



# Agile Record

The Magazine for Agile Developers and Agile Testers



Requirements in Agile Projects

[www.agilerecord.com](http://www.agilerecord.com)

free digital version

made in Germany

ISSN 2191-1320

August 2012

**issue 11**



# Rope in Value with User Stories

by Allison Pollard

## Lasso Value in the Product Backlog

In traditional projects, a large amount of effort was focused on documenting and understanding upfront the comprehensive requirements of a project in order to deliver the product. Experience tells us that this approach does not work – it is impossible to predict and document the requirements of complex software development projects. For this reason, the Agile Manifesto includes the following principle:

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Requirements can be like animals – difficult to manage in numbers. After transitioning to Agile, is your team focused on adding value to the product, or are you trying to deliver the project requirements (changes and all) faster? When driving cattle, cowboys had to strike a balance between speed and the weight of the cattle, which equated to their value. Similarly, those writing agile requirements need to balance anticipation and adaptation<sup>1</sup> in order for teams to deliver valuable product.

## Managing the Herd

In Scrum, the Product Owner role is the single voice that drives the delivery team, and it is his responsibility to communicate requirements to the team throughout the project. The Product Owner populates the Product Backlog with the requirements and prioritizes them. He has the hardest role because he is responsible for “maximizing the value of the product and the work of the Development Team.”<sup>2</sup> Because the Product Backlog reflects the work of the team, developers and testers should be seriously interested in it.

In my experience, the backlog can easily become quite large – if each item is written on an index card, the backlog can easily cover the walls of a conference room. Suddenly it can seem like the product backlog is managing the product owner instead of the other way around because of its overwhelming size. In fact, the bottom of the backlog may never change because low priority items stay low priority, and large items may not get the attention needed to become right-sized for sprints. Sprint-sized items are delivered by the team, but do they reflect the most valuable work they could've done? If this product backlog is a herd of animals, it is too large for one cowboy to manage, some of the animals are weak, and the strongest animals might be in danger of wandering off from the herd. Teams and Scrum Masters may need to help the Product Owner recognize the state of the Product Backlog and improve it.

Odds are that such a large product backlog is not really focused on value – it is full of ideas and requests that are anticipating users' needs instead of remaining adaptable. We know from numerous studies that the biggest source of waste in software delivery is rarely or never used functionality; a backlog containing every idea and request can lead to making sure everything is implemented rather than asking “can we release what is implemented now?” The Product Owner must know what creates value – what functionality leads to a product that can be released now – and be aggressive in grooming the backlog to keep it focused on value.

Evaluate the herd. Does the team have a clear picture of the animals closest to them? The items at the top of the backlog should be sufficiently well-understood and estimated so they can be brought into sprints and the Product Owner can effectively plan releases. If team members are unclear on the backlog items, they should collaborate with the Product Owner to make them better. The backlog items that never move from the bottom are weak animals; can they be made stronger and more valuable? If not, remove them. Are the large animals in danger of getting lost or wandering off? Large backlog items lose value if they are not broken down in time for the team to pull into a sprint and deliver. Look at the

1 Jim Highsmith, Agile Software Development Ecosystems

2 Jeff Sutherland and Ken Schwaber, The Scrum Guide



large backlog items for any that might be time-sensitive and work with the Product Owner to ensure the order of the backlog is such that it can be delivered when it is needed by forecasting based on item estimates and the team's velocity. The result of evaluating the herd should be a product backlog that is DEEP<sup>3</sup>: detailed appropriately, estimated, emergent, and prioritized.

### Caring for the Animals

Often the Product Owner populates the product backlog with user stories. According to Ron Jeffries, user stories have three critical aspects: card, conversation, and confirmation. The card contains just enough text to identify the requirement and remind everyone what the story is, the requirement itself is communicated through conversation, and the confirmation tells the team how the Product Owner will confirm that they've done what is needed. By creating the card, the Product Owner is able to catch the requirement animal in the lasso's noose, but it is the conversation with the team that tightens the rope. The conversation drives the value of the increment that will be delivered, so it is important to have a well-written card to start the conversation that will ensure the team is delivering the right requirement.

User stories are often written using the format made popular by Mike Cohn: *As a <type of user>, I want <some goal> so that <some reason>*. These stories can make for fine cattle in the Product Backlog herd, but in the quest to deliver value, it can be beneficial to use a different story format to emphasize it: *In order to <achieve some value>, as a <type of user>, I want <some functionality>*.<sup>4</sup> By moving value to the beginning of the user story, Product Owners and teams are more likely to answer the question, "How can I make this story more valuable?" Well-understood user stories lead to better decisions by team members because they can balance what is best for the user and what is easiest to implement.

A good user story meets the INVEST criteria<sup>5</sup>: it is Independent, Negotiable, Valuable, Estimable, Sized appropriately, and Testable. Just as the herd is composed of individual cows or steers, the backlog should be composed of stories that are self-contained so they are not dependent on other user stories. Stories should leave some flexibility for team members and Product Owners to later flesh out details and reflect value to users or customers. In order to plan effectively, stories need to be estimable; stories that are sized appropriately small enough can be worked by the team within a sprint, which helps teams validate that work is getting done and receive feedback on the product early to confirm value is being delivered. Appropriately sized stories are also easier to integrate, test, and deploy.

Team members should review the story to ensure it is testable and the confirmations to understand how the Product Owner will determine the story is done. The confirmations (also known as

acceptance criteria or conditions of satisfaction) can be written in a number of ways, but one common way is: Given [initial context], when [event occurs], then [ensure some outcomes]<sup>6</sup>. Developers and testers can maximize value now and in the future by automating tests that show the confirmations have been met. By knowing how the animal will be evaluated at the end of the drive, the cowboys can better ensure its delivery will be as valuable as possible.

### The End of the Drive

Agile projects emphasize working software over comprehensive documentation but can fall victim to delivering lesser valuable product increments due to large, hard to manage Product Backlogs and poorly written user stories. Product Owners and teams need to collaborate and use the Product Backlog as a powerful tool to drive the team's work and forecast releases. Just as cattle are evaluated based on the quality and quantity of beef they provide, agile requirements should be evaluated based on the quality of software they produce.

#### > About the author



#### Allison Pollard

is a Senior Consultant for Improving Enterprises in the Dallas, Texas area. She has worked with Agile teams for over four years in project management, Scrum Master, and coaching roles. Allison also volunteers locally as one of the organizers of the DFW Scrum user group.

3 Roman Pichler, *Agile Product Management with Scrum: Creating Products that Customers Love*

4 Elizabeth Keogh with credit to Chris Matts, <http://www.infoq.com/news/2008/06/new-user-story-format>

5 Bill Wake, <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

6 Dan North, <http://dannorth.net/introducing-bdd/>

