Microsoft

Professional Scrum Development with Azure DevOps

Professional

Richard Hundhausen

Foreword by Ken Schwaber, Co-creator of Scrum

Praise for this book

"Scrum is described as taking 10 minutes to learn and a lifetime to master. In this book, Richard provides tips and tricks to mastering Scrum. He marries the practical with the abstract, providing a foundation of learning that helps Developers deliver high-value products and solve complex problems. If you are using Azure DevOps and want to get better at doing it, then this is the book for you."

—Dave West, Scrum.org Product Owner and CEO

"Like it or not, many teams need tooling to help them with their Scrum implementation. That's where Richard comes in. His knowledge and passion shine through in all that he touches—especially in this essential guide for how to use Azure DevOps for Scrum Teams. If you know anything about Richard, and you are using Azure DevOps with Scrum, then you'll know this book is a must-read."

—Daniel Vacanti, Co-founder, ActionableAgile

"In this book, Richard Hundhausen does a great job explaining and connecting the domains of Professional Scrum with professional development using Microsoft Azure DevOps. Richard introduces the history and current state in both domains and makes the book even richer with personal tips and illustrations through case studies."

—Gunther Verheyen, independent Scrum Caretaker, Professional Scrum Trainer

"Scrum is a framework that is easy to understand but difficult to master. Richard takes the difficult out of the equation for you. What sets him apart from all others is his ability to help others not only understand Scrum, but become masters at it."

—Chris Roan, Wells Fargo Agile Transformation Leader

"If you're working on a Scrum team, do yourself a favor and read this book. In it, Richard distills his many years of practical experience leading Scrum teams in order to help you and your team accelerate your DevOps transformation. If you want to deliver more customer value at higher velocity, there's no better place to start."

—Jeff Beehler, Senior Director, Product Operations, GitHub, Inc.

"During my time on the Azure DevOps team, I became aware of Richard's passion for Professional Scrum and his desire for us to build the tool in a way that Scrum Teams would love it. The essence of DevOps is to get a right blend of processes, tools, and people working seamlessly to deliver customer value. Combine that with Scrum and

Next, I make changes to the workflow states by adding two new states: Ready (which maps to the Proposed category) and Forecasted (which maps to the In Progress category). I keep the default colors for these new states. Next, I hide the Approved and Committed states, replacing them with the Ready and Forecasted states that I just created, as you can see in Figure 3-17.

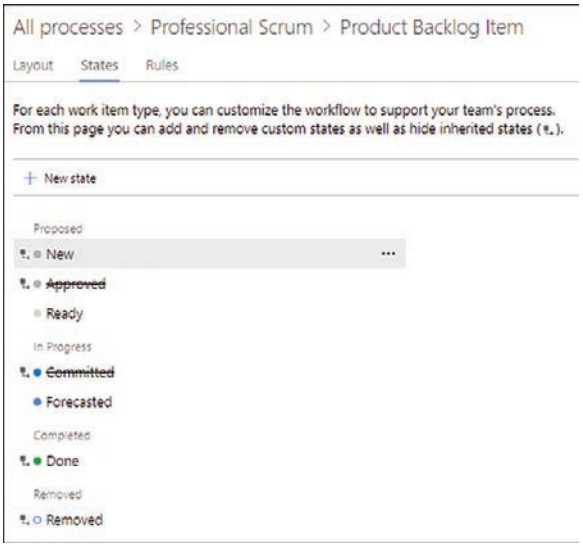

FIGURE 3-17 Customizing PBI workflow states in the Professional Scrum process.

For organizations and teams that use the Epic and Feature work item types, you can make similar customizations by hiding those extraneous fields that you don't use (e.g., *Priority*, *Time Criticality*, *Value Area*, and even *Start Date* and *Target Date*). You could also rename the labels and normalize the workflow states, as I did for the PBI work item type.

I also hide the *Priority* and *Activity* fields from the Task work item type. The last customization I do is to rename the lowest leaf-level backlog from "Backlog items" to "Stories" (or whatever the organization/team would like it to be called). Leaving it named *Backlog items* is confusing, because in actuality, all backlog levels contain "backlog items."

After these changes are made, I can start creating projects based on the Professional Scrum process. If I have any existing projects, I can also change them to use the new Professional Scrum process, which you can see in Figure 3-18. Later, if I make any changes to the Professional Scrum process, all projects based on it will instantly reflect that change.

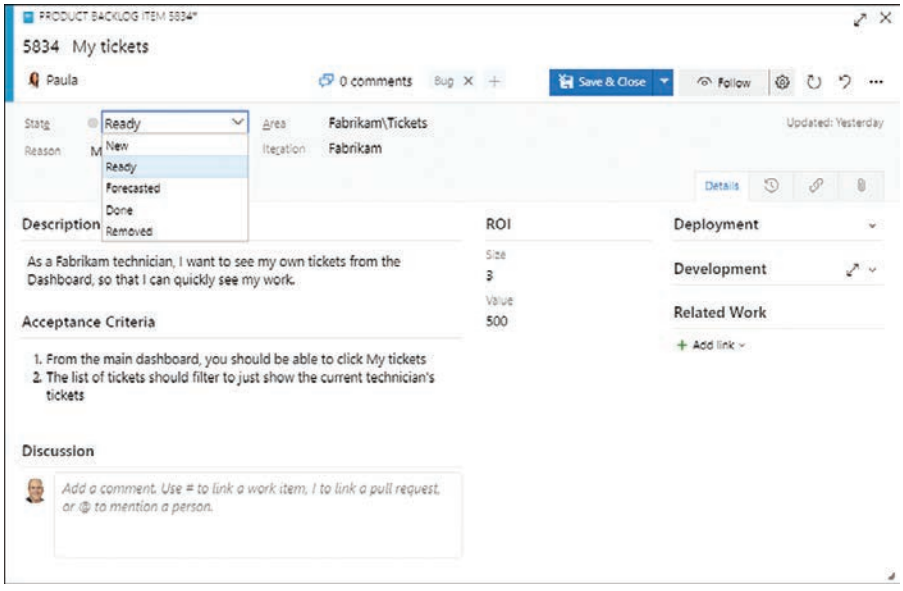

FIGURE 3-18 PBI work item form after applying the Professional Scrum process.

Other Customizations

Beyond matching the Scrum process to the *Scrum Guide*, organizations and teams may want to make additional customizations to work items and backlogs. Here are some examples I've collected over the years from various teams and other consultants:

- Add a *Team* field to the PBI work item type to indicate which team owns it, rather than using the *Area* field gimmick.
- Add a *Value Stream* work item type and corresponding top-level backlog that sits above Epic.
- Add a *Planned Sprint* field to the Impediment work item type for planning when the improvement will be done. The system *Iteration* field can be used for the Sprint where the impediment was discovered.
- Add a new workflow state to the Impediment work item type to indicate which improvement(s) are currently *in-progress*.
- Add an *Improvement* work item type to plan and track any improvement experiments being performed.
- Add a *Remaining Work* field to the PBI work item type to store the rolled-up sum of any child task *Remaining Work* values. External automation, such as an extension, would be required to do the rolling up.
- Add default user story description text ("As a <*type of user*>, I want <*some goal*> so that <*some reason*>") to the *Description* field of the PBI work item type.

- Add default text to the PBI work item type *Acceptance Criteria* fields to suggest a "given-when-then" behavior-driven development (BDD) or "given-when-then-fail" format.
- Add a *Hypothesis* work item type to support hypothesis-driven development.

Tip Use your Scrum or Professional Scrum process the way that it was designed for a few Sprints before customizing anything. I've seen teams want to immediately make their new project look and behave like their old project or culture. For example, if after reading this book you decide to abandon your Agile process project, creating a new Professional Scrum project, don't immediately add the fields that used to be in your old project (for example, the *Original Estimate* and *Completed (hours)* fields in the Task work item type—which we removed from the Scrum process for a reason; tracking original estimates and actual hours are generally considered waste). Just know what you are doing and why you are doing it before making any "improvements" and weigh the benefits against potential waste and misuse. Don't inadvertently change the rules of Scrum by customizing the tool!

Fabrikam Fiber Case Study

The Scrum Team has decided to follow the guidance in this section and create and use the Professional Scrum process, inherited from the Scrum process. They will apply this new process to their existing Fabrikam project. This will be the process that is referenced in the coming chapters. For that reason, you may want to take a moment and create a Professional Scrum process yourself so that you can better follow along.

Chapter Retrospective

Here are the key concepts I covered in this chapter:

- **Process** When creating a project, you will need to select a process. Microsoft provides several out-of-the-box processes, referred to as system processes.
- **Scrum process** A Scrum-centric process created through a collaboration of Microsoft and the Professional Scrum community.
- **Work item types** Although there are more than a dozen Azure DevOps work item types, including a number of hidden ones, the ones that apply to planning and managing work are Product Backlog Item, Bug, Epic, Feature, Task, and Impediment. Task and Test Case work items should be created and linked only during the Sprint in which you are working on their parent PBIs.

- **Queries** There are a number of queries that a Scrum Team could create and share to track and manage the work in the Scrum development effort.
- **Scrum Guide drift** Over the years, the *Scrum Guide* has evolved while the Scrum process has not. This issue can be overcome by creating and customizing an inherited process.
- **Inherited process** A child of one of the system processes that can be customized in a structured way, inherited processes can be used to create new projects as well as applied to existing projects. Future changes to the inherited process are instantly visible in all the projects that use that process.

This page intentionally left blank

PART II

Practicing Professional Scrum

In this part of the book, I will begin demonstrating how to practice Professional Scrum and Azure DevOps together effectively. The previous part established a baseline understanding of the three areas of knowledge required before proceeding: Scrum, Azure DevOps, and Azure Boards. Over the next several chapters, you will see how these three fit together and how a team can optimize their use to deliver business value in the form of working software.

I will begin with the discussion and activities surrounding product planning. This will take us up to the beginning of the first Sprint. I refer to this collection of activities as the *pre-game*—everything from envisioning the product, provisioning the Azure DevOps environment, setting up the project, organizing the team, building and refining the Product Backlog, and preparing for the first Sprint falls into the pre-game. As you can imagine,