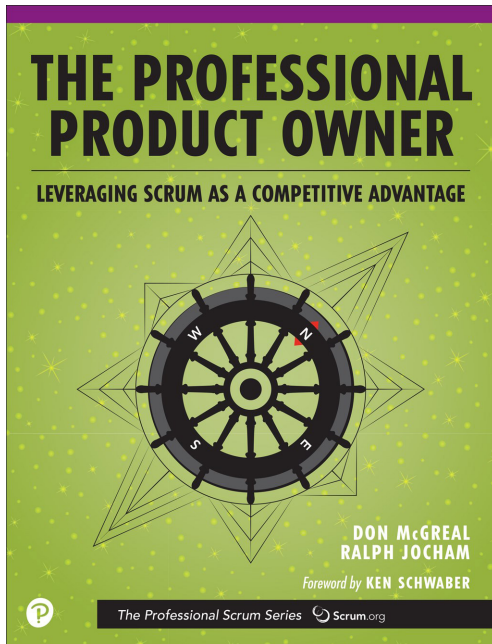


# The Professional Product Owner's Guide to Maximizing Value with Scrum



*"This book presents a method of communicating our desires, cogently, coherently, and with a minimum of fuss and bother."*

—Ken Schwaber, Chairman & Founder, Scrum.org

The role of the Product Owner is more crucial than ever. But it's about much more than mechanics: it's about taking accountability and refocusing on value as the primary objective of all you do. In **The Professional Product Owner**, two leading experts in successful Scrum product ownership, Don McGreal and Ralph Jocham, show exactly how to do this. You'll learn how to identify where value can be found, measure it, and maximize it throughout your entire product lifecycle.

McGreal and Jocham guide you through all facets of envisioning, emerging, and maturing a product using the Scrum framework. They discuss strategy, lay out best-practices for managing complexity and continuously delivering value, and they define the concrete practices and tools you can use to manage Product Backlogs and release plans, all with the goal of making you a more successful Product Owner. Throughout, the authors share revealing personal experiences that illuminate obstacles to success and show how they can be overcome.

- Define success from the "outside in," using external customer-driven measurements to guide development and maximize value
- Bring empowerment and entrepreneurship to the Product Owner's role, and align everyone behind a shared business model
- Use Evidence-Based Management (EBMgt) to invest in the right places, make smarter decisions, and reduce risk
- Effectively apply Scrum's Product Owner role, artifacts, and events
- Populate and manage Product Backlogs, and use just-in-time specifications
- Plan and manage releases, improve transparency, and reduce technical debt
- Scale your product, not your Scrum
- Use Scrum to inject autonomy, mastery, and purpose into your product team's work

## ORDER & SAVE

### Save 35% When You Order

from [informit.com/TPPO](http://informit.com/TPPO) and enter the code **SCRUMORG** during checkout

**FREE US SHIPPING** on print books

### Major eBook Formats

Only InformIT offers PDF, EPUB, & MOBI together for one price

## OTHER AVAILABILITY

Through O'Reilly **Safari** subscription service

**Booksellers** and online retailers including Amazon/Kindle store and Barnes & Noble | [bn.com](http://bn.com)

# The Professional Product Owner

**LEVERAGING SCRUM AS A  
COMPETITIVE ADVANTAGE**

Don McGreal

Ralph Jocham

◆◆Addison-Wesley

Boston • Columbus • Indianapolis • New York • San Francisco • Amsterdam • Cape Town  
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City  
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [intlcs@pearson.com](mailto:intlcs@pearson.com).

Visit us on the Web: [informit.com/aw](http://informit.com/aw)

Library of Congress Control Number: 2018934196

Copyright © 2018 Scrum.org

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screenshots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. Screenshots and icons reprinted with permission from the Microsoft Corporation. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit [www.pearsoned.com/permissions/](http://www.pearsoned.com/permissions/).

ISBN-13: 978-0-13-468647-9

ISBN-10: 0-13-468647-0

1 18

To my incredibly patient, supportive, and awesome wife Marita; also, to my little M&M's, Meagan and Molly, for being constant sources of laughter, love, and inspiration.

—Don

To my family—thanks for your understanding and lasting support. All of this, the travels, the long hours, would not have been possible without you. Natacha, you are the best and I love you. Noémie and Anaïs, thanks for keeping me down to earth. You are the best daughters I could wish for.

—Ralph

# CONTENTS

Foreword		xiii
Introduction		xv
Acknowledgments		xxi
About the Authors		xxiii
PART I	Strategy	I
Chapter 1	Agile Product Management	3
	Quiz	3
	Product Mindset versus Project Mindset	4
	What Is Product Management?	7
	The Product Management Vacuum and the Three Vs	9
	Vision	11
	Value	13
	Validation	14
	Product Management and Scrum	15
	The Product Owner	18
	Defining a Product	22
	Quiz Review	31

## CONTENTS

---

<b>Chapter 2</b>	<b>Vision</b>	<b>33</b>
	Quiz	33
	Business Modeling	35
	Business Model Canvas	35
	Product Vision	40
	Focused	41
	Practical versus Emotional	46
	Pervasive	49
	Visioning with Scrum	50
	Technical Strategy	51
	Quiz Review	53
<b>Chapter 3</b>	<b>Value</b>	<b>55</b>
	Quiz	55
	Value Defined	56
	Delivering Value	57
	Value Metrics	60
	Evidence-Based Management	65
	Current Value	68
	Time to Market	73
	Ability to Innovate	78
	Tracking Metrics	83
	Where Your Money Goes	85
	Negative Value	86
	Visible	87
	Invisible	87
	Value Neutrality	89
	Perversion of Metrics	89
	Quiz Review	92
<b>Chapter 4</b>	<b>Validation</b>	<b>93</b>
	Quiz	93
	Stakeholder Feedback	95
	Marketplace Feedback	97
	Minimum Viable Product	97
	Minimum Viable Product through Kano	98

	MVP Patterns	102
	Promotional MVP	102
	Mining MVP	103
	Landing Page MVP	103
	Wizard of Oz MVP	103
	Single-Feature MVP	104
	Pivot or Persevere	104
	Quiz Review	108
<b>PART II</b>	<b>Scrum</b>	<b>109</b>
<b>Chapter 5</b>	<b>Empiricism</b>	<b>111</b>
	Quiz	111
	It's a Complex Problem	111
	Certainty Quiz	115
	Visualizing Complexity	116
	Cynefin	118
	Obvious	119
	Complicated	120
	Complex	121
	Chaos	123
	Putting It All Together	123
	Types of Complexity	126
	Managing Risk	126
	Quiz Review	133
<b>Chapter 6</b>	<b>Scrum</b>	<b>135</b>
	Quiz	135
	Why a Framework?	136
	The Pillars of Scrum	139
	Transparency	140
	Inspection	140
	Adaptation	141
	Scrum Roles	141
	Product Owner	142
	Development Team	145

## CONTENTS

---

Scrum Master	149
Others	151
Scrum Artifacts	154
Product Backlog	155
Sprint Backlog	157
Increment	158
Others	159
Scrum Events	164
Sprint	164
Sprint Planning	167
Daily Scrum	173
Sprint Review	176
Sprint Retrospective	179
Other	180
Iterative and Incremental	184
Agile Manifesto for Software Development	186
Quiz Review	188

## **PART III      Tactics      189**

<b>Chapter 7</b>	<b>Product Backlog Management</b>	<b>191</b>
	Quiz	191
	What Is a Requirement?	192
	Product Backlog	193
	User Stories	195
	Nonfunctional Requirements	199
	Epics	202
	Acceptance Criteria	205
	Spikes	208
	Product Backlog Ordering	209
	Measuring Value, Risk, and Size	213
	“Done”	215
	Definition of “Done”	215
	Example Definition of “Done”	220



“Ready” Is a Mindset	224
Getting to Ready	227
Lean Requirements Management	230
Story Mapping	232
Steps to Creating a Story Map	233
Explore the Story Map	234
Story Maps and Product Backlogs	235
The Past and the Future	236
Impact Mapping	236
Success Criteria	239
Specification by Example	241
Quiz Review	248
<b>Chapter 8 Release Management</b>	<b>249</b>
Quiz	249
Reasons to Release	250
Release Strategy	251
Major Releases	252
Minor Releases	256
Functional Releases	256
Estimation and Velocity	259
Managing Multiple Teams	264
Scaling Products	267
One Product, One Development Team	268
Several Products, One Development Team	268
Several Products, Several Development Teams	270
One Product, Several Development Teams	271
The Nexus Framework	272
Reporting	273
Forecasting Basics	274
Forecasting Across Multiple Products	278
Percentage of Completion	280
Monte Carlo Simulation	282
Which Color Is Your Velocity?	287

## CONTENTS

---

	Budgeting	289
	Governance and Compliance	294
	Kickoff	300
	Quality	304
	Definitions	304
	Types of Quality	306
	Keeping Quality	307
	Quiz Review	313
<b>Chapter 9</b>	<b>The Professional Product Owner</b>	<b>315</b>
	Understanding Product Owner Success	316
	The Receiving Product Owner	316
	The Initiating Product Owner	317
	You	317
	Skills and Traits	318
	Measuring Success	321
<b>Index</b>		<b>323</b>

---

# FOREWORD

---

## COMMUNICATION, INTENTIONALITY, AND FULFILLMENT

You may have something you want done. You may have a vision you want fulfilled (a new way for people to travel between two places); a product you want created (a water slide at a lake, or a quantum computer); an improvement that you want to make to something (customer service is quicker, more effective, friendlier). Regardless, you are a person who has the authority and resources to try to get what you want. But you find it very, very hard.

Your skill is visioning what you want. Your skill is not necessarily building what you want (even if you could, you don't have time to do so).

You just have to tell someone what you want, fund and provision them adequately, and the result will be great (or at least satisfying).

We assume when we communicate with other people that they understand what we mean. Not necessarily so. You may be inarticulate or incomprehensible. The people you communicate with may be dunderheads.

## FOREWORD

---

Even more distressing, we assume that we know what we mean, even though we may not have thought it through accurately or completely. Our attempted communication may be premature. But, who has time to wait?!

This book presents a method of communicating our desires, cogently, coherently, and with a minimum of fuss and bother. Otherwise, we can alienate our allies, waste our capital, and cause irreparable damage. I know.

A great strength of Scrum is your ability to frequently check the clarity of what is being communicated, as well as how well the others perceive what you are communicating.

Frequent inspection of the clarity of communications and the consequences is important. It is particularly important at the start of an endeavor when we are just learning how to communicate what the results may be. As we get better at communicating, the communications become more precise. With less effort, we learn how to determine the unknown and turn it into clarity, and we get the results we desire.

As the person who wants something, when you use Scrum to improve communications and the outcomes, your role is called the Product Owner. Don McGreal and Ralph Jocham have written this book to help you use Scrum to do so. They should know because they have done it.

—Ken Schwaber

---

# INTRODUCTION

---

This book describes how to effectively manage man-made products, mostly software products. But it just as easily could address other man-made products such as electrical grids, nuclear power plants, apple orchards, nano-robots, even storm-drainage systems. Anything envisioned, created, sustained, and eventually retired or replaced by people is within our purview.

We specifically address complex products, where more is unknown about their context than is known. The product's creator—its Product Owner—perceives a space of ideas and conceives something that others might find valuable or useful.

Take as an example the first version of iOS developed for the iPhone. As this product was being conceived and created, more was unknown than known, and a certain degree of success and failure was involved. The Product Owner brought the vision to a small group of people who had expertise in the necessary technology, market, and products. This group employed empiricism and small self-organizing teams to manage the creation and development of iOS, to control the risk, and to create value.

## INTRODUCTION

---

Sometimes an idea isn't ready for prime time. The technology might not be adequate for the vision, or the people may not be adequately skilled. However, the risk is controlled through short cycles of experimentation.

This process is called Scrum,<sup>1</sup> a framework for creating and then further developing complex products. Scrum identifies the Product Owner as the person who brings the product to life, from vision to creation, and who remains responsible for the product's viability as it develops and continues through its life cycle. A Product Owner is the one person who is accountable for a product at any point in time.

A product does things, performs functions, or causes change or results. A product's life cycle includes the following components:

- **Creation**—A product is envisioned and parts of it come to life, so it has some of the capabilities and can perform some of the functions that have been envisioned for it.
- **Emergence**—As the product is used and time passes, new capabilities and functions appear. These may be created for it or may be appended to it by interfaces to other products.
- **Maturity**—The product reaches maturity when fully capable, as envisioned and as emerged, as shaped by marketplace forces, new technologies, and the capabilities of its Product Owners.
- **Senescence**—Over the hill, the product is still used but has been eclipsed by newer, easier, more appealing products that have more or fewer capabilities, at greater or lesser prices, but are preferred by the marketplace and are more valuable.

A product could be a computer, software the computer is operating, a security system, a camera, a car, a workflow system, just-in-time-inventory software, a rocket, or a business function that uses one or more of the above and performs a function for an organization.

---

1. "Scrum Guides," Scrum.org, accessed March 4, 2018, <http://scrumguides.org>.

Examples of notable products and Products Owners include the following:

- Self-landing rockets—*Elon Musk*
- Electric cars—*Elon Musk*
- iPhones—*Steve Jobs*
- Polaroid camera—*Edwin Land*
- Model T car—*Henry Ford*
- Scrum—*Ken Schwaber and Jeff Sutherland*

These Product Owners were visionaries, people who imagined different methods of doing things, envisioned products to accomplish these things, and then caused these products to emerge. For these products to be remembered, their Product Owners had to guide them to maturity in the marketplace, where they proved themselves useful to people or organizations.

Scrum helps Product Owners during the visionary phase by simplifying demands on them. Product Owners who can excite, can envision, can cause the product to emerge are sometimes less skilled at managing and administering the product as it matures. That requires a person trained in more traditional skills such as manufacturing, inventory, marketing, sales, support, service, and invoicing. A Product Owner who has both sets of skills is the professional Product Owner.

Most of us are familiar with Product Owners who oversee products in the mature phase of their life cycles. To the extent that they are working closely with their stakeholders and are imbued with the vision, they are successful. Product Owners who also run the business, respond to market forces, and help the product morph as new technologies and ideas emerge help it to become more useful.

Senescence is a difficult part of the product life cycle. We have all seen products from IBM, CDC, Xerox, Kodak, Motorola, Nokia, Blackberry, Wang, DEC, and other organizations that reach this point in their life cycles. To the extent that these products are gracefully ridden into their graves, they sustain the organizations that hosted them through maturity. Now they are on life

## INTRODUCTION

---

support. If they have been successfully carried through maturity, they may have provided an opportunity for new visionaries to come up with new products that can sustain the organization. Usually not.

This book describes how a person acting in the role of Product Owner can use Scrum to envision, emerge, and mature a product. Throughout the life cycle, the product is passed from person to person. In our view, it has to be passed from one accountable individual to the next accountable individual. That single person, the Product Owner, is responsible for everything that happens regarding the product, its value to its host organization, and to those who use it.

The Product Owner causes the product to live and grow in many different ways, such as development, partnerships, and interfaces. However, this one individual is the “buck stops here” person; he or she alone, not a committee or group, fulfills this function.

We have a great example of this in the United States with the Affordable Care Act (ACA) and the healthcare.gov site. When it was time for ACA to come to life, it didn’t. It didn’t respond on the Internet, it would not allow people to register, it confabulated data. And who was responsible for this disaster, this embarrassment? Nobody was sure.

*A successful child has many parents. A failure has none.*

—based on Tacitus, Agricola, ca. 98 AD

Eventually, the Secretary of Health, Education, and Welfare said that she was responsible, but she meant only that she was in the wrong place at the wrong time – not the Product Owner.

If you have a product at some point in its life cycle, and you want to use Scrum to create more value in it for its users, you will be its Product Owner. This book aims to tell you how to do so.

This book is part of a series of books by Scrum.org known as the professional series. This series was founded on a uniform set of values that bind the



people involved in work, so that they can trust each other, minimize waste, and succeed.<sup>2</sup>

## HOW TO READ THIS BOOK

This book is laid out in a way that assumes you have some Scrum knowledge. If you are new to Scrum, we suggest you look first at Part II.

We reference the official *Scrum Guide* by Ken Schwaber and Jeff Sutherland (creators of Scrum) throughout the book and have made every effort to stay consistent with its language. For example, we capitalize official Scrum terms for roles, artifacts, and events the same way the *Scrum Guide* does.

Each chapter is relatively independent. Chapters are grouped into three parts:

- **Part I: Strategy**—This part has very little to do with Scrum itself. Instead our focus is proper agile product management and maximizing the ROI of a product. We introduce the three Vs (vision, value, validation) as a way to achieve this.
- **Part II: Scrum**—This part starts by defining empirical process control and how Scrum is a tool for managing complexity and continuous delivery of value. With help from the *Scrum Guide*, we define each role, artifact, and event with particular attention to the Product Owner role.
- **Part III: Tactics**—Part III introduces more concrete practices and tools for managing Product Backlogs and release plans, and concludes by examining what it means to be a professional Product Owner.

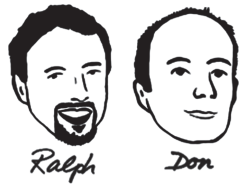
Each chapter begins with a short quiz. The intent is to set the stage for the chapter by inviting you to think independently about the core topics. We review the quiz at the end of the chapter to determine whether your thinking has changed at all—consider it Test-Driven Reading.

---

2. Other publications can be found or referenced at <http://scrum.org>.

## INTRODUCTION

---



*Along the way, we share personal anecdotes from relevant experiences. These blurbs have one or both of our faces alongside them. Just like this one.*

We put a lot of thought and practice into the Scrum.org Professional Scrum Product Owner course we both teach and maintain. We feel that this book is a perfect companion to that course as it contains all the same information and much more.

*We sincerely hope that you enjoy reading it as much as we do teaching, discussing, coaching, and writing about these topics.*

—Don and Ralph

Register your copy of *The Professional Product Owner* on the InformIT site for convenient access to updates and/or corrections as they become available. To start the registration process, go to [informit.com/register](http://informit.com/register) and log in or create an account. Enter the product ISBN (9780134686479) and click Submit. Look on the Registered Products tab for an Access Bonus Content link next to this product, and follow that link to access any available bonus materials. If you would like to be notified of exclusive offers on new editions and updates, please check the box to receive email from us.

---

xx

---

# ABOUT THE AUTHORS

---

**Don McGreal**, in his role as VP of Learning Solutions at Improving (improving.com), is a hands-on agile consultant and instructor. He specializes in agile coaching at the enterprise and product management levels within larger organizations.

Don is a Scrum.org Professional Scrum Trainer who has authored and taught classes for thousands of software professionals around the globe. He is also cofounder of TastyCupcakes.org, a comprehensive collection of games and exercises for accelerating the adoption of agile principles.

Don is an Irish French-speaking Canadian living in Texas.

**Ralph Jocham** is a German citizen who spent the last 20 years collecting professional software and product development experience in France, the United Kingdom, the United States, and now Switzerland. He became an agile evangelist in 2000 and perfected his approach at ThoughtWorks.

Ralph is also Europe's first trainer with Scrum.org and has taught thousands of professionals around the globe. When not busy running his company,

## ABOUT THE AUTHORS

---

effective agile ([effectiveagile.com](http://effectiveagile.com)), or helping all kinds of enterprises in Europe, he enjoys teaching at universities.

Ralph nevertheless finds time to spend quality time with his family on a regular basis, treating them to home-cooked international fine cuisine and going on long walks with the family dog.

Both Don and Ralph are course stewards for the Scrum.org Professional Scrum Product Owner course taught around the world.

# AGILE PRODUCT MANAGEMENT

*We didn't do anything wrong, but somehow, we lost.*

—Nokia CEO, Stephen Elop

## QUIZ

To set the stage for this chapter, try answering each of the following statements with Agree or Disagree. Answers appear at the end of the chapter.

Statement	Agree	Disagree
Schedule, budget, and scope are the best ways to measure project success.	<input type="checkbox"/>	<input type="checkbox"/>
Product management is an essential practice for Product Owners.	<input type="checkbox"/>	<input type="checkbox"/>
A Product Owner should act as a proxy between the business and technology.	<input type="checkbox"/>	<input type="checkbox"/>
A Product Owner establishes a vision for a product.	<input type="checkbox"/>	<input type="checkbox"/>
Scrum provides all the tools necessary for successful product management.	<input type="checkbox"/>	<input type="checkbox"/>
Every development effort has a product.	<input type="checkbox"/>	<input type="checkbox"/>

## PRODUCT MINDSET VERSUS PROJECT MINDSET

A *project* starts with an idea. If the idea sounds promising, a project can be initiated to make it a reality.

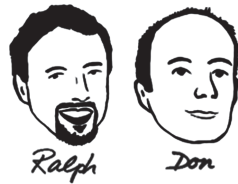
Important projects require organization and accountability. This is where a *project manager* comes in. A project manager is someone skilled in managing tasks and people, but not necessarily knowledgeable or passionate about the domain.

This project manager gathers enough information to create a plan. The plan will outline in detail the scope of the initiative over several project milestones and estimate how much time and money will be needed to complete it all. That manager then asks for resources (including people), forms a team around the project, and keeps each team member on task to ensure a successful outcome. Success is ultimately measured by how closely the project manager sticks to the initial plan—*scope, time, budget*.

This seems perfectly reasonable—at first.

But what about projects that remain on time, on budget, and within scope, and yet still do not succeed? For years, Nokia led the market in producing mobile phones. It delivered phone after phone on tight schedules. It knew how to run a project, and each individual project likely succeeded. But what about the company's phones? Is Nokia still around? Yes, as a department of Microsoft. What is interesting is that Microsoft paid more for Skype (\$8.6 billion), a few hundred person services company, than it did for Nokia (\$7.2 billion), a company that at one point had vast infrastructure, millions in hardware assets, and 100,000 employees.

If you or your company think in terms of projects and less in terms of products and value, the tide of fortune can turn rather quickly.



Interestingly, the word “project” used to mean to do something before (pro-) acting (-ject). In the 1950s, project management became more mainstream with the introduction of several techniques within the engineering and defense industries. This expanded the definition of “project” to include both planning and execution and has since expanded to hundreds of techniques across many other industries.

Don’t worry, this is not a project management book. However, an understanding of the current state of project management should provide some context for why we wrote it.

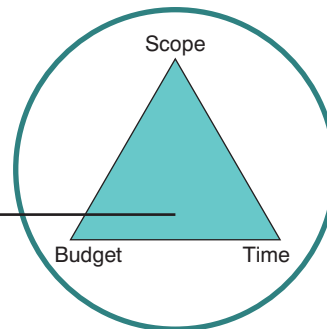
This *project mindset* (see Figure 1-1) defines success from the “inside out,” using internal measurements that are more about task management and about how accurately the initial plan was followed.

### Project Mindset

Success *upfront* defined inside out:

- Scope
- Time
- Budget

Leads to less business involvement, more task management.



**Figure 1-1** Visual representation of a project mindset

What is the alternative?

What do your customers value? Projects or products? The objective is not to deliver projects but to deliver value through *products*—products that ultimately lead to higher revenue and lower costs for your organization. Products

that are so great that your customer base will grow and existing customers will stick around.

So forget about projects. Approach your idea and its business case as a product. Give your product idea to a capable team and introduce them to meaningful business targets such as user adoption, sales, and stakeholder satisfaction. Impress on the team that the only way to influence targets like these is to release something of value as soon as possible and validate your business assumption.

Now that the team has the proper mindset, the next question should be: “What can we ship first that will have the biggest impact on our targets?”

Suddenly, delivery and business start to align and bridge their communication gap.

This *product mindset* (see Figure 1-2) is an “outside in” approach that uses external measurements to actively guide the development of the product to maximize value.

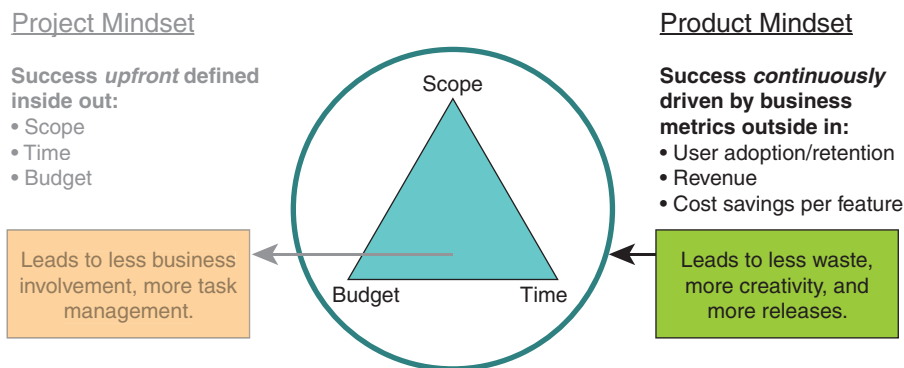


Figure 1-2 Product versus project mindset



A product mindset:

- encourages more frequent releases, which results in earlier feedback from the marketplace;
- communicates objectives instead of tasks; teams get more creative with their solutions and take more ownership over their plans; and
- eliminates waste by depending less on task assignment, reporting, and management decisions.

Contrast this with the project mindset, which leads to:

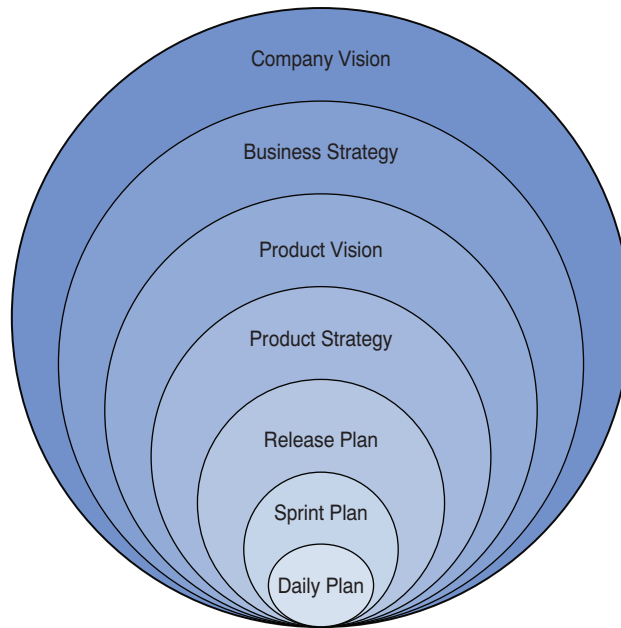
- less business involvement;
- more handoffs;
- more task management; and
- more people management.

But what if there isn't a product?

Read on. Later in this chapter, we make the case that software development always involves a product.

## WHAT IS PRODUCT MANAGEMENT?

Figure 1-3 shows the different layers of planning within an organization that develops products of any kind. At the core of any Scrum-based approach is a *daily plan* within a *Sprint plan*. (Sprints and Sprint Planning are described in Chapters 5 and 6.) Both these plans belong to the Development Team that is doing the work. They have autonomy in developing a plan on how to best meet the Sprint Goal and inspect and adapt that plan every day.



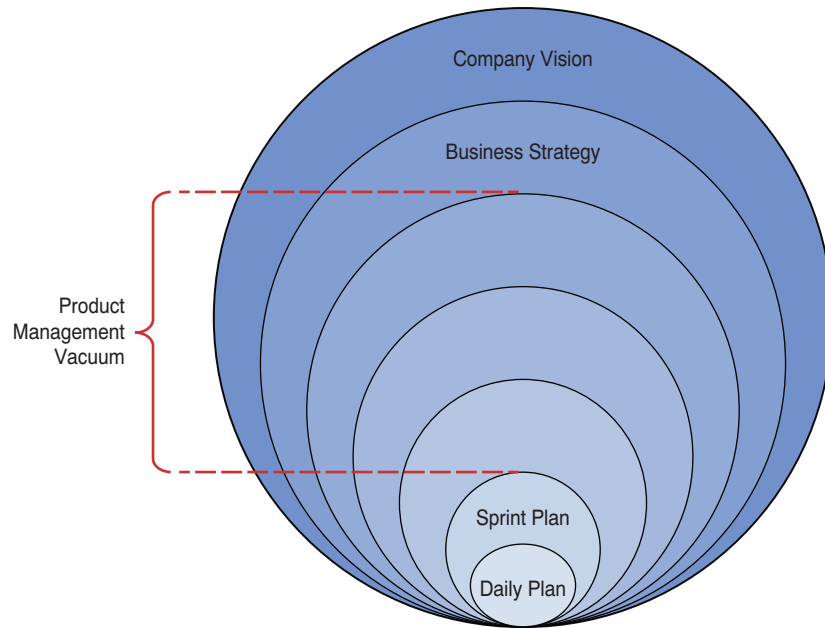
---

**Figure I-3** The different layers of planning

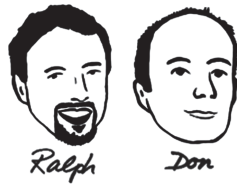
The two outside planning layers are the overall *company vision* and *business strategy*. Both are typically owned by an executive team or CEO, who establishes and communicates a company-wide vision and promotes an overall strategy under that vision.

In between the larger organizational goals and the work that Development Teams do day-to-day is a gap (see Figure 1-4).

## THE PRODUCT MANAGEMENT VACUUM AND THE THREE Vs



**Figure I-4** The Product Management Vacuum



*We call this the Product Management Vacuum, and it is a major motivation for writing this book.*

## THE PRODUCT MANAGEMENT VACUUM AND THE THREE Vs

The thing about any vacuum is that it has an innate need to be filled.

If you are not careful, the Product Management Vacuum will get filled with meaningless busy work and extensive task management, often guided by

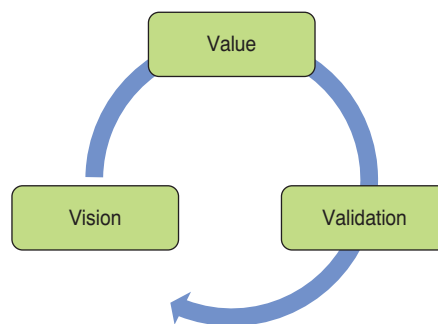
project metrics as described earlier. All the layers of budgeting, project charters, handoffs, and tasks breakdown mask the true intention of the initiative. You run the risk of being busy without a clear direction.

The bigger the vacuum:

- the more disconnected the technology groups are from the business;
- the less engaged the people on the ground become;
- the more reliance there is on project and task management;
- the more hierarchies and handoffs emerge;
- the more complexity is introduced;
- the harder it is to shift directions when the business climate changes;
- the more “busy work” is created;
- the more waste and rework occurs; and
- the less value is delivered to customers.

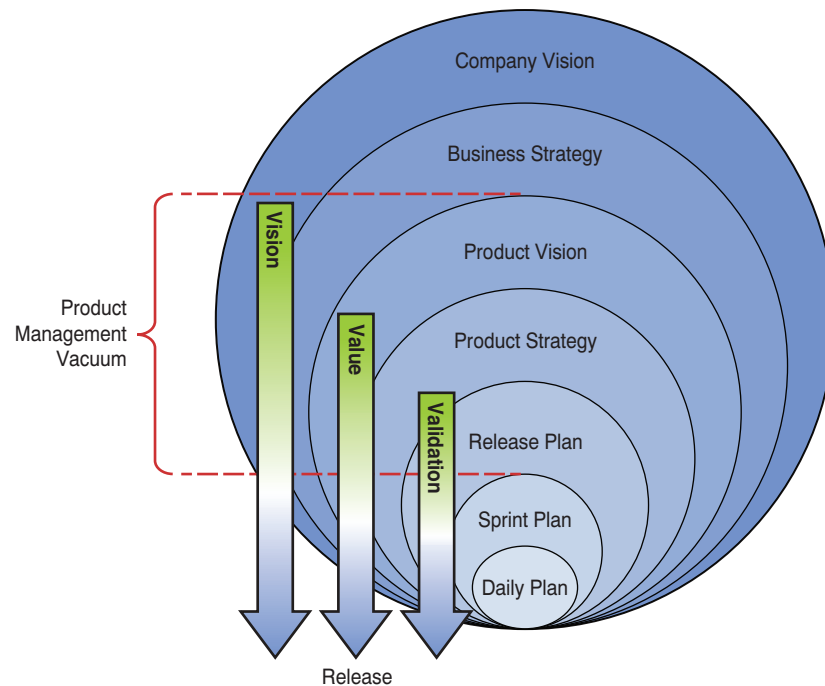
True product management is about embodying agility throughout the whole organization from the top down to the bottom and thereby filling the Product Management Vacuum. Done right, this creates a true competitive advantage.

To fill the vacuum the right way, use the three Vs shown in Figure 1-5. Figure 1-6 represents how the three Vs fill the vacuum.



**Figure 1-5** The three Vs

## THE PRODUCT MANAGEMENT VACUUM AND THE THREE Vs



**Figure I-6** How Vision, Value, and Validation fill the Product Management Vacuum

Let's take a brief look at each of the Vs.

### VISION

**Vision** creates *Transparency*.

The successful Product Owner establishes a clear product vision for her team, much like a military commander establishes clear intent for his subordinates. Doing so allows subordinates to act without direct orders if necessary to carry out the commander's intent.

From the book *Auftragstaktik: The Basis for Modern Military Command?* by Major Michael J. Gunther:

The use of mission tactics [*Auftragstaktik*] allow[s] subordinate commanders . . . to interpret how best to achieve the commander's intent based upon their understanding of the tactical situation. . . . *The success of the doctrine rests upon the recipient of orders understanding the intent of the issuer and acting to achieve the goal even if their actions violate other guidance or orders they have received.*<sup>1</sup>

Self-organization does not just happen. Much like in the military, the two main ingredients are shared vision and clear boundaries.

In the context of product development, the boundaries are provided by Scrum and the vision is provided through the Product Owner's strong leadership and communication.

The product vision anchors everything in the process. This vision creates transparency because it forms the basis for all following conversations, leading to a common understanding for why you are building the product and what your customers' needs are. According to Richard Hackman, 30 percent of a team's success depends on how it was launched.<sup>2</sup> A great and well-communicated vision is paramount for a successful launch.

You need to communicate the product vision again and again to keep everyone on board and honest. Never forget that this vision represents the voice of the customer. If you stop listening to your customers, the resulting product will be of less value or even alienate them.

- 
1. Michael J. Gunther, *Auftragstaktik: The Basis for Modern Military Command?* (Fort Leavenworth, KS: School of Advanced Military Studies, U.S. Army Command and General Staff College, 2012), 3 (emphasis added).
  2. J. Richard Hackman, *Leading Teams: Setting the Stage for Great Performances* (Boston, MA: Harvard University Press, 2002).



*I too often see the term “resources” used for human beings. These resources are given a catalog of requirements to implement and rarely have any idea about why. They lack the context of the vision and therefore are deprived of situational awareness. They fulfill exactly what was specified but often still miss the goal.*

Making the connection to the product’s (or even the organization’s) vision helps team members make goal-driven commitments and feel like they are a part of something bigger than themselves. So how do you create a clear vision? Techniques are explored in Chapter 2, “Vision.”

## VALUE

Defining **value** provides you with something to *Inspect*.

Imagine the vision as a long thread. Value is the individual pearls you attach as you progress. Vision provides a foundation and direction, but without the pearls attached to it, there is no value. A Scrum Team’s job is to identify and then attach pearls (value) to the thread (vision).

The first pearl could be either a large business initiative or a smaller distinct feature. Aim for the most valuable item first and attach it completely before moving on to the next one. In other words, always be in a position to deliver value.

“If you could have only one thing, what would it be?” is often a good opening question when identifying the most valuable.

*If everything is important, then nothing is.*

—Patrick Lencioni<sup>3</sup>

3. *Silos, Politics, and Turf Wars: A Leadership Fable about Destroying the Barriers That Turn Colleagues into Competitors* (San Francisco: Jossey-Bass, 2002).

Once you get this answer, often after persistent digging, try to truly understand the other person's view; try to grasp the underlying *why*. You might even go so far as to question the product's utility. Eventually, when you have narrowed it down and are convinced, go ahead and define how the value will manifest. Would a process take fewer clicks? How many? Will the behavior of a user change? How? Will a transaction be faster? How much faster? You need to provide more leadership than simply saying "let's do this!" If you are not able to quantify success or to prove the realization of value, then the chances of being on the wrong track are rather high. Do not forget that the only real proof, though, is through the customer. Everything before is nothing but a hypothesis.



*I like the idea that through the functionality we develop, we actually improve the world. It might not be world peace we reach, but we make a positive impact on someone's life—even if it is just one click less in a process.*

So how do you measure value? Techniques are explored in Chapter 3, "Value."

## VALIDATION

Validation causes *Adaptation*.

Most business assumptions are plainly wrong.<sup>4</sup> They look good on paper but do not hold up in the real world. Each valuable idea has to be validated as soon as possible. One place this is done in Scrum is the Sprint Review. The Sprint Review is the chance for all stakeholders to review the current state of the product and to get an insight into next steps. The closer the reviewers are to actual customers and the closer the Increment is to being released, the more realistic the feedback and subsequent adaptations.

---

4. Ronny Kohavi et al., "Online Experimentation at Microsoft" (ThinkWeek paper, Microsoft Corp., Seattle, WA, 2009).



Even if the Sprint Reviews go well and everyone seems happy, you still do not have true validation until the product or feature is in production and used. In Scrum, “Done” means the Increment is potentially releasable. However, to have ultimate validation and reduce the overall product risk, you need to establish a feedback loop with the marketplace. For this you need to release as frequently as the business can support.

The two core feedback loops in Scrum are on the process side and the product side:

- *Process* validation is about inspecting and adapting *how* the Scrum Team is working.
- *Product* validation is about inspecting and adapting *what* the Scrum Team is building.

Validation in the context of product management and the three Vs is about the latter: product validation.

Chapter 4, “Validation,” describes this in more detail.

## PRODUCT MANAGEMENT AND SCRUM

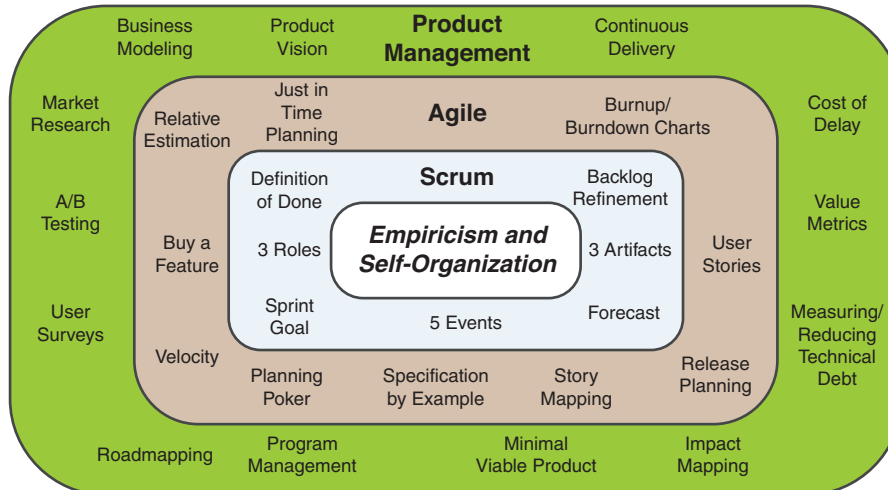
Building products requires that you consider a series of strategic activities. Consider the following list:

- Analyzing the industry and competition
- Maximizing ROI
- Forecasting and assessing feasibility
- Developing product strategy
- Planning releases
- Identifying customers and their needs
- Roadmapping the product
- Auditing results
- Creating outbound messaging

- Sustaining the product
- Executing the release
- Creating the business case
- Identifying product requirements
- Launching the product
- Developing customer retention strategy
- Defining product features and initiatives
- Retiring the product
- Marketing and branding

How many of these fit squarely in the realm of Scrum? Maybe three or four, but not many.

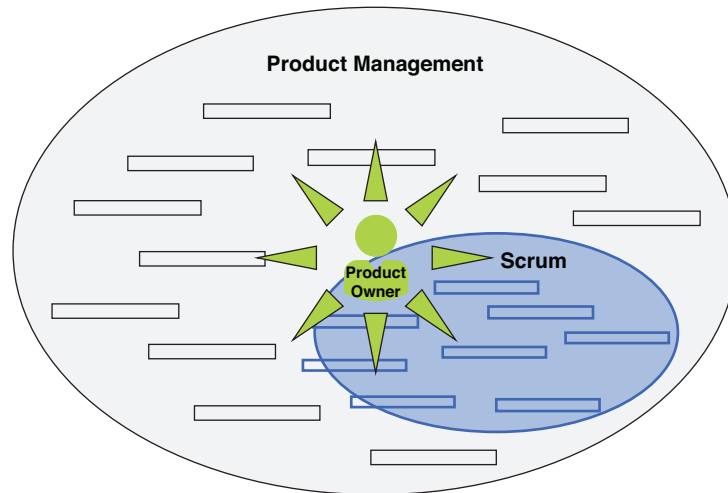
The world of product management is a lot bigger than Scrum, as suggested in Figure 1-7, which quite possibly explains why there is such a large Product Management Vacuum in the software industry.



... and many, many more.

**Figure 1-7** Scrum is augmented by many, many practices. (Figure © 1993-2016 Scrum.org. All rights reserved.)

This is where the Product Owner comes in. A Product Owner is one of the three Scrum roles. Although most product management activities are not part of the Scrum framework, as shown in Figure 1-8, a good Product Owner will take them on to fill the Product Management Vacuum.



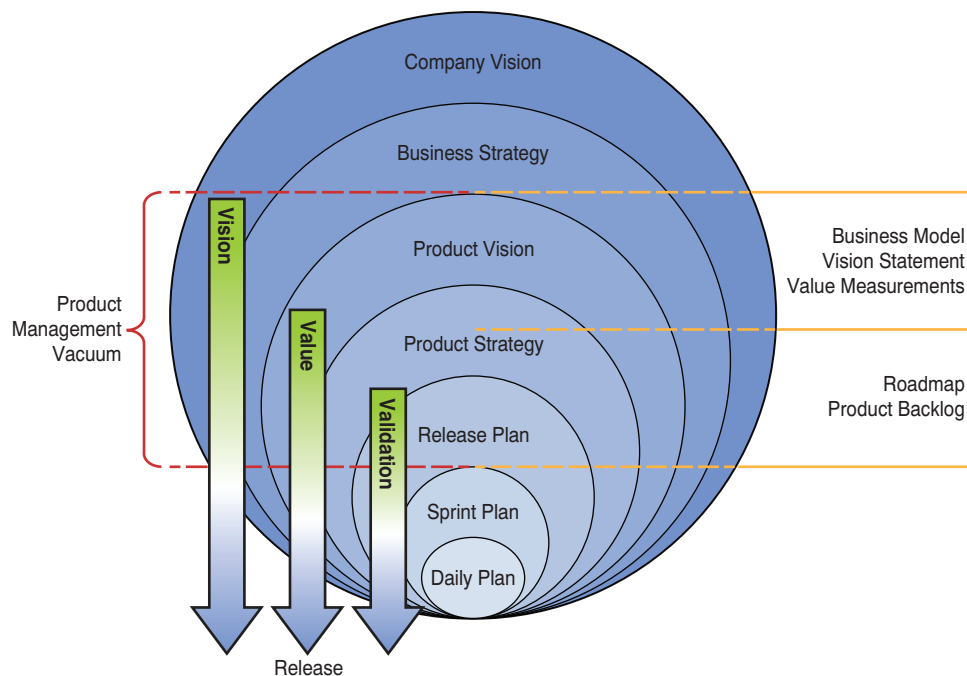
**Figure 1-8** Product Ownership is agile product management leveraging Scrum.

In other words, a good Product Owner is an agile Product Manager.



*When I work with an organization to identify ideal candidates for the Product Owner role, I first ask about the strategic activities above and who does them. If the organization already has a Product Manager who does many of these activities, then I consider them an ideal candidate for Product Owner. If nobody is doing these activities or that person is not willing or able to commit to a Product Owner role, then I insist on empowering our Product Owner to **own** these activities as a way of elevating the role beyond the tactical to the strategic.*

An empowered and entrepreneurial Product Owner fills the Product Management Vacuum, as shown in Figure 1-9.



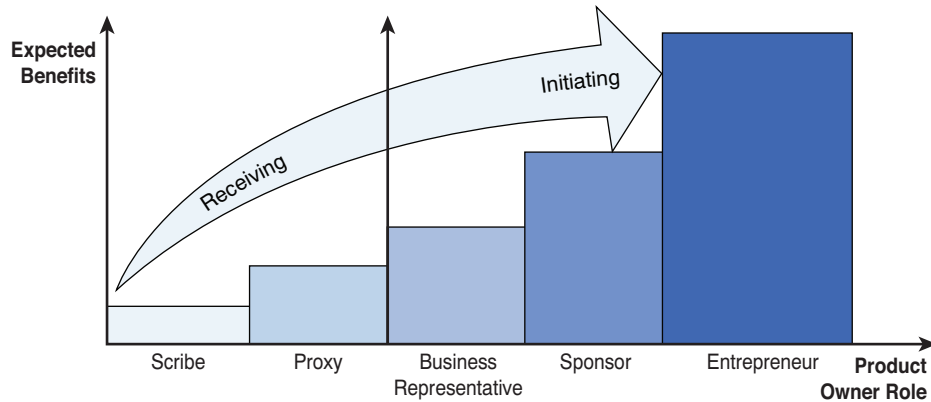
**Figure I-9** The Product Owner and the Product Management Vacuum

## THE PRODUCT OWNER

Who should play the role of Product Owner?

Later in this book, more time is spent on the specific tasks and responsibilities of a Product Owner. This chapter stays at a more strategic level.

In Scrum, you need a Product Owner. But it obviously takes more than just filling the role. The effectiveness of that role, and of the overall Scrum implementation, depends a lot on the type of person in that role (see Figure 1-10).



**Figure I-10** Product Owner role types by expected benefits

A scribe is likely someone on the technology side who has been tasked with capturing requirements for the Development Team. This person is often seen as the team's secretary and is asked to write everything down during meetings with the business. He has little to no decision-making ability, which severely impacts his effectiveness as a Product Owner and causes delays.

A proxy is still likely to come from the technology side but is seen as a representative of the business, maybe even for a particular product manager on the business side. She often has a business analyst or system analyst title. The problem is that a proxy creates an unnecessary indirection between the Development Team and the real influencers.

What is a likely response from the proxy whenever the team has a question for her? "Let me go ask," resulting in more delays.

Could the proxy get protective of her role and shut down any direct communication between the team and the business? Very likely. You surely see it all the time.

*Both the scribe and proxy are on the receiving side of what goes into the product. They get told what to do. At best, they work out the details.*

A **business representative** is someone from the business side rather than the technology side. This role is a clear improvement over the proxy role, as it demonstrates a commitment from the business toward the product. In contrast to the proxy Product Owner, this role provides more direct access to domain knowledge and stakeholder expectations. However, there may still be decision-making delays, as the business representative often has limited autonomy over product management.

A business **sponsor** is a big improvement on the Product Owner role. A sponsor is someone who spearheaded the initial business case and acquired the budget. Consequently, he has the trust and the mandate to make financial and product decisions on the spot. This creates fewer hiccups, less context switching, and largely improved flow. The Development Team can then focus more and get more things done sooner.

The ultimate Product Owner is an **entrepreneur**. This is someone who is spending her own money to fund the development of the product. This gives her complete responsibility over all product management decisions for both business and IT strategy. Now while this may not be a realistic situation for many organizations, an entrepreneurial mindset is still something that Product Owners should assume. They should expect to see a return on investment (ROI) as though their own money was at stake.

*The business representative, sponsor, and entrepreneur are more on the initiating side of the product. Because of their deeper business understanding and their two-way communication, they develop a stronger customer empathy. This empathy allows them to initiate rather than simply receive the right requirements.*

Keep in mind that the closer a Product Owner is to a sponsor the further he may disengage from the Development Team. Finding ways to maintain that sense of connection with the product vision becomes even more important and is what is expected of an entrepreneurial Product Owner.

Figure 1-11 provides a summary of this discussion of Product Owner roles types.

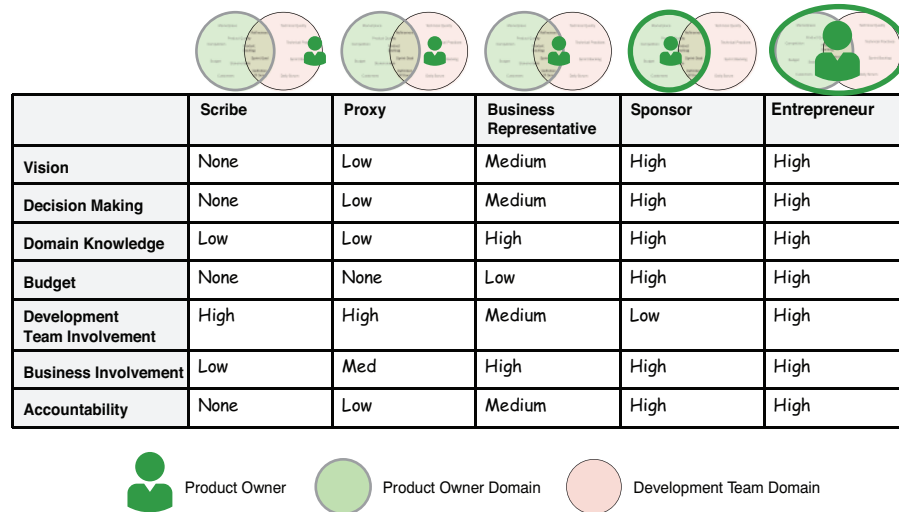


Figure 1-11 Summary of Product Owner role types



*I often find myself writing these five Product Owner role types down when talking to organizations. I like to ask them where they see their Product Owners. Not surprisingly, they are more often than not closer to the scribe and proxy roles. This then turns into a discussion of more practical actions a Product Owner could do to move up the list. Could a scribe start to be more representative of the business people? Could a proxy start learning more about the business side? Could a business representative ask to be more involved with budgeting and strategic direction? If the answer is no to any of these questions, then we should ask ourselves if we have the right person in the Product Owner role.*

## DEFINING A PRODUCT

What is a product? On the surface, this question seems simple. But it's not.

A product is anything that can be offered to a market that might satisfy a want or need.<sup>5</sup>

Companies sell many products—retail, financial products, seats on planes, cars.

In the world of software development, you make products too:

- Some are straight commodities—word processors, games, operating systems.
- Others are channels to other products—online retail, travel booking websites, mobile banking apps, search engines.

When this book refers to products, generally it means software products. However, many of the concepts discussed apply well beyond the realm of software.

To start, here are some assertions:

1. There is always a product. It may not always be obvious, but it is there and it needs to be identified.  
AND
2. Every product has a *customer* who is a:
  - a. *Consumer*: Anyone who gets value from your product, whether or not they pay for it
  - b. *Buyer*: Anyone who pays for your product, whether or not they use it
  - c. *Both*AND

---

5. Philip Kotler, Linden Brown, Stewart Adam, Suzan Burton, and Gary Armstrong, *Marketing*, 7th ed. (Frenchs Forest, New South Wales: Pearson Education Australia/Prentice Hall, 2007).



3. Every product has a *producer* who receives a core benefit through:
  - a. Revenue increase
  - b. Cost decrease or avoidance
  - c. Societal benefit

What happens far too often is that smaller areas of functionality or even smaller technical components are considered products, with no clear customer or value proposition.



*I had in one of my Product Owner trainings a participant who introduced himself as “My name is . . . and I am the Product Owner for testing.” Normally I am known for being swift with my words, but in this case I was speechless.*

Another common pattern is Conway’s Law:

Organizations which design systems . . . are constrained to produce designs which are copies of the communication structures of these organizations.<sup>6</sup>

The result of Conway’s Law is a series of interconnected “systems” built by different departments (or hierarchies) within an organization with little focus on the actual product or on who the customer even is.

This is the reason Scrum names the role *Product* Owner and not System, Feature, or Component Owner.

The best approach is to think about the product from a customer’s perspective. Which customer needs are you addressing? What do customers expect? What product improvements will make customers’ lives easier?

6. Melvin E. Conway, “How Do Committees Invent?” *Datamation* 14, no. 5 (1968): 28–31.



Ralph

*When we work with companies to define their products, we emphasize that this is ultimately a business decision. Let's align our technology groups based on the products our business wants the end-customers to see.*



Don

*For example, Company A may have a strategic direction of presenting their customers with a series of products to fulfill their needs, while Company B decides it would rather present its customers with one larger product. Both are valid options, and neither should be constrained by the technical software delivery structure. This needs to remain a strategic business choice, and technology should be able to adapt as seamlessly as possible.*

*This bridges the chasm between business and technology.*

When you think of a car, what is the product? The engine, the entertainment system, the steering, the air conditioning, the car itself? What exactly is the product?

As stated above, start with the customer. Who is the consumer and who is the buyer?

If you are buying the car for yourself, then you are both the buyer and the customer. If you are a parent buying a car for your child, you are the buyer but the child is the consumer.

Car companies must design products with both in mind. Cars must include airbags and safety ratings for the parent and fun colors and multimedia systems for the teen.

But you can look at it from a different angle: The car company is the buyer as well as the consumer for components within the car. The products are the engines, entertainment systems, air bags, and so on that are produced by vendors or even internal groups.

So, to know your product, you need to know the consumer and buyer—your product's customers.

Getting back to software, Table 1-1 presents some realistic examples. See whether these assertions still hold true.

**Table 1-1** Product Examples

Producer	Description	Buyer	Consumer	Producer Benefit
Microsoft	Professional office software for daily or occasional office work	Enterprise Individual user	Employee Individual user	Revenue (licensing)
Technology Department within an organization	Replace legacy backend system with updated web service API	Technology Department	Individual programmers	Cost savings (lower maintenance costs)
<i>Wikipedia</i>	Nonprofit encyclopedia with volunteer contributors	Donors	Information seekers	Social benefit (free information)
Call Center	A system for customer service agents to log calls	Call Center	Customer service agents in Call Center	Cost savings (shorter call times)
QA Department	Creating automated tests for development teams	QA Department	Internal Development Teams	Cost savings (better quality)
Business Analyst Group	Creating documents to hand over to IT for implementation	Business Department	IT Development Department	Revenue (salary)
Amazon.com	Retail website	Individuals	Individuals	Revenue (sales)
Facebook	Social media	Advertisers	Individuals	Revenue (advertising sales)

After you have identified your product, the next question is, “Is it a viable product?”

A viable product is one where the stated producer benefit comes to fruition. This means that you should have ways to measure the benefit along the way. Chapter 3, “Value,” explains more about how to do this. For now, consider this a major responsibility of a Product Owner for maximizing ROI.

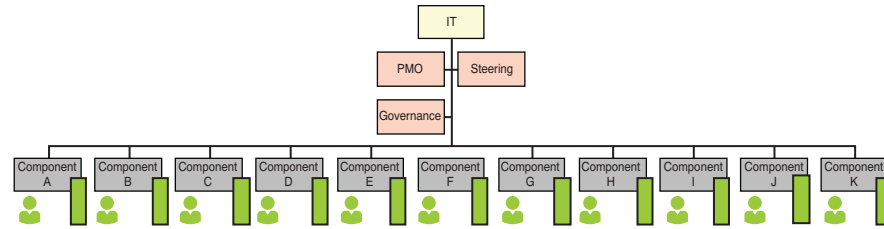
The preceding examples of the separate automated testing and business analysis groups rarely produce a viable ROI.

Admittedly, you will see examples of initiatives that have failed, time and time again. Is the automated testing group developing a product? Sure. But is the stated benefit of better quality even viable? Will the other Development Teams embrace testing environments that they did not develop themselves? Who will maintain this testing system after its implementation? The ROI is rarely there unless these efforts live with the Development Teams themselves. Regardless, it is important to understand the cost and perceived benefit and set up a feedback mechanism to ensure that the product is still viable.

An important takeaway when defining your product is to go as high as possible without losing sight of your core objectives. You have to find the right value area from the customer’s point of view. What does the customer need, and what is the customer willing to pay for? What provides value?

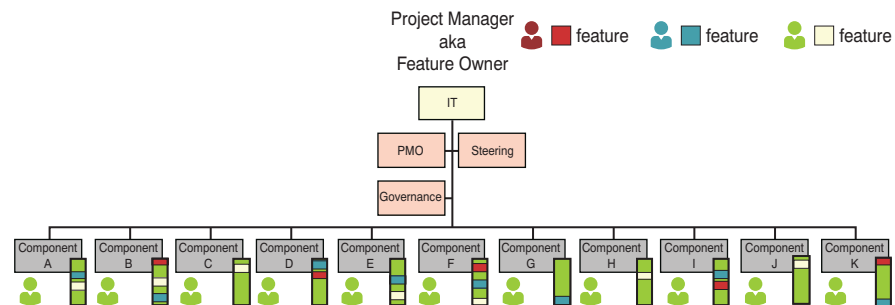
Coming back to the car example, would you buy a steering wheel or a seat? You most likely want a usable product, a complete car in this case. This product provides a benefit to you as a consumer.

If you structure your teams below the value areas, you will see too many Product Owners, with all the resulting coordination overhead. Imagine having a separate Product Owner for each car component with no real centralized vision for the whole car as one product (see Figure 1-12).



**Figure I-12** A Product Owner for each car component

Now when you add a new feature or large initiative for a value area, it needs to be broken down into units of work for each of the affected components. This creates enormous overhead, with many dependencies to be managed and resulting in skill shortage, timing, quality and integration issues. To handle all these dependencies, you will likely need to find a “feature” owner to take care of all the coordination and to be accountable for the feature. Some call this role a project manager (see Figure 1-13).



**Figure I-13** Project managers are feature owners.

The problems mentioned above can lead to the following:<sup>7</sup>

### 1. Sequential life cycle

The dependencies from the various components need to be aligned, which requires a plan taking handoffs into account. Once Component B is completed, it is handed over to Component F, and so on.

7. Cesário Ramos, “Scale Your Product NOT Your Scrum,” Scrum.org (February 2016).

**2. Unnecessary dependency management, coordination, and overhead management complexity**

In a plan-driven approach, all the coordination of the dependencies is left to managers, many of whom are nontechnical. By instead providing a clear goal and leaving the implementation details to feature teams, you get commitment and working functionality more quickly for review.

**3. Working on low-value features**

The feature is composed of functionality of various degrees of value, from very low to very high. Since the whole feature is being decomposed and top-down managed and coordinated in the components, there is no feedback to identify high- or low-value functionality.

**4. Handoffs, information scatter, and high inventory**

Individual feature activities need to be completed before the next activity can start. Communication and knowledge transfer at those handoffs is done by documents and specification. All those interim activities lead to high inventory, spread-out information, and too many handoffs.

**5. Loss of customer and whole-product focus**

When work is organized by activities to be completed by all those component teams, naturally the customer becomes secondary. There are enough dependencies, challenges, and other obstacles to manage that a customer, especially a customer providing feedback, can quickly become a nuisance.

**6. Opaque measure of progress**

The sequential approach means you lose the feedback, which then means you are stuck measuring progress toward your plan instead of toward your value goals.

**7. Inventing work**

What do you do when your component team runs out of work? You need to justify its existence. So you come up with “important stuff” that needs to be done so that you can justify your team’s payroll.

## 8. Bad quality

Since you focus on each component's quality, you lose sight of the final overall feature quality. Those local optimizations are known to cause problems for the real goal. Also, technical problems between those components are often not addressed correctly for lack of time. Remember the plan to be followed? This leads to technical quality issues in the long run.

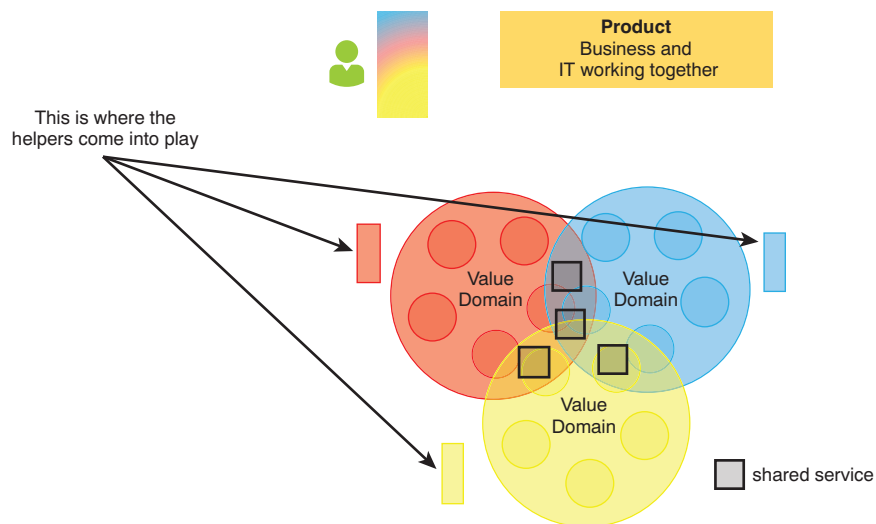
Instead of thinking in components, you need to think in terms of one (larger) product. This can go a long way. A Product Owner should be able to handle a fairly large number of Development Teams for a single product.

Remember, the ideal Product Owner is an entrepreneur—a single voice for the vision of the product, regardless of its size. A company has just one CEO, whether it has 100 or 100,000 employees.

This is referred to as the Santa Claus rule. No matter how busy Santa gets, he does not bring on other Santas. How does he cope? Elves.

As products scale, Product Owners should find themselves less involved with the day-to-day tactics of product development and more involved with strategy and direction. They can get help (elves) from teams, stakeholders, assistants, and so on for the more tactical work. As Santa, you can delegate responsibility, but you still remain accountable for the success of Christmas.

If this is not sufficient, then split the product into value domains on the right level of abstraction (see Figure 1-14). This also requires that you set up cross-functional feature teams underneath your product accordingly. The resulting intercommunication is essentially the steering committee, Project Management Office (PMO), and governance in one body. This addresses the problems mentioned previously.



**Figure I-14** Independent customer-focused value domains, sharing common services



Don

*I had a client who had several Product Owners for a large product. The client's solution was to allow each one to select something from his list for the next Sprint, round-robin style. While this may seem fair at first, is it the right strategic solution for the organization? Are the Development Teams truly working on the most valuable features? Who ultimately decides which features are best, regardless of how many voices there are? That person is the real Product Owner.*

The shared services between the value domains represent dependency management and integration on a technical level. With today's continuous integration, infrastructure as code, and test automation, this area is well understood. Coordination on a technical level is far more predictable. You either have a working product or you do not.

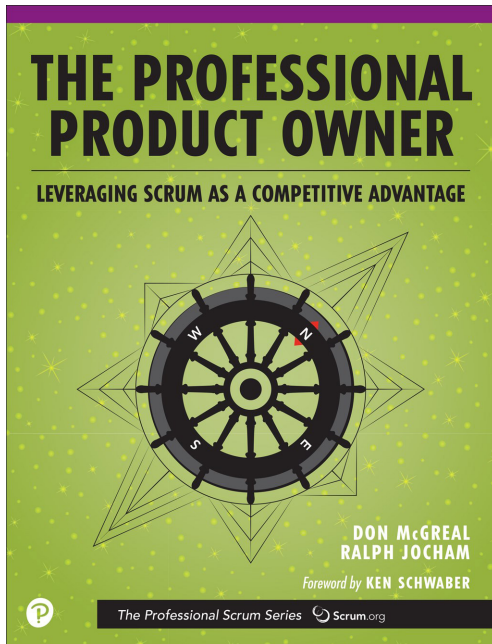


## QUIZ REVIEW

Compare your answers from the beginning of the chapter to the ones below.  
Now that you have read the chapter, would you change any of your answers?  
Do you agree with the answers below?

Statement	Agree	Disagree
Schedule, budget, and scope are the best ways to measure project success.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Product management is an essential practice for Product Owners.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
A Product Owner should act as a proxy between the business and technology.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
A Product Owner establishes a vision for a product.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Scrum provides all the tools necessary for successful product management.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Every development effort has a product.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

# The Professional Product Owner's Guide to Maximizing Value with Scrum



*"This book presents a method of communicating our desires, cogently, coherently, and with a minimum of fuss and bother."*

—Ken Schwaber, Chairman & Founder, Scrum.org

The role of the Product Owner is more crucial than ever. But it's about much more than mechanics: it's about taking accountability and refocusing on value as the primary objective of all you do. In **The Professional Product Owner**, two leading experts in successful Scrum product ownership, Don McGreal and Ralph Jocham, show exactly how to do this. You'll learn how to identify where value can be found, measure it, and maximize it throughout your entire product lifecycle.

McGreal and Jocham guide you through all facets of envisioning, emerging, and maturing a product using the Scrum framework. They discuss strategy, lay out best-practices for managing complexity and continuously delivering value, and they define the concrete practices and tools you can use to manage Product Backlogs and release plans, all with the goal of making you a more successful Product Owner. Throughout, the authors share revealing personal experiences that illuminate obstacles to success and show how they can be overcome.

- Define success from the "outside in," using external customer-driven measurements to guide development and maximize value
- Bring empowerment and entrepreneurship to the Product Owner's role, and align everyone behind a shared business model
- Use Evidence-Based Management (EBMgt) to invest in the right places, make smarter decisions, and reduce risk
- Effectively apply Scrum's Product Owner role, artifacts, and events
- Populate and manage Product Backlogs, and use just-in-time specifications
- Plan and manage releases, improve transparency, and reduce technical debt
- Scale your product, not your Scrum
- Use Scrum to inject autonomy, mastery, and purpose into your product team's work

## ORDER & SAVE

### Save 35% When You Order

from [informit.com/TPPO](http://informit.com/TPPO) and enter the code **SCRUMORG** during checkout

**FREE US SHIPPING** on print books

### Major eBook Formats

Only InformIT offers PDF, EPUB, & MOBI together for one price

## OTHER AVAILABILITY

Through O'Reilly **Safari** subscription service

**Booksellers** and online retailers including Amazon/Kindle store and Barnes & Noble | [bn.com](http://bn.com)