Managing Your Backlog

A brief guide for Scrum Product Owners

by Richard Lawrence

humanizing WORK

Last updated 12/10/2020

The Challenge of the Product Owner Role

For years, I've advised Scrum teams to limit how many stories they work on at one time. To limit work in progress, or WIP, in other words. Those teams who do so see improvements in focus, collaboration, self-organization, and productivity. Teams with a task board can see at a glance how many stories are open and can easily manage their WIP.

Product Owners have more difficulty managing WIP. On any given day, a PO is expected to:

- Discuss long-term product strategy with executives
- Discuss long- and medium-term feature road maps with key customers
- Refine stories for the next couple of sprints
- Answer questions from the team about current stories
- Work with the team to size upcoming stories
- Coordinate dependencies and related features with other POs and teams
- Review, provide feedback on, and accept recently completed stories

Most POs are overwhelmed. Few can honestly say they spend most of their time on the most valuable tasks. POs I talk with frequently find themselves mired in conversations about the details of features months ahead on the backlog and too busy to engage with the team on current priorities. Or, they're too busy working closely with the team on current stories that they lack the time to look ahead to maximize the value of their product.

To make matters worse, product ownership is often a role added on top of the regular job responsibilities of a line manager. If a full-time PO is overwhelmed, a part-time PO is buried without hope of digging out.

This book will help you structure and manage your backlog so you can be an effective—and sane—Product Owner.

Structuring the Product Backlog

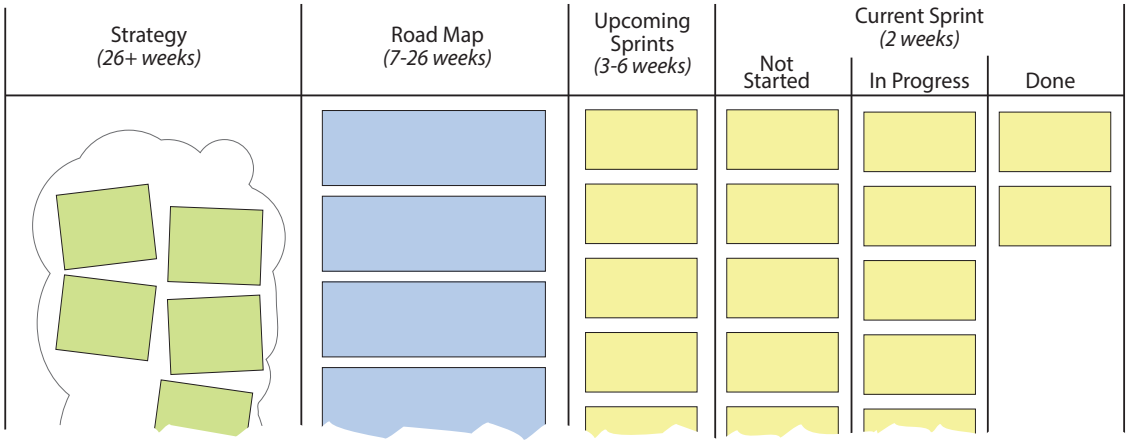
Like most Scrum trainers, I teach teams and POs that a good backlog is more detailed at the top than the bottom, requiring continuous grooming as backlog items make their way up.

The typical model for a backlog. Well intentioned, but too vague to be practical.

Unfortunately, this advice seems to be too vague to prevent detailed conversations about items well down the backlog. I've met far too many POs who heard this advice in training but are, after just a few months, buried under a backlog with 18 months of small stories.

To make the backlog structure more tangible, I now teach a model I call the "PO Board." The PO Board is analagous to a Scrum team's task board or Kanban board. It's a visual representation of the Product Backlog.

On the PO Board, work moves from left to right as it gets closer to done, with specific criteria to move from one column to the next and limits on how much work can be in each column. For the PO, this means backlog items become more detailed as they move from left to right. This limits the number of items that need discussion at a particular level of detail at any given time.



The PO Board divides the backlog into distinct sections, each with a particular level of detail.

Download a poster of the PO Board at http://www.agileforall.com/resources/po-board/

The Current Sprint

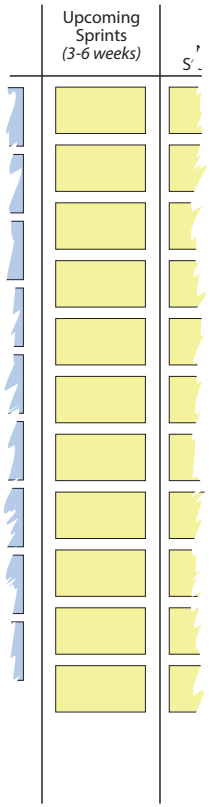
The three columns on the right of the PO Board represent user stories in the current sprint. At the end of Sprint Planning, all stories would be in Not Started. As work begins on a story, it moves into In Progress. When it's done, it moves to Done.

This is an overview of the current sprint, optimized for PO. A team probably has its own view of the work with more detail, such as tasks or scenarios. When multiple teams are working from a single backlog, each team probably has its own board, and the PO's board would reflect the combined stories across the teams.

Most of the PO's work for the stories in the current sprint relates to those stories in the In Progress column. The PO will be answering questions from the team, elaborating the details of the stories, and reviewed completed stories to accept them and move them into Done.

The illustration to the right assumes 2 week sprints. Replace this with your sprint length if it's different.

Upcoming Sprints



The next column to the left represents work likely to be in the upcoming sprints. These stories are being refined to be ready for Sprint Planning. Stories here ought to give the team enough detail to make decisions in the current sprint that leave options open for the upcoming items.

This column should be sized large enough to allow enough time to refine the stories before they make it into a sprint but small enough that the items and detail in them is stable and fresh. For most backlogs, two sprints ahead of the current sprint seems to be the sweet spot. (As in the previous section, the illustration assumes 2 week sprints. You can scale accordingly.)

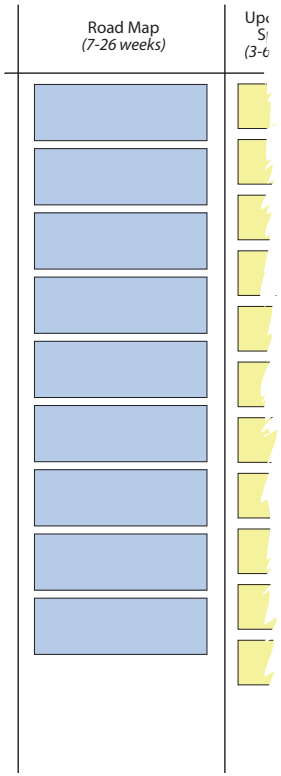
The PO's work for stories in this column is to get them ready for Sprint Planning. They should be split so that 6-10 would fit in a sprint. They should be sized by the team. They should have enough detail associated with them to be estimable and testable (but not so much detail that the team and PO miss an opportunity to learn as the software take shape during the sprint).

Refining stories in this column will require help from the team. By herself, a PO doesn't have all the information she needs to split a large story evenly. She needs to work with team members who understand what makes a story large or complex so that they can slice through that complexity. Similarly, only the team can estimate stories—that's not the PO's role.

This help tends to come from the team in one of two ways. Either they have a standing backlog grooming meeting or (my preference) the PO brings her questions to the Daily Scrum and the team self-organizes to give her the help she needs that day.

Road Map

For most backlogs, user stories are too small and detailed for the backlog beyond the next couple of sprints. It's hard to make sense of a backlog with dozens or hundreds of stories. And the stories are likely to change many times before they reach the top of the backlog. Therefore, something larger a user story is necessary to make the backlog more comprehensible and less volatile.



Beyond the next couple sprints and out to about six months, I like to see the backlog made up of Minimum Marketable Features (MMFs).

An MMF is a container for a set of stories that together make enough functionality to be worth releasing. Of course, beyond two sprints, I don't recommend that the actual stories exist yet, but when the time comes, an MMF will be decomposed into stories.

Because MMFs are minimal, every story in a particular MMF is essential. To be marketable most often—to maximize the value of the product—it makes sense to work MMFs one at a time in priority order.

The column to the left of the upcoming sprints, therefore, is a ranked list of MMFs. As space becomes available in the upcoming sprints column, the top MMFs will be decomposed into stories to fill that column.

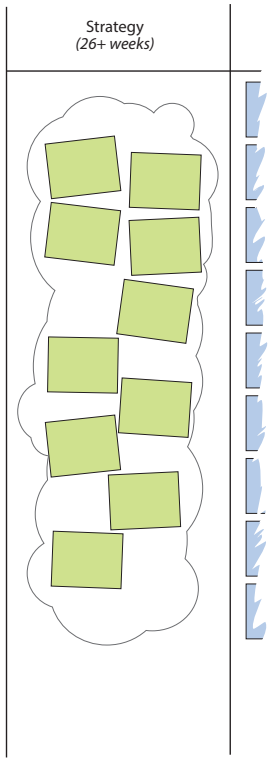
Just as stories can be sized relative to other stories, MMFs can be sized relative to other MMFs. Then, if you measure the relationship between story points and feature points, you can project a velocity in story points out over features, even though they don't yet have stories to size. (More on this in the Forecasting section later.)

Strategy

Beyond 6 months, even MMFs are too detailed to be stable on most backlogs I come across. Nonetheless, it can be useful to know the general direction of the product for the next year or two in order to make good short-term decisions that leave long-term options open.

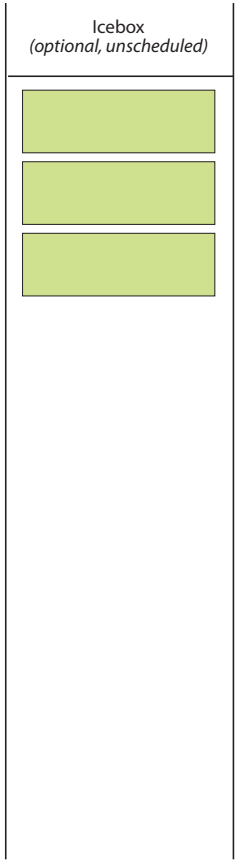
The strategy column contains general themes and features. They're not sized. They may not even be prioritized—we represent this section as a cloud of options rather than a ranked list, with the most likely options to be exercised closest to the top.

Quite often, MMFs will be created by pulling elements from multiple items near the top of the Strategy column instead of by decomposing a complete item into MMFs.



Icebox

The Icebox column is a place to capture ideas you're intentionally not working on. Backlogs have a tendency to collect every idea anybody has about what should be in your product. This makes the backlog unwiedly. Worse yet, backlog items function as implicit promises to build the thing at some point. An Icebox allows you to say, "Yes, we've heard that request, and we're deciding not to do it right now."



Forecasting

We've seen how a good structure for the backlog brings simplifies the job of refining the backlog. It also helps facilitate forecasting over the backlog.

As a Product Owner, you've probably faced questions like,

- "Will my feature make it in the June release?"
- "When will this collection of features be done?"
- "Can you squeeze this new feature into the next release?"

In order to answer these questions, it's necessary to be able to project time over the backlog with some level of confidence.

By sizing stories and observing the team's velocity, we can calculate how many sprints remain until a particular backlog item or, conversely, which backlog items fit into a fixed number of sprints. However, this assumes:

1. completed items really are complete (i.e. no work was left for later, such as regression testing),
2. the team's velocity will be stable, and
3. nothing is going to change on the backlog to move the items up or down.

Are you comfortable making those assumptions about your backlog and team? Enough to promise the bottom feature in a release to a key customer? I wouldn't be.

To be able to forecast beyond the next couple weeks, we need a way to account for risks such as instability in the team's velocity or backlog and items being less than perfectly done.

Let's look at one way to do this…

In *The Art of Agile Development*, Jim Shore and Shane Warden tackle this problem of accounting for risk on agile projects (p. 228). Based on empirical work from Tom DeMarco, Tim Lister, and Todd Little, they propose using the factors in the table to the right to adjust velocity when forecasting.

Confidence Level	Rigorous	Risky
10%	1	1
50%	1.4	2
90%	1.8	4

The first step is to determine which risk profile you have. If your backlog items are fully complete in each sprint, your velocity is relatively constant, and your backlog is fairly stable, you're "Rigorous." Otherwise, you're "Risky."

Next, divide your velocity by the factors in the appropriate column. You'll use these adjusted velocities to forecast over your backlog for different confidence levels.

For example, suppose you have the "Risky" profile, a historical velocity of 40 points per sprint, and 3 sprints remaining in your current release. You'll get the adjusted velocities and points remaining as shown in the table on the right.

Confidence Level	Risk Factors	Adjusted Velocity	Points Remaining
10%	1	40	120
50%	2	20	60
90%	4	10	30

You can say with high confidence that you'll deliver the stories in the next 30 points. There's a 50-50 chance you'll get the stories between 30 and 60 points. You might get the stories between 60 and 120 points down the backlog. Beyond 120 points is very unlikely.

Depending on who's asking, you might communicate with this nuance. Or you might just avoid committing to stories beyond 30 points when talking with key customers.

Does this mean the team is only going to deliver 10 points in the next 3 sprints? Of course not. But things happen. The team might hit an unexpected road block. Certain stories might take longer than expected. Backlog items might change. These factors incorporate a variety of risks.

In a typical backlog, only stories get sized. Thus, we can only forecast as far as we have sized stories. Since we're often asked to forecast more than a couple sprints into the future, this leads to waste as we plan in too much detail too far out.

As I mentioned earlier, we can size MMFs relative to other MMFs just as we size stories relative to stories. We can measure the relationship between MMF points and story points. For example, we might observe than a 1 feature point MMF tends to become about 8 story points and a 2 feature point MMF becomes about 16 story points. This allows us to convert feature points to story points in the Road Map column of PO Board in order to project the adjusted velocities beyond 2 sprints without having to know all the stories.