looking for the trim tabs, which are small changes in behavior, process, or environment that will cause the team to move, change, or evolve, in a desired direction.

There are many potential trim tabs that could be used. Some of the best ones focus on the team values we discussed in chapter xx (focus, respect, courage, etc), or provide changes in the “doneness” criteria we discussed in chapter xx. It is my opinion that any trim tab that either causes more conversations or causes the team to do work in smaller “bites” is a good one. Let me tell you a story about the best trim tab I ever saw…

I was visiting a client site where I had previously established many scrum teams, and was coaching some of the newer ScrumMasters. One of the established ones – call him Dave – came up to me and said: “Dan, come look at my trick.” Now, Dave was a seasoned Project Manager turned ScrumMaster, so I knew that any trick he had was going to be a good one. So, I went with him to witness his Team’s Daily Standup.

---

[2] This is one version, anyway, that is useful to us here.

[3] Buckminster Fuller was a famous scientist/engineer and this discussion was in Playboy, in 1972.

The trick was very simple. For every Story that was completed, Dave gave a single M&M® candy to everyone who had worked on that story. The members of the scrum team had naturally developed a friendly competition to see who could get the most candy. I heard one of the senior team members talk about filling a shot glass with M&M's® in order to eat them all at once[4].

I immediately saw why this was a cool trick: it both encouraged conversation (more people working on the same story) and caused smaller bites of work (stories got smaller so that more got finished). I asked Dave if he had any idea of what improvements this had caused. I was confident that Dave had an answer, because he was a metrics guy from way back.

He told me that this had caused a 25% increase in delivering releasable features over the previous two months. I took him at his word, as I had no time to discuss how he measured this. So, I asked him how much it cost, and he answered "two dollars a week."

Wow! A 25% improvement in feature delivery for $2 a week! I've never heard of a better ROI or a better trim tab. Brilliant! Go Dave!

So, in your Retrospectives, first determine what you like, and don't change that. Then figure out what you'd like to change. Then try to look for trim tabs that will cause that change. If you can't do it with a trip tab, use brute force. Good luck!

Sprint Planning (3pm to 5pm)

Finally, at the end of the day the Team plans the next Sprint. The Team knows what the Stakeholders thought of the previous Sprint's work, they also know what changes they'll be making in their process, and they may even have the next Sprint's Goal. So, it's time to plan.

Now, the Sprint Plan is not really a plan. What it is is a commitment, by the Team, about what they'll be doing in the next Sprint. This commitment is amongst the Team Members (including the Product Owner) with the Product Owner representing the interests of the Stakeholders. This commitment can also be thought of as a commitment of the Team to the Stakeholders.

---

[4] The only proper way to eat M&M's®, as we all know ☺

So, we have a commitment. We may also have focus, direction, goals, priorities, and so on. What we're trying to do in Sprint Planning is get everybody on the team on the same page, which is represented by the Sprint Backlog. Then the Stakeholders also get to know what that page is by looking at the Sprint Backlog.

If we're trying to do this Planning Meeting in two hours, it should be clear that this meeting is merely the culmination of a series of planning activities, as getting enough information to make an informed commitment is much too complex to do in just two hours.

The Planning Meeting is a formal meeting where the team agrees upon, and commits to, a collection of stories and their "doneness" Agreements (see chapter xxx for further discussion of these Agreements). In order for this meeting to take place, there needs to have been frequent (or constant) Backlog Grooming in order to have stories that are ready to be planned with (see chapter xx on Backlog Grooming).

There are two common methods of coming to this commitment:

- Velocity-Based Planning, and
- Commitment-Based Planning.

In the following sections we give brief descriptions of each of these methods.

Velocity-Based planning

Velocity-Based Planning is the name I give to the "by the book" planning we find in the scrum literature. It was originally designed to be done by a Team that has an external Product Owner; that is, one that is not available to the team all the time. In the original scrum books, this process was intended to take a complete day, though most teams can get the process down to two hours through proper backlog grooming. The process is simple, and I'll just give an outline here:

1. Assume that the Team has an existing velocity (see chapter xx). The Product Owner will bring the team an appropriate number of stories (for example, 125% of the team velocity's worth) to work on.
2. The Product Owner and team will discuss these stories and arrive at a common agreement of what "done" means for each of them.

3. The Product Owner leaves, and the team tasks out each of these stories, and provides Task Hour estimates of effort for these tasks. With these hourly task effort estimations, the team can arrive at a Task Hour estimate for each story.
4. When the Product Owner returns, the Team and the Product Owner negotiate which of the Stories will actually be committed to, based on the Task Hour effort estimates for the stories.

The result of this planning is a collection of committed-to stories, with doneness agreements and tasks with Task hour estimates. There are likely to be a few uncommitted tasked-out stories remaining in the Backlog.

Commitment-Based Planning

Commitment-Based Planning is relatively new, but is gaining a lot of traction amongst scrum aficionados. One of the primary reasons we can do it with a modern scrum team is that the Product Owner is part of the team, and can be expected to be an active participant in Sprint Planning from beginning to end. As in Velocity-Based Planning, the success of Commitment-Based Planning is largely due to doing a significant amount of Backlog Grooming throughout the Sprint (see chapter xx). The Commitment-Based planning process is reasonably simple, and the next chapter will give a more complete description of it. For now, let's just have a simple outline:

1. It all starts with the Product Owner having a prioritized list of Stories, sufficient to fill the Sprint. Typically, based on StoryPoints and Velocity, you would need 125% of what you would expect to be a single sprint's worth of stories available for planning. This is no different than in Velocity-Based planning.
2. At the beginning of planning, the PO selects a single story to work with. The Team (including the Product Owner) negotiates the "doneness" Agreement for this single story, and the Team (without undue influence from the Product Owner, or consideration of the Story's size in SPs) evaluates and commits to this single story, if it can.
3. After a Story is committed to, or if a story is "skipped" because it can't be committed to, the PO gives the Team another story to consider. This is usually the next one on the list, but the order may be changed because of dependencies, or to try to "make it fit." Once again, the Team comes to the "doneness" Agreement and commits to adding the Story to the list of already-

committed-to Stories. This process is repeated until the Sprint is “full” and the Sprint Plan is complete.

This form of planning results in a list of committed-to stories with doneness agreements, and the way we think of it is that the doneness agreements are what the team has actually committed to. When done well, there should be no more than one extra story on the backlog that has a doneness agreement, and it is there only if the final story considered wouldn’t actually fit in the sprint.

Summary

It is possible, and desirable for most teams, to spend only one day between Sprints. In this day the Team does its Sprint Review with its Stakeholders, conducts its Retrospective, and does its next Sprint’s Planning. In addition, the Product Owner works with the Stakeholders to adjust the Release Plan.

In order to do all of this in one day, the team has to stay focused. It also must do considerable Backlog Grooming throughout the Sprint so that the Backlog is in a good state for planning.

Good Luck!