


# Refine your backlog

18/11/2018 • 4 minutos para ler • Colaboradores 

## Neste artigo

[Role of the product owner](#)

[Acceptance criteria](#)

[Refine your backlog](#)

[Capture and manage spikes](#)

[Additional resources](#)

**Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013**

A great backlog conveys customer needs and value. Over the course of the project, your team will add detailed information to each backlog item, break them down into smaller items, prioritize and estimate them, and finally, implement them and deliver the results to your customers.

To get started, see [Create your backlog](#).

## Role of the product owner

Product owners play an important role in Scrum, primarily as the interface between customers and the team. To enable product owners to perform the following responsibilities, they need to be added to the Contributors group.

- Analyzing customer requirements and articulate them as user stories, features, or requirements
- Building, prioritizing, and refining the product backlog
- Representing customer and stakeholder requirements to the team and responding to questions your team has about them
- Meeting regularly with stakeholders to address their needs and keep them informed
- Helping stakeholders understand the decisions underlying the priority order of your backlog
- Responding to any and all requests from your team for more information concerning backlog priorities and requirements

If they will also be responsible for configuring team settings, [add them as a team administrator](#).

A product owner can reduce the need for detailed specifications by being more responsive to the team's questions about implementation details and clearly articulating acceptance criteria within each requirement.

## Acceptance criteria

Acceptance criteria define what "Done" means by describing the conditions that the team should use to verify whether a requirement or bug fix has been fully implemented. You can capture these criteria in the work item. Clear acceptance criteria help with estimating and developing requirements and with testing.

Product owners are the ultimate deciders of the criteria that create customer value.

**Tips from the trenches: Start to love and embrace acceptance criteria.**

Ask 10 mature agile teams "How do you know when you're "done done"? and you'll get the same answer from each one. . . get serious about writing acceptance criteria.

Acceptance criteria are the handshake between the product owner and the team on what "done" really means.

Until the acceptance criteria are met, the team isn't done with the story. Period. However, the value of acceptance criteria only starts here.

Acceptance criteria provide the stage for some of most meaningful conversations and interactions that can happen on an agile team. On my own team we routinely have some of our best interactions as we start digging into the acceptance criteria for each story on our backlog. Inevitably we all start with our own ideas about what "done" means for a given story.

However, as we begin to discuss the acceptance criteria presented by the product owner what ensues is a series of "ah-ha moments." A shared understanding of the story begins to emerge. A comment one team member might elicit the following response from someone else. . . "Ah-ha, great point. . . I never thought of that."

Regardless of who is being enlightened, the power is in the fact that the product owner and the team are building together a shared understanding of what "done" means for each backlog item. And, this is happening before the team has written a single line of code... before any work has been done... before commitments have been made... and before the sprint has begun.

By collaborating on acceptance criteria the team is minimizing risk and greatly increasing the chance of delivering successfully. I don't think it's a coincidence that the first bullet in the [Agile Manifesto](#) states ". . . we have come to value **individual and interactions** over processes and tools". Agile teams work together. And by working together, they create better software.

Start learning to love acceptance criteria and see if your team isn't more successful delivering software.

—Aaron Bjork, Principal Product Manager, Visual Studio Cloud Services

## Refine your backlog

Backlog refinement supports your sprint planning efforts and helps minimize these often seen challenges:

- Long, unfocused, and ineffective sprint planning meetings
- Insufficient thought given to design requirements
- Poor sprint planning and execution
- Defocus on the business value team wants to achieve
- Inability to forecast

A meeting to refine the backlog should occur separate from the sprint planning meeting. Use this meeting to perform these activities:

- Right-size backlog items by splitting larger items into smaller items. No backlog item should be larger than it will take to complete in a single sprint.
- Identify and fill in gaps in the product backlog. Capture new ideas and stories, architecture and design requirements, and other spikes.
- Reorder the backlog to represent today's priorities and business value focus.
- Ensure well defined acceptance criteria has been added to each item.
- Revisit estimates made to backlog items and adjust upwards or downwards based on recent understanding about scope and acceptance criteria.
- Review all potential backlog items to consider for the upcoming sprint to make sure they are well understood and that any additional work required to support their development is well understood by both product owner and the team.

You'll know that you've done a good job refining your backlog when your sprint planning meetings run smoothly and efficiently. Such meetings shouldn't contain a lot of surprises, and your team should feel they can contribute fully.

[Refining your Agile backlogs for success](#) provides a nice quality checklist to guide your backlog refinement efforts

## Capture and manage spikes

In addition to new features and requirements to build, you can capture non-feature work that still needs to be done for a healthy ecosystem of delivery. This work can include necessary research, design, exploration, or prototyping. Any work done that doesn't directly lead to shippable software can be considered and captured as a spike.

As the need to perform this work arises, capture it along with other items on your backlog. To track that it is a spike, you can either preface the title with the word "[Spike]" or add the tag "Spike" to the work item.

## Additional resources

- [What is Agile?](#)
- [Building productive, customer focused teams](#)