Process 515: Develop a Product Backlog

Last Updated: 10/28/2016

Audience:	Frequency:
CWDS Service Delivery Teams	Within 30 days of your service team kickoff

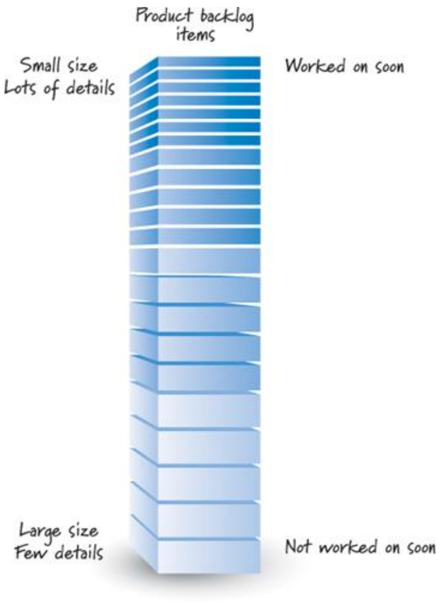
After reading this procedure, the audience will be able to understand how to develop a product backlog that best reflects the overarching scope of the service team.

Overview:

The **product backlog** in Scrum is a prioritized features list, containing short descriptions of all functionality desired in the product. It's not necessary to start a project with a lengthy, upfront effort to document all requirements. The product backlog is really the core deliverable that maintains and evolves the requirements in an agile environment.

Typically, a service team and its service manager begin by writing down everything they can think of for agile backlog prioritization. This agile product backlog is almost always more than enough for a first sprint. The Scrum product backlog is then allowed to grow and change as more is learned about the product and its customers.

A product backlog contains a complete list of all requirements under consideration, written using a user story syntax and rank ordered. A typical backlog comprises the following different types of items:



1. **Features** (i.e. "As a shopper, I can review the items in my shopping cart before checking out so that I can see what I've already selected.")
2. **Bugs** (i.e. a defect)
3. **Technical work** (i.e. "Upgrade all developers' workstations to Windows 7.")
4. **Knowledge acquisition** (i.e. "Research various JavaScript libraries and make a selection.")

Some of the global benefits of Agile that apply to our effort are:

1. Quicker time to market - effort is quicker to start and yield an outcome
2. Flexibility (change, learning, customer satisfaction) – change is a part of what we do every day as well as getting customer feedback.

We want the product backlog to support both of the above concepts. A product backlog is a long list of the stuff we want to "build" – a set of features. Each "thing" in the product backlog is a "product backlog item" or PBI, best expressed as a user story. The product backlog replaces the old concept of the Requirements Specification in waterfall. A well-formed product backlog adheres to some basic rules:

1. It is a single listing (in priority order) of what you want to do
2. It is the right size for our team.

Developing User Stories:

The easiest way for a service team to express features on the agile product backlog is in the form of user stories, which are short, simple descriptions of the desired functionality told from perspective of the user.

The service manager shows up at the sprint planning meeting with the prioritized agile product backlog and describes the top items to the team. The team then determines which items they can complete during the coming sprint. The team then moves items from the product backlog to the sprint backlog. In doing so, they expand the product backlog item into one or more sprint backlog tasks so they can more effectively share work during the sprint.

The appropriate user story format is:

As a (who needs this?)

I want (what is needed?)

So that (what is the value?)

The INVEST Concept:

We need to **INVEST** for a well formed backlog item by following the INVEST model (http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/). The INVEST mnemonic for agile software projects was created as a reminder of the characteristics of a good quality Product Backlog Item (PBI), commonly written in user story format.

Mnemonic	Value	Description
I	Independent	We want to be able to develop in any sequence.
N	Negotiable	Avoid too much detail; keep them flexible so the team can adjust how much of the story to implement
V	Valuable	Users or customers get some value from the story

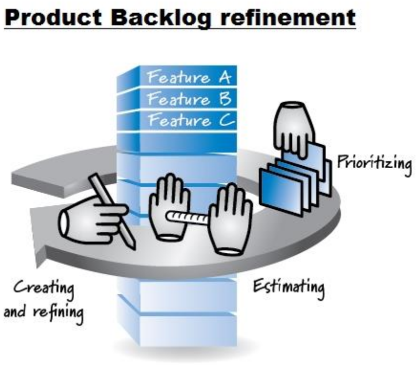
Mnemonic	Value	Description
E	Estimable	The team must be able to use them for planning.
S	Small	Large stories are harder to estimate and plan. By the time of iteration planning, the story should be able to be designed, coded, and tested within the iteration.
T	Testable	Document acceptance criteria, or the definition of done for the story, which lead to test cases.

The service manager needs input from the team and stakeholders so that everyone can "buy in" to the final product. The feedback loop within each sprint construct constantly refines the product backlog and re-emphasizes that what we are doing is meeting with customer needs.

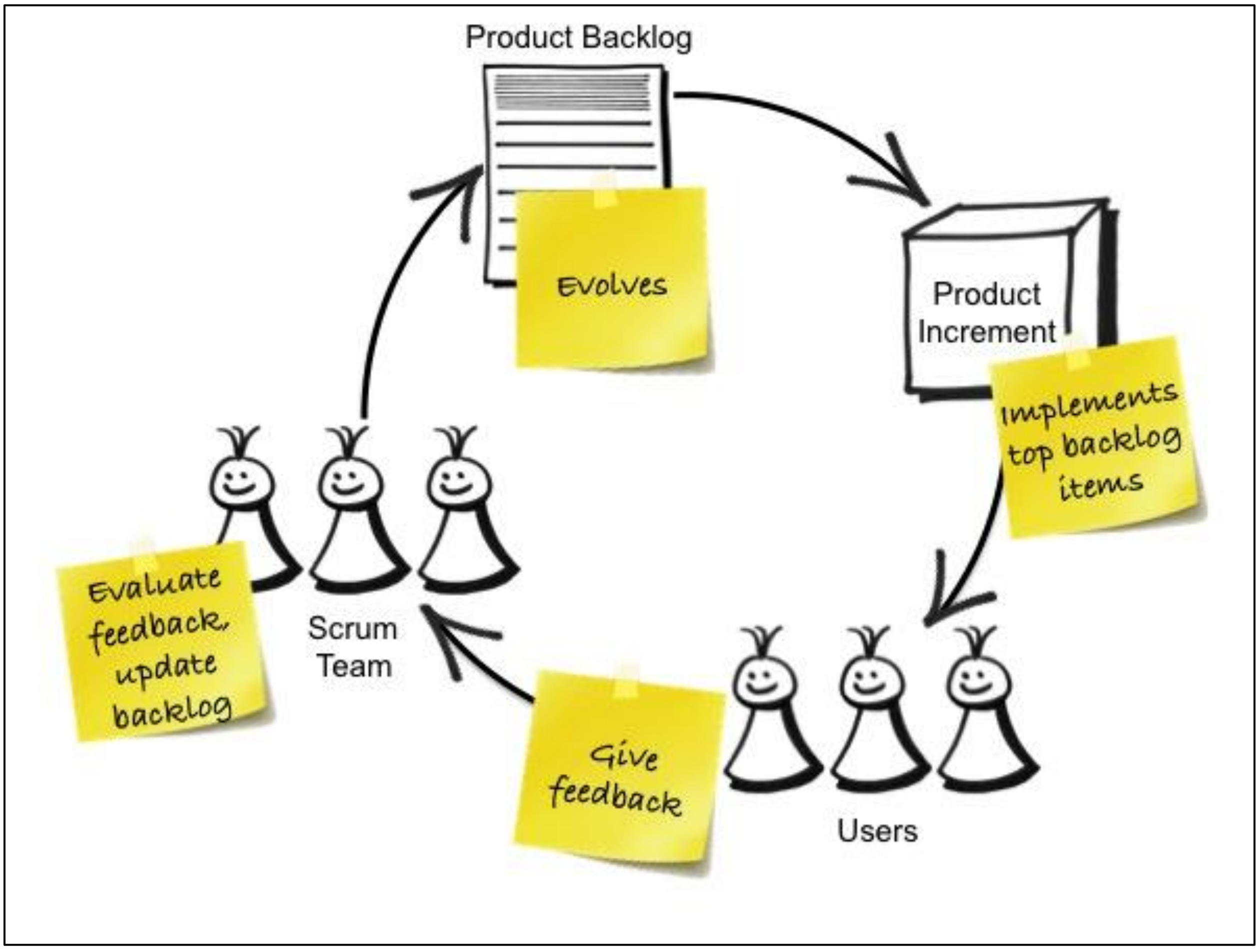
Start developing the Product Backlog by identifying the high-level Epics – the main features we want to develop that will be outputs to the customers. Everything we deliver needs to have value to the customer!

Approach to Develop a Product Backlog (including background requirements definition):

Step	Procedure
1.	**Develop a scope statement of a high-level feature list.** A business analyst or business SME creates a fairly traditional scope statement and features list. The features list is fundamentally business-driven.
2.	**Review feature list with development team.** At a kick-off with the development team, the service team reviews the draft of the feature list and starts discussions around detailed requirements communication.
3.	**Develop the Product Backlog.** Based on the feature list, items on the backlog should ideally be expressed in business terms that are of some value to the user (or customer, or business). Not expressed as technical tasks. A team's feature list (roadmap) and requirements provide the foundation for the product backlog. Roadmap initiatives break down into several epics, and each epic will have several requirements and user stories. The team will identify user stories that align with the features list. Identifying user stories blends elements of both requirements and design.

Step	Procedure
4.	**Work through the Backlog.** Functional requirements should be expressed as features. Non-functional requirements can be put on the Backlog too (i.e. 'the product needs to be faster', 'we need to ensure the product is secure', 'we need to get off the old platform', 'there's a high risk of downtime due to a single point of failure'.) These might not be features, as such, but they are completely justified as items on the Product Backlog. Driven by your understanding of what the business wants, and keep asking the question of, "How will it make sense to build this in a delivery cycle?"
5.	**Prioritize the Backlog.** The service manager prioritizes the user stories in rank order. The priority is determined simply by the order of the list. The most important items are shown at the top of the product backlog so the team knows what to deliver first. Things towards the top of the Product Backlog may be done in the foreseeable future. Therefore they are likely to be better understood. And they need to be well enough defined that the team could work on them. These features (or Backlog items) should be defined individually, so they can stand alone as discrete, deliverable pieces of work. Things at the bottom of the list may be way off and may or may not ever get done. Things down the bottom are likely to be fuzzy and ill-defined. Don't waste time defining things you may never get to, or not get to for some time. The development team doesn't work through the backlog at the service manager's pace and the service manager isn't pushing work to the development team. Instead, the development team pulls work from the product backlog as there is capacity for it, either continually (kanban) or by iteration (scrum)
6.	**Refine (Manage) the Backlog.** Over the course of the project, the product backlog continues to evolve and is "refined". Service Managers should review the backlog before each sprint planning meeting to ensure prioritization is correct and feedback from the last iteration has been incorporated. Regular review of the backlog is often called "backlog refinement" in agile circles.  Product Backlog refinement Feature A Feature B Feature C Prioritizing Creating and refining Estimating As a service team drills into the requirements behind some of the earlier user stories, they may discover they were bigger than anticipated. In that case,

Step	Procedure
	break apart user stories into two or more product backlog items, or on occasion, combine stories. Once the backlog gets larger, service managers need to group the backlog into near-term and long-term items. Near-term items need to be fully fleshed out before they are labeled as such. This means complete user stories have been drawn up, collaboration with design and development has been sorted out, and estimates from development have been made. Longer term items can remain a bit vague, though it's a good idea to get a rough estimate from the development team to help prioritize them. The key word here is "rough": estimates will change once the team fully understands and begins work on those longer term items.


Product Backlog
Evolves
Product Increment
Implements top backlog items
Users
Give feedback
Scrum Team
Evaluate feedback, update backlog