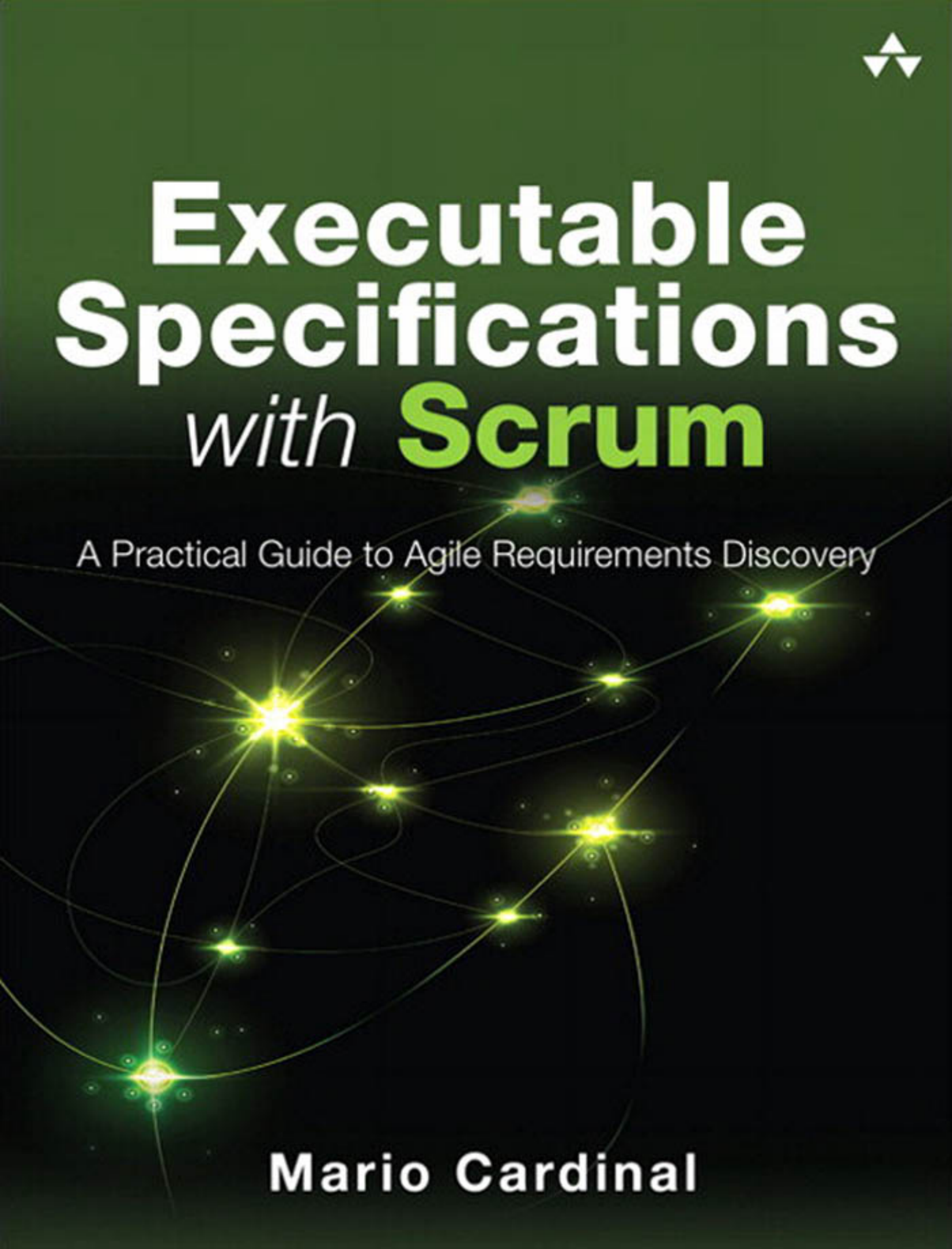
Executable
Specifications
with Scrum
A Practical Guide to Agile Requirements Discovery
Mario Cardinal

Praise for *Executable Specifications with Scrum*

"This is a great book that demonstrates the value of putting effort behind requirements in an Agile environment, including both the business and technical value. The book is well-written and flows nicely, approachable for both the manager and the developer. I am recommending this book to all Scrum teams who need to integrate business analysts and architects as active teammates."

—**Stephen Forte**, Chief Strategy Officer at Telerik and Board Member at the Scrum Alliance

"Cardinal's book brings to light one of the most important and neglected aspects of Scrum: Having user stories that are ready to sprint. Teams often complain about this, and the author offers practical advice on how to get it done right!"

—**Steffan Surdek**, co-author of *A Practical Guide to Distributed Scrum*

"*Executable Specifications with Scrum* doesn't shine through its depth but its breadth. This compendium of proven agile practices describes an overarching process spike touching important aspects of product development in a cohesive way. In this compact book, Mario Cardinal clearly explains how he achieves a validated value stream by applying agile practices around executable specifications."

—**Ralph Jocham**, Founder of agile consulting company effective agile. and Europe's first Professional Scrum Master Trainer for Scrum.org

"Cardinal provides deep insights into techniques and practices that drive effective agile teams. As a practitioner of the craft Cardinal describes, I now have a written guide to share with those who ask, 'What is this [ATDD/BDD/TDD/Executable Specification/etc] thing all about?' Regardless of the name de jour, Cardinal gives us what works."

—**David Starr**, Senior Program Manager, Microsoft Visual Studio

"Scrum is barely a process, only a framework. It is a tool, and you have to provide many complementary practices to reach true business agility. This book is perfect for teams that are using Scrum and want to learn about or get started with executable specifications."

—**Vincent Tencé and François Beauregard**, Scrum Trainers at Pyxis Technologies

"This book maps out the important place of specifications in an agile landscape to the benefit of agilists of all roles."

—**Erik LeBel**, Technology and Development Consultant at Pyxis Technologies

Executable Specifications with Scrum

This page intentionally left blank

Executable Specifications with Scrum

A Practical Guide to Agile Requirements Discovery

Mario Cardinal

Addison-Wesley
Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Library of Congress Control Number: 2013939927

ISBN-13: 978-0-32-178413-1

ISBN-10: 0-32-178413-8

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.

First printing, July 2013

Editor-in-Chief
Mark Taub

Executive Editor
Chris Guzikowski

Senior Development Editor
Chris Zahn

Marketing Manager
Stephane Nakib

Managing Editor
Kristy Hart

Senior Project Editor
Lori Lyons

Copy Editor
Apostrophe Editing Services

Senior Indexer
Cheryl Lenser

Proofreader
Paula Lowell

Editorial Assistant
Olivia Basegio

Cover Designer
Chuti Prasertsith

Senior Compositor
Gloria Schurick

To my four outstanding children:
Dominic, Lea-Marie, Romane, and Michael.

This page intentionally left blank

Contents

Contents

Figure List

Preface

There is a wide range of books that have been written about specifications. Unfortunately, most of them are not useful for software development teams. These books rely on traditional engineering practices. They assume requirements are known upfront and, once specified, will not change for the duration of the project. And if changes happen, they presume they will be minor, so they could be tracked with a change management process. They promote a sequential process starting with a distinct requirements phase that delivers a detailed requirements specification before starting to design and build the product.

Goal of This Book

It is my belief that traditional engineering practices are not suitable for software development. Central to the process of software specification is a high level of uncertainty, which is not the case with traditional engineering. Fortunately, with the growth of the agile community in the past decade, a body of knowledge more suited to the reality of software development has emerged. Many books explaining agility have become must-read books for anyone interested in software development. A large majority of them contain at least a chapter or two on requirements, some almost totally dedicated to this topic. Because I believe these texts are important, I will include citations from them and reference them throughout this book.

I wrote this book to add to this body of knowledge. It is a compendium of the agile practices related to executable specifications. Executable specifications enable us to easily test the behavior of the software against the requirements. Throughout this book, I will explain how you can specify software when prerequisites are not clearly defined and when requirements are both difficult to grasp and constantly evolving. Software development practitioners will learn how to trawl requirements incrementally, step-by-step, using a vision-centric and an emergent iterative practice. They will also learn how to specify as you go while writing small chunks of requirements.

This book aims to explain the technical mechanisms needed to obtain the benefits of executable specifications. It not only provides a sound case for iterative discovery of requirements, it also goes one step further by teaching you how to connect the specifications with the software under construction. This whole process leads to the building of executable specifications.

It is important to recognize that even with the best intentions you cannot force agreement upon stakeholders. The following African proverb explains this succinctly: "You can't make grass grow faster by pulling on it." When knowledge is incomplete and needs are constantly changing, we cannot rely on approaches based on traditional engineering. Instead, it is critical that you emphasize empirical techniques based on the iterative discovery of the requirements. The objective sought is not only to solve the problem right, but also to solve the right problem—this is the paramount challenge of software construction.

This book is unique in that it teaches you how to connect requirements and architecture using executable specifications. You learn how to specify requirements as well as how to automate the requirements verification with a Scrum framework. As a result of reading this book, you can select a tool and start using executable specifications in future agile projects. Here are five advantages to reading this book:

- You can understand how the work of business analysts changes when transitioning from traditional to agile practices.
- You learn how to groom emergent requirements within the Scrum framework.
- You get insight about storyboarding and paper prototyping to improve conversations with stakeholders.
- You discover how to build an emergent design while ensuring implementation correctness at all times
- You can understand that software architects who are adopting agile practices are designing incrementally and concurrently with software development.

Who Should Read This Book?

Readers of this book have already adopted the Scrum framework or are transitioning to agile practices. They understand the fundamentals of agility but are unfamiliar with executable specifications. They want to understand why the executable specifications are useful and most important how to start with this new practice.

With the massive adoption of Scrum framework, the next major challenge facing agile teams is to integrate business analysts and architects as active teammates. Anyone who is a Scrum master, manager or decision maker who faces this challenge should read this book. In addition, all team members involved in agile projects will benefit from this book. It goes without saying that business analysts and software architects will be happy to find a book that directly addresses their concerns.

Advanced or expert agilists will be interested in the book's concise overview of executable specifications. They could use this book to successfully guide their teammates down this path. In addition, the terminology used throughout the book can help leaders to communicate effectively with their peers.

Road Map for This Book

Executable specifications require a change in mindset. This book focuses on this issue. Executable specifications help reduce the gap between what stakeholders want the software to do (the "What"), and what the software really does (the "How"). Executable specifications address requirements in a way that makes it easy for the development team to verify the software against the specifications and this as often as requirement changes occur.

To facilitate this change in mindset, this book offers a unique approach to the process that spans nine chapters:

- **Chapter 1: Solving the Right Problem**

  This chapter explains the need to respond efficiently to the constantly changing requirements using iterative discovery and executable specifications.

- **Chapter 2: Relying on a Stable Foundation**

  This chapter explains how to identify what will hardly change: the core certainties on which the team should rely. Those certainties are not requirements. They are high-level guardrails that ensure a solution can be built. They create a stable foundation to ensure that an iterative requirements discovery is possible.

- **Chapter 3: Discovering Through Short Feedback Loops and Stakeholders' Desirements**

  This chapter shows that to tackle uncertainties, teams must discover stakeholders' desires and requirements (desirements) through short feedback loops.

- **Chapter 4: Expressing Desirements with User Stories**

  This chapter teaches you how to express desirements with user stories and how to record them using the product backlog.

- **Chapter 5: Refining User Stories by Grooming the Product Backlog**

  This chapter explains how to groom the product backlog so that you can plan sprints that can increase the likelihood of success of the feedback loops.

- **Chapter 6: Confirming User Stories with Scenarios**

  This chapter demonstrates how to confirm user stories by scripting behaviors with scenarios.

- **Chapter 7: Automating Confirmation with Tests**

  This chapter explains how to turn scenarios into automated tests so that you can easily confirm the expected behavior of the software against the evolving specifications.

- **Chapter 8: Addressing Nonfunctional Requirements**

  This chapter teaches you how to ensure quality software by specifying nonfunctional requirements.

- **Chapter 9: Conclusion**

  This last chapter summarizes the key elements of the book.