UNLOCKING AGILE'S MISSED POTENTIAL

ROBERT WEBBER

IEEE PRESS

IEEE computer society

WILEY

Table of Contents

List of Tables

Chapter 3

Chapter 4

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Chapter 15

Chapter 16

Appendix A

Appendix B

List of Illustrations

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 8

Chapter 9

Chapter 11

Chapter 12

Chapter 13

Chapter 14

Chapter 16

Appendix A

Appendix C

Appendix E

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

About IEEE Computer Society

IEEE Computer Society is the world's leading computing membership organization and the trusted information and career-development source for a global workforce of technology leaders including: professors, researchers, software engineers, IT professionals, employers, and students. The unmatched source for technology information, inspiration, and collaboration, the IEEE Computer Society is the source that computing professionals trust to provide high-quality, state-of-the-art information on an on-demand basis. The Computer Society provides a wide range of forums for top minds to come together, including technical conferences, publications, and a comprehensive digital library, unique training webinars, professional training, and the Tech Leader Training Partner Program to help organizations increase their staff's technical knowledge and expertise, as well as the personalized information tool my Computer. To find out more about the community for technology leaders, visit http://www.computer.org.

IEEE/Wiley Partnership

The IEEE Computer Society and Wiley partnership allows the CS Press authored book program to produce a number of exciting new titles in areas of computer science, computing, and networking with a special focus on software engineering. IEEE Computer Society members receive a 35% discount on Wiley titles by using their member discount code. Please contact IEEE Press for details.

To submit questions about the program or send proposals, please contact Mary Hatcher, Editor, Wiley-

IEEE Press: Email: mhatcher@wiley.com, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774.

Unlocking Agile's Missed Potential

Robert Webber

IEEE computer society

IEEE PRESS

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data Applied for:

Hardback ISBN: 9781119849087

Cover Design: Wiley

Cover Image: © Studio Romantic/Shutterstock.com

Author Biography

Robert Webber introduces a novel approach for Agile portfolio management and planning based on his extensive experience in leading software development organizations at GTE, AT&T, and three successful startups. His roles as VPs of engineering and product management, CEO, and consultant for Fortune 500 companies provided the technical and business perspectives to facilitate Agile planning while improving schedule and financial predictability.

Robert has an Electrical Engineering degree (honors) from the University of British Columbia. He holds five patents and received an Academy of Motion Pictures Arts and Sciences Technical Achievement Award for his innovative work on cable-mounted cameras.

Foreword

As the author of *Code Complete* I have been asked to write forewords for other people's books more times than I can count. I say “no” nearly every time. If readers see a foreword by the author of *Code Complete*, I believe they will expect the book to be as good as *Code Complete*. I rarely have enough confidence in other books to make that promise.

This book is a rare exception.

I have known Bob Webber for more than 20 years, and during that time we have had hundreds of discussions about the topics in this book. We've noted the unique ways that software executives see the world – often correctly, and sometimes flawed. We have shared observations about the relationship between the typical software development organization and product management, sales, and top executives. We've analyzed the strengths and weaknesses of Agile implementations – based to some degree on book knowledge, but based more on what we have seen with the numerous teams and companies we have helped first hand. We've puzzled about the reasons that Agile so often fails to live up to expectations. We have had many lengthy conversations about managing portfolios of software projects, releases, value streams, and features – and how all that relates to Agile development.

This book begins with Bob's identification of a major problem: The typical organization's commitment to Waterfall planning undermines its ability to be truly agile, and that in turn prevents it from realizing the benefits of Agile development. Don't mistake this for the common doctrinaire argument, “You can't realize the benefits of

Agile unless every remote corner of your organization becomes Agile.” Bob's analysis is more focused and far more nuanced – it has to do with the specific interactions between top executives, product management, and software development that give rise to a destructive mismatch between Waterfall planning at the business level that eviscerates the use of Agile practices in the software organization.

After his clear explanations of this fatal weakness in Agile implementations, Bob presents an innovative solution, which is to treat software initiatives as a portfolio of investments, both small and large, and to prioritize them in a specific way that maximizes economic return. The key is in how investments are sequenced – some sequences produce far higher return than others, even when the work in total is all the same. Common sense turns out to be a weak method for sequencing investments, and this book presents a sophisticated approach that works markedly better. There is some math in that section, but it's worth it. When this approach is used, prioritization of “features” and initiatives becomes more objective and easier. Historical friction between engineering and product management dissipates. The organization becomes more successful.

The investment prioritization algorithm in this book – the sequencing – is a core that supports improved organizational performance in other areas. The core provides guidance for roadmap planning, increased productivity, and increased innovation. These are not small benefits. Roadmap planning is perhaps the most common Achilles heel I have seen in Agile implementations. Many organizations treat increased productivity and increased innovation as primary goals. Bob's software investment portfolio approach supports them all.

Despite containing a significant amount of math and some economic discussion, this is not an academic book. It is based on real-world experience, and it provides a step-by-step guide that you can use to implement this software investment portfolio approach in your organization. Bob provides practical resources such as an investment planning template and an extended case study that illustrates how all the concepts and techniques work together in practice.

Most books can't measure up to the promise of *Code Complete*, but the approach in this book can. Most organizations perform their software work in inefficient ways. Simply by doing the same work in a different order – changing the sequence – organizations can increase the returns on their software investments dramatically.

From my conversations with Bob over more than two decades, I can say with certainty that there is no one in the world who has spent more time understanding how to construct a portfolio of software initiatives and how to optimize the returns on such portfolios. This book describes Bob's best thinking on the subject, and I recommend it whole heartedly.

Steve McConnell, author of *Code Complete*

Preface

This is a journey that began in 2011 when I started a second career as a product development consultant. I wanted to leave behind the pressure of delivering software products and transition to a role that would allow me more time flexibility to pursue my interests. I wouldn't change anything in my first career. It provided me with a perspective from the engineering and business side to solve a software industry problem – the missed expectations of Agile development.

I would never have become a consultant if it wasn't for Steve McConnell, an icon of software engineering. I had read his books and admired how he updated proven software engineering practices for the current generation of software developers. I became a client of Steve's training and consulting company, Construx Software.

When I decided to leave the corporate treadmill, Construx was the only company for which I wanted to consult. Steve let me determine how I could best contribute and gave me the flexibility to focus on my interests. It gave me the opportunity to apply my knowledge and experience to help software companies build higher value software and release it faster. It also provided me with insight into what was really going on in Agile, and it wasn't the Agile I was reading about in books.

This was a period when everyone was “going Agile.” I saw larger companies stumble when they tried to apply a simplistic Scrum model designed for small organizations. Product development organizations had the biggest challenges. Where is the product owner who knows everything about everything in an international organization with a complex domain and diverse customers? And how can engineering do real Agile development when product management plans large releases with fixed schedules stuffed with features?

My consulting assessments always pointed out that organizations were not really doing Agile. A basic tenet of Scrum is that teams have time to incorporate feedback and correct defects during sprint development. Instead, teams were still driven by imposed schedules and barely had time to create basic functionality. Teams were demotivated from continually missing schedules. Sprints were not “potentially shippable.” Defects mounted up during sprints which led to major rework in a “hardening sprint.” Engineers were feature implementors instead of providing innovative solutions to problems conveyed to them in user stories and epics.

The value of releasing faster seemed obvious. Why did these organizations continue to use traditional Waterfall release planning methods with fixed schedules and content? Waterfall planning hadn't ever achieved the predictability they sought. How did they think that Agile roadmap commitments could be made prior to requirements development given that requirements were allowed to change?

Companies continue to use Waterfall planning because they are forced to make multiyear financial commitments. The sales organization needs to know the products and features they can sell before they will commit to revenue. And projects need financial justification based on delivery schedule and content commitments. Internal operations groups must budget with the expectation of increased productivity dependent on new features. Agile didn't address these issues. The somewhat flippant response of the Agile community was, “The business needs to be more

Agile.” Even though Waterfall planning leaves a lot to be desired in terms of predictability, it is the only way product managers know how to plan.

Some companies implemented the SAFe Agile framework to integrate business planning and Agile development. However, SAFe just institutionalized feature-based Waterfall planning. Product managers liked it because they didn't have to change the way they planned large releases at the feature-level. SAFe included an option for Lean Canvas planning but is not widely used in larger organizations. It's difficult to get funding approved for market experimentation. The business wants to see the releases and features they are paying for before funding development.

I've experienced a few epiphanies in my career. One occurred at the annual Construx software leadership summit in 2011 that influenced this book. Donald Reinertsen, author of *The Principles of Product Development Flow* [1], was a keynote speaker. His book presents a queue-based model of software development analogous to manufacturing organizations. The book includes a concept that can potentially change the way software is planned. It's called “Cost of Delay.”

Cost of Delay is a simple yet elegant perspective of the software development process: “How much more money would you make if you could release your software one month earlier?” Reinertsen went on to show that prioritizing projects based on something called, “Weighted Shortest Job First” (WSJF), would optimize income. WSJF is the Cost of Delay divided by the project cycle time. It essentially says that companies make more money when they limit functionality and release and earn income faster. I recognized that WSJF can facilitate the mythical “Minimum Marketable Feature Set” (MMFS) advocated by

Agile. I say mythical because sales organizations refused to compromise on functionality. They wanted anything and everything that could possibly lead to a sale in any country.

Thus began the journey to this book. Cost of Delay and WSJF were great concepts, but they were not widely adopted. Companies found it difficult to calculate Cost of Delay for non-linear post-release income profiles. Reinertsen's formula works for linear income profiles with constant income per month. I recognized that WSJF had to be simplified and product managers need to have methods and tools to simplify the calculations for any income profile.

My conclusion was that most companies have not achieved the potential of Agile because they are not really doing Agile the way it was envisioned. They are still trying to force-fit Agile into a Waterfall planning framework. This book enables these organizations to reap the benefits of their investments in Agile development by incorporating Agile planning and deployment. It also lets larger organizations replicate the powerful dynamics of a startup instead of the divisiveness and friction I observed between product management and engineering. The book describes how larger organization can build collaborative Agile product teams.

There is one other major influence that helped shape my solution. Tom Gilb has a long history of contributing to the software industry in the areas of software metrics, software inspections, and evolutionary development processes. Tom gave a talk on stakeholder value analysis at Construx in 2011. His method provides a way to dig down to what users really want before getting into requirements detail. It was the perfect fit for my proposed approach to Agile planning.

I want to thank Steve McConnell for his patience and support for this journey. Steve is a deep thinker who kept me on track by challenging many of my assertions. His philosophy of hiring smart people and letting them figure out how they can best contribute gave me the freedom to pursue my ideas. I also had the benefit of stimulating collaborations with other Construx consultants. Steve's recent book, *More Effective Agile* [2], is a useful and practical guide for engineering leaders that complements my work.

Don Reinertsen's book and principles provided the foundation for this new approach for Agile planning. He is an inspirational speaker with the analytical mind of an electrical engineer. I had the pleasure of talking to Don about some of my ideas at the Construx leadership summit he attended two years after his first talk. I highly recommend reading, *The Principles of Product Development Flow*. WSJF was a way to illustrate the value of reducing cycle time. The book contains much more. He goes on to show how cycle times can be drastically reduced by managing queues that build up in software development.

I have one other person to thank – my wife, Susan. In addition to being a great partner and mother, her PhD in Psychology and experience in leading change at high-tech companies gave me insight into the “soft” side of software development that needs to be addressed to realize the full potential of Agile. We can put all the procedures and processes in place, but little happens without human motivation. This book explains how Agile can create high-performance self-directed teams, yet how it has been squandered by the constraints and controls from traditional development, especially those driven by the perpetuation of Waterfall planning.

And, of course, I want to thank Susan for infinite patience and support for the years I've spent developing and refining the Investment approach.

References

1 Reinertsen, D.G. (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing.

2 McConnell, S. (2019). *More Effective Agile: A Roadmap for Software Leaders*. Construx Press.

Introduction

The Lost Potential of Agile Development

I wrote this book because I want to see Agile development fulfill its promises. My assertion is that organizations that retain Waterfall planning have not obtained the true benefits of Agile. I found that organizations often referred to their software development frameworks as a “hybrid” approach. After almost a decade of software development consulting, I learned to recognize that the term “hybrid” was a sure sign of a failed Agile implementation. It meant that engineering was still held to fixed schedule and release content commitments. This book will show that Waterfall planning is the antithesis of Agile. However, product managers can now adopt new planning methods that support Agile development while meeting the business need for predictability. Engineering can finally do Agile with content flexibility.

After a successful career in software development, I joined Construx Software as a consultant. I gave seminars on requirements, innovation, and project management. I also conducted software development organizational assessments, which included many Fortune 500 companies. I reviewed their practices and organizational structures to recommend how to improve software development productivity and predictability, and to reduce time-to-market. I had the opportunity to see what was really happening “under the hood” in Agile.

While at Construx, I also had the opportunity to moderate group sessions at the annual Construx Leadership Summit