Chapter 1
Product Ownership Is a Team Sport

Shane Hastie
Software Education, New Zealand

ABSTRACT
A number of agile brands downplay the need for business analysis and requirements management on agile projects, putting large store in the role of the Product Owner. This paper tackles some of the problems this misconception can result in and shows how effective product ownership almost always requires a team with a variety of skills and backgrounds to be effective. Product Ownership requires clarity of vision, alignment with organizational strategy, understanding of the development process and the ability to communicate with a wide variety of stakeholders across all levels both inside and outside the organization. The complexity of the role is most often more than a single person can (or should) cope with – effective product ownership requires a teamwork approach covering a variety of skills and knowledge.

INTRODUCTION
A number of agile brands downplay the need for business analysis and requirements management on agile projects, putting large store in the role of the Product Owner. This paper tackles some of the problems this misconception can result in and shows how effective product ownership almost always requires a team with a variety of skills and backgrounds to be effective.

Product Ownership requires clarity of vision, alignment with organizational strategy, understanding of the development process (Methodology, 2014) and the ability to communicate with a wide variety of stakeholders across all levels both inside and outside the organization. The complexity of the role is most often more than a single person can (or should) cope with – effective product ownership requires a teamwork approach covering a variety of skills and knowledge (Andrea, 2005).

In this chapter we examine various aspects of Product Ownership, the characteristics needed in the team for successful product ownership, look at the capabilities needed in a product ownership team, examine the difference between value and velocity and provide some practical tools which can be used for product ownership in agile and non-agile projects (Currim, Mintz, & Siddarth, 2015).

DOI: 10.4018/978-1-4666-9858-1.ch001
PRODUCT OWNERSHIP IN CONTEXT

The Product Owner Role

The Product Owner is one of the three roles defined in Scrum, and is referenced in many of the agile brands. The term Product Owner has become almost ubiquitous yet there is a lot of confusion and lots of misinformation about what the role is and what it actually entails.

The Scrum Guide defines the Product Owner role as follows:

*The Product Owner is responsible for maximizing the value of the product and the work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.* (Education, 2014)

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed. (Schwaber, 2013)

Extreme Programming talks about needing the Customer onsite all the time:

*One of the few requirements of extreme programming (XP) is to have the customer available. Not only to help the development team, but to be a part of it as well. All phases of an XP project require communication with the customer, preferably face to face, on site. It’s best to simply assign one or more customers to the development team.* (Programming, 2014)

This puts a huge demand on the individual taking the Product Owner or Onsite Customer role, one which for the vast majority of real-world projects is beyond the skills and capabilities of any one individual.

Project Types

The nature of knowledge work is that it is inherently unpredictable, especially in complex environments such as software development projects. The typical software development project today requires interfacing with a number of existing systems both internal to and outside of the control of the organization building the product, solving problems for a variety of stakeholders from different business units or divisions spread across multiple locations in different time zones (Hastie, 2014).

Add to this complexity the nature of product development projects – often the problem to be solved is unclear and unknowable until the solution.
21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product’s webpage: www.igi-global.com/chapter/product-ownership-is-a-team-sport/145031?camid=4v1


Related Content

An Empirical Study on Novice Programmer’s Behaviors with Analysis of Keystrokes
Dapeng Liu, Shaochun Xu and Huafu Liu (2013). International Journal of Software Innovation (pp. 68-87). www.igi-global.com/article/empirical-study-novice-programmer-behaviors/77619?camid=4v1a

Mobile Interaction in Real Augmented Environments: Principles, Platforms, Models, Formalisms, Development Processes, and Applications

Teaching Software Architecture in Industrial and Academic Contexts: Similarities and Differences

Model Checking of Multitasking Real-Time Applications Based on the Timed Automata Model Using One Clock