

Managing Scope and Procurement

- » Finding out how scope management is different on agile projects
- » Managing scope and scope changes with agile processes
- » Seeing the different approach agile processes bring to procurement
- » Managing procurement on agile projects

TABLE 12-1

Traditional versus Agile Scope Management

Scope Management with Traditional Approaches	Scope Management with Agile Approaches
Project teams attempt to identify and document complete scope at the beginning of the project, when the teams are the least informed about the product.	The product owner gathers high-level requirements at the beginning of the project, breaking down and further detailing requirements that are going to be implemented in the immediate future. Requirements are gathered and refined throughout the project as the team's knowledge of customer needs and project realities grows.
Organizations view scope change after the requirements phase is complete as negative.	Organizations view change as a positive way to improve a product as the project progresses. Changes late in the project, when you know the most about the product, are often the most valuable changes.
Project managers rigidly control and discourage changes after stakeholders sign off on requirements.	Change management is an inherent part of agile processes. You assess scope and have an opportunity to include new requirements with every sprint. The product owner determines the value and priority of new requirements and adds those requirements to the product backlog.
The cost of change increases over time, while the ability to make changes decreases.	You fix resources and schedule initially. New features with high priority don't necessarily cause budget or schedule slip; they simply push out the lowest-priority features. Iterative development allows for changes with each new sprint.
Projects often include <i>scope bloat</i> , unnecessary product features included out of fear of mid-project change.	The scrum team determines scope by considering which features directly support the product vision, the release goal, and the sprint goal. The development team creates the most valuable features first to guarantee their inclusion and to ship those features as soon as possible. Less valuable features might never be created, which may be acceptable to the business and the customer after they have the highest-value features.



Managing Agile Scope

- »» The product owner ensures that the rest of the project team — the scrum team plus the project stakeholders — clearly understands the existing scope for the project, the product vision, the current release goal, and the current sprint goal.
- »» The product owner determines the value and priority of new requirements in relation to the product vision, release goal, sprint goals, and existing requirements.
- »» The development team creates product requirements in order of priority to release the most important parts of the product first.

Introducing scope changes

- » User community feedback, including groups or people who are given an opportunity to preview the product
- » Business stakeholders who see a new market opportunity or threat
- » Executives and senior managers who have insight into long-term organizational strategies and changes
- » The development team, which is learning more about the product every day, and is closest to the working product
- » The scrum master, who may find an opportunity while working with external departments or clearing development team roadblocks
- » The product owner, who often knows the most about the product and the stakeholders' needs

Managing scope changes



Managing scope changes

When you get new requirements, use the following steps to evaluate and prioritize the requirements and update the product backlog.

1. Assess whether the new requirement should be part of the product, the release, or the sprint by asking some key questions about the requirement:

- a. *Does the new requirement support the product vision statement?***
 - If yes, add the requirement to the product backlog and product roadmap.
 - If no, the requirement shouldn't be part of the project. It may be a good candidate for a separate project.

b. *If the new requirement supports the product vision, does the new requirement support the current release goal?*

- If yes, the requirement is a candidate for the current release plan.
- If no, leave the requirement on the product backlog for a future release.

c. *If the new requirement supports the release goal, does the new requirement support the current sprint goal?*

- If yes and if the sprint has not started, the requirement is a candidate for the current sprint backlog.
- If no or the sprint has already started or both, leave the requirement on the product backlog for a future sprint.

2. Estimate the effort for the new requirement.

The development team estimates the effort. Find out how to estimate requirements in Chapter 7.

3. Prioritize the requirement against other requirements in the product backlog and add the new requirement to the product backlog, in order of priority.

Remaining Product Backlog

New →

Product
Budget →

User Story

User Story

User Story

User Story

User Story

User Story

User Story

User Story

FIGURE 12-2:
Adding a new
requirement to
the product
backlog.

Using agile artifacts for scope management

TABLE 12-2

Agile Artifacts and Scope Management Roles

Artifact	Role in Establishing Scope	Role in Scope Change
Vision statement: A definition of the product's end goal. Chapter 7 has more about the vision statement.	Use the vision statement as a benchmark to judge whether features belong in the scope for the current project.	When someone introduces new requirements, those requirements must support the product vision statement.
Product roadmap: A holistic view of product features that create the product vision. Chapter 7 has more about the product roadmap.	Product scope is part of the product roadmap. Requirements at a feature level are good for business conversations about what it means to realize the product vision.	Update the product roadmap as new requirements arise. The product roadmap provides visual communication of the new feature's inclusion in the project.

TABLE 12-2 (continued)

Artifact	Role in Establishing Scope	Role in Scope Change
<p>Release plan: A digestible mid-term target focused around a minimum set of marketable features. Chapter 8 has more about the release plan.</p>	<p>The release plan shows the scope of the current release. You may want to plan your releases by themes — logical groups of requirements.</p>	<p>Add new features that belong in the current release to the release plan. If the new user story doesn't belong in the current release, leave it on the product backlog for a future release.</p>
<p>Product backlog: A complete list of all known scope for the product. Chapters 7 and 8 offer more about the product backlog.</p>	<p>If a requirement is in the scope of the product vision, it is part of the product backlog.</p>	<p>The product backlog contains all scope changes. New, high-priority features push lower-priority features down on the product backlog.</p>
<p>Sprint backlog: The user stories and tasks in the scope of the current sprint. Chapter 8 has more about the sprint backlog.</p>	<p>The sprint backlog contains the user stories that are in scope for the current sprint.</p>	<p>The sprint backlog establishes what is allowed in the sprint. After the development team commits to the sprint goal in the sprint-planning meeting, only the development team can modify the sprint backlog.</p>