LEARNING MADE EASY

4th Edition

Agile Project Management

for dummies®

A Wiley Brand

Deliver value more quickly and effectively

Learn why diverse teams are more profitable

Put agile techniques into action with AI

Mark C. Layton
CST, PMP, MBA²

Steven J Ostermiller
CST, CAL, PMP

Dean J. Kynaston
CSP-SM, CSP-PO, MBA

Agile Project Management

4th Edition

by Mark C. Layton, Steven J Ostermiller, and Dean J. Kynaston

for dummies®
A Wiley Brand

Agile Project Management For Dummies®, 4th Edition

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Published simultaneously in Canada

Library of Congress Control Number: 2025938069

ISBN 978-1-394-33209-0 (pbk); ISBN 978-1-394-33211-3 (ebk); ISBN 978-1-394-33210-6 (ebk)

Contents at a Glance

Table of Contents

Introduction

Welcome to *Agile Project Management For Dummies*, 4th Edition. Agile project management has grown to be as common as any management technique for product development — and not only software product development. For two and a half decades, we have trained and coached companies big and small, all over the world, about how to become more nimble, adaptive, and responsive in both the development of their products and their organizations — in other words, how to become more agile. Through this work, we found there was a need to write a digestible guide that anyone, regardless of experience, could understand.

About This Book

Agile Project Management For Dummies, 4th Edition is more than just an introduction to agile practices and approaches; you also discover the steps to become more agile in mindset and behavior. The material here goes beyond theory and is meant to be a field guide for all experience levels, giving you the tools and information you need to be successful with agile techniques in the trenches of product development.

Foolish Assumptions

This book was written as a reference guide for anyone wanting to learn more about business agility. If you strive to be more agile in responding to customer needs and problems — whether or not you're an organizational leader, a project manager, a product manager, a member of a product team, an agile enthusiast, or a product stakeholder — this book will help you on your journey.

Regardless of your experience or level of familiarity, this book provides insights you may find helpful. We hope it brings clarity to any confusion or myths regarding agile product development you may have encountered.

Icons Used in This Book

Throughout this book, you'll find the following icons.


TIP

Tips are points to help you along your agile product development journey. Tips can save you time and help you quickly understand a particular topic, so when you see them, take a look!


REMEMBER

The Remember icon is a reminder of something you may have seen in past chapters. It also may be a reminder of a commonsense principle that is easily forgotten. These icons can help jog your memory when an important term or concept appears.


WARNING

The Warning icon indicates that you want to watch out for a certain action or behavior. Read these to steer clear of big problems!


TECHNICAL STUFF

The Technical Stuff icon indicates information that is interesting but not essential to the text. If you see a Technical Stuff icon, you don't need to read it to understand agile product development, but the information there might just pique your interest.

Beyond the Book

Although this book broadly covers the agile project management spectrum, we can cover only so much in a set number of pages! If you find yourself at the end of this book thinking, "This was an amazing book! Where can I learn more about how to advance my products under an agile approach?" check out Chapter 24 or head over to `www.dummies.com` for more resources.

We've provided a cheat sheet for tips on assessing your current product development efforts in relation to agile principles as well as free tools for managing projects using agile techniques. To get to the cheat sheet, go to `www.dummies.com`, and then type *Agile Project Management For Dummies Cheat Sheet* in the Search box. This is also where you'll find any significant updates or changes that occur between editions of this book.

Where to Go from Here

We wrote this book so that you could read it in just about any order. Depending on your role, you may want to pay extra attention to the appropriate sections of the book. For example:

- If you're just starting to learn about product development and agile approaches, start with Chapter 1 and read the book straight through to the end.
- If you're a member of a product team and want to know the basics of agile product development, check out the information in Part 3 (Chapters 9 through 12).
- If you're a project manager transitioning to agile approaches to product development, you may be interested to learn how agile techniques improve the management of time, cost, scope, procurement, quality, and risk. Review Part 4 (Chapters 13 through 17).
- If you know the basics of agile product development and are looking at bringing agile practices to your company or expanding your agile footprint across your organization, Part 5 (Chapters 18 through 20) will provide you with helpful information.

1 Understanding Agility

IN THIS PART . . .

Understand the flaws and weaknesses in historical approaches to project management.

Become acquainted with the foundation of agile product development: the Agile Manifesto and the 12 Agile Principles.

Discover the advantages that your products, projects, teams, customers, and organization can gain from adopting agile techniques.

Understand why agile techniques make the customer central to every decision, functionality, and problem.

IN THIS CHAPTER

- Understanding why project management needs to change
- Seeing how agile project management is becoming agile product management
- Finding out about agile product development

Chapter 1
Modernizing Project Management

Agile is a descriptor — an adjective synonymous with adaptive, nimble, flexible, responsive, and lightweight. When we speak of becoming more agile as an organization or taking an agile approach, we are referring to all these terms.

Agile describes an organization's capability to sense and respond or the approach a project team takes to adapt a plan to a customer's needs or to evolve with changing technologies. *Agility* in project management focuses on early delivery of customer value as well as continuous improvement of the product being created and the processes used to create the product. At its essence, an agile mindset seeks customer-centric scope flexibility, team input, and the delivery of well-tested, valuable products that reflect customer needs.

In this chapter, you find out why agile approaches to software development project management emerged in the mid-1990s and why these ways of working and thinking have caught the attention of project managers, customers, and executives. While business agility is popular in software product development, agile values, principles, and techniques (which you learn about in the following

chapters) apply in a multitude of industries and applications — not just software. This chapter also explains the advantages of agile approaches over historical project management methodologies.

Project Management Needed a Makeover

A *project* is a planned program of work that requires a definitive amount of time, effort, and planning to complete. Projects have goals and objectives and often must be completed in some fixed period of time and within a certain budget.

Because you're reading this book, you're likely a project manager or someone who initiates projects, works on projects, or is affected by projects in some way. Agile approaches are a response to the need to modernize project management. To understand how agile approaches are revolutionizing product development, it helps to know a little about the history and purpose of project management and the issues that projects face today.

The origins of modern project management

Projects have been around since ancient times, from the Great Wall of China to the invention of the internet. As a formal discipline, project management as we know it has been around only since the middle of the twentieth century. Around the time of World War II, researchers around the world were making major advances in building and programming computers, mostly for the United States military. To complete those projects, they started creating formal project management processes. The first processes were based on step-by-step manufacturing models the United States military used during World War II.

People in the computing field adopted these step-based manufacturing processes because early computer-related projects relied heavily on hardware, with computers that filled entire rooms. Software, by contrast, was a smaller part of computer projects. In the 1940s and 1950s, computers could have thousands of physical vacuum tubes but fewer than 30 lines of programming code. The 1940s manufacturing process used on these initial computers is the foundation of what has become known as *waterfall* in project management.

In 1970, a computer scientist named Winston Royce wrote "Managing the Development of Large Software Systems," an article for IEEE (Institute of Electrical and Electronics Engineers) that described the phases of a waterfall in project management. The term *waterfall* was coined later, but the phases, even if they are sometimes titled differently, are essentially the same as defined by Royce:

1. Requirements
2. Design
3. Development
4. Integration
5. Testing
6. Deployment

On waterfall projects, you move to the next phase only when the prior one is complete — hence the name waterfall.

Pure waterfall project management — completing each step in full before moving to the next step — is actually a misinterpretation of Royce's suggestions. Royce identified that this approach was inherently risky and recommended developing and testing within iterations to create products — suggestions that were overlooked by many organizations that adopted waterfall.

Variations of waterfall were the most common project management approaches in software development until it was surpassed by improved approaches based on agile values and principles around 2008.

The problem with the status quo

Computer technology has, of course, changed a great deal since the last century. Many people have a computer on their wrist with more power, memory, and capabilities than the largest, most expensive machine that existed when people first started using waterfall methodologies.

At the same time, the people using computers have changed as well. Instead of creating behemoth machines with minimal programs for a few researchers and the military, people create hardware and software for the general public. In most countries, almost everyone uses a tablet or smartphone, directly or indirectly, every day. Software runs our cars, our appliances, our homes; it provides our daily information and daily entertainment. Even young children use computers — generative artificial intelligence is helping them develop mobile apps and professional-looking graphics and videos. The demand for newer, better products is constant and rapidly evolving.

Somehow, during all this growth of technology, processes stagnated. Software developers are still using project management methodologies from the 1950s, and all these approaches were derived from manufacturing processes meant for the hardware-heavy computers of the mid-twentieth century.

Today, traditional projects that do succeed often suffer from one problem: *scope bloat*, the introduction of unnecessary product features. Think about the software products you use every day. For example, the word-processing program we're typing on right now has many features and tools. Even though we write with this program every day, we use only some of the features all the time. We use other elements less frequently. And we have never used quite a few features — and come to think of it, we don't know anyone else who has used them, either. The features that few people use are the result of scope bloat.

Scope bloat appears in all kinds of software, from complex enterprise applications to websites that everyone uses. Figure 1-1 shows data from a Standish Group study that illustrates just how common scope bloat is. In the figure, you can see that 80 percent of requested features are infrequently or never used.

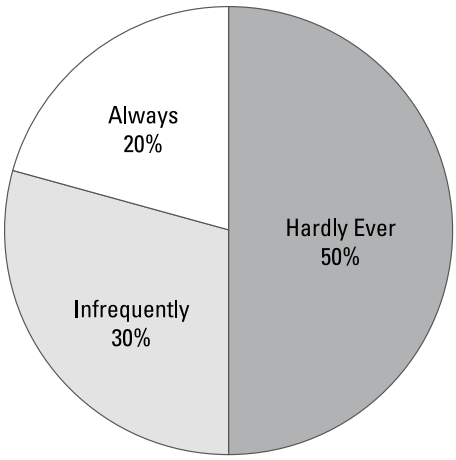

FIGURE 1-1: Actual use of requested software features.

Figure 1-1 illustrates the waste resulting from traditional project management processes. Traditional project managers and stakeholders believe that change is not welcome mid-project, so their best chance of getting a potentially desirable feature is at the start of a project. Therefore, they ask for

- Everything they need
- Everything they think they may need
- Everything they want
- Everything they think they may want

The result is the bloat in features that results in the statistics in Figure 1-1.

The problems associated with using outdated management and development approaches are not trivial. These problems waste billions of dollars a year. It's no wonder billions of dollars are lost in waterfall project failures (see the sidebar, "Software project success and failure"), especially if success means every feature delivered on time, on budget, and with perfect quality.

Over the past three decades, people working on projects have recognized the growing problems with traditional project management and have been working to create a better model.

Modern impacts of artificial intelligence

Emerging technologies affect project management and product development, and artificial intelligence has had one of the most significant effects we've seen to date. Artificial intelligence, or AI, is the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages. AI is transforming project management by reducing the administrative work required by teams, enabling them to redirect their effort, time, and thinking to be more creative, lateral, and inclusive.

While AI creates many benefits, there are also precautionary implications. AI isn't here to replace everything people can do. For instance, AI can handle scheduling, task tracking, and data analysis, but human intuition from experience is vital in high-stakes decisions. The benefits and precautions will be discussed throughout this book.

Modern impacts of diversity and inclusion

Rapidly advancing technologies such as AI as well as increased mobility and access to information have not only opened us up to more diverse experiences and social circles but have also created more diversity and a need for inclusion of opinions, backgrounds, experiences, cultures, styles, and perspectives.

Although we've been in a global economy for centuries, product development organizations were traditionally made up of people in a specific geographical region or professional network. The product problems we're solving today span continental boundaries and disciplines. We no longer work every day only with people who look and think like we do.

When we harness this diversity and effectively engage people from multiple perspectives, we're finding greater alignment to the needs of the customer. Leveraging those differences is essential for team and organizational success.

SOFTWARE PROJECT SUCCESS AND FAILURE

Stagnation in traditional project management approaches is catching up with the software industry. In 2020, a software statistical company called the Standish Group did a study on the success and failure rates of 50,000 projects globally. The results of the study showed that

- *66 percent of technology projects end in partial or total failure.* Although larger projects are more prone to failure, even smaller projects fail 10 percent of the time.
- *31 percent of US projects failed outright.* The projects were cancelled before they finished and did not result in any product releases. Unless something important was learned to inform a successful pivot for these organizations, these projects delivered no value.
- *53 percent of US projects were challenged.* The projects were completed but had gaps between expected and actual cost, time, quality, or a combination of these elements. The average difference between the expected and actual project results — looking at time, cost, and features not delivered — was well over 100 percent.
- *Only 16 percent of US projects succeeded.* The projects were completed and delivered the expected product in the originally expected time and budget.

Of the hundreds of billions of dollars spent on product development in the US alone, billions of dollars were wasted on projects that never deployed a single piece of functionality.

Introducing Agile Project Management

The seeds for agile values, principles, and practices have been around for a long time and are simply a codification of common sense. Figure 1-2 shows a quick history, dating to the 1930s with Walter Sherwart's Plan-Do-Study-Act (PDSA) approach to project quality.

In 1986, Hirotaka Takeuchi and Ikujiro Nonaka published an article called "The New New Product Development Game" in the *Harvard Business Review.* Takeuchi and Nonaka's article described a rapid, flexible development strategy to meet fast-paced product demands. This article first paired the term *scrum* with product development. (*Scrum* refers to a player formation in rugby.) Scrum eventually became one of the most popular agile frameworks for delivering value to customers.

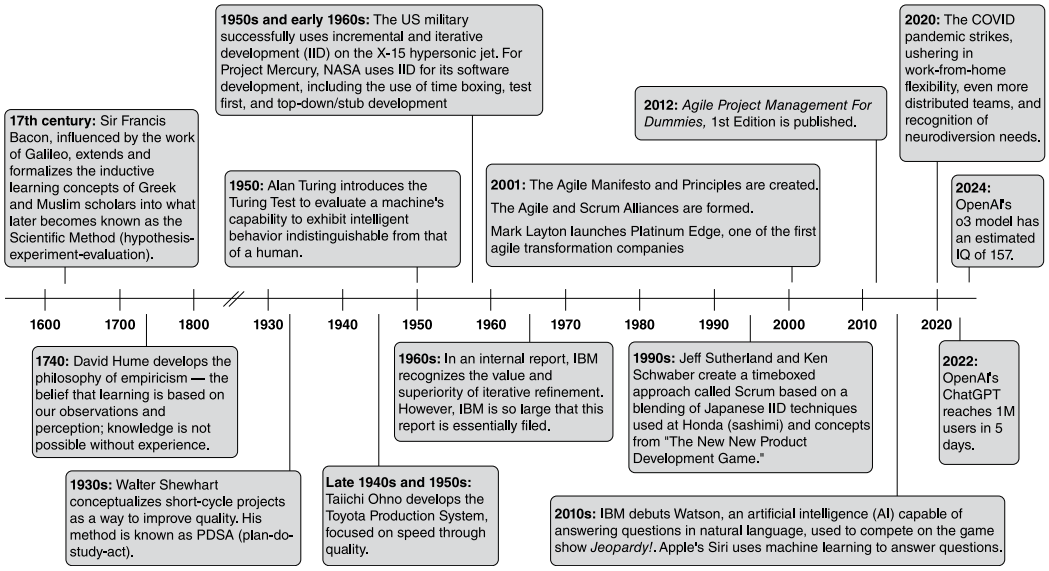

FIGURE 1-2: Agile project management timeline.

In 2001, a group of software and project experts got together to talk about what their successful experiments in modernizing projects had in common. This group created the *Manifesto for Agile Software Development* (commonly referred to as the Agile Manifesto), a statement of values for successful software development:

> Manifesto for Agile Software Development*
>
> We are uncovering better ways of developing
>
> software by doing it and helping others do it.
>
> Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
>
> **Working software** over comprehensive documentation
>
> **Customer collaboration** over contract negotiation
>
> **Responding to change** over following a plan
>
> That is, while there is value in the items on
>
> the right, we value the items on the left more.

These experts also created 12 principles behind the Agile Manifesto that help support the values in the Agile Manifesto. We list the Agile Principles and describe the Agile Manifesto in more detail in Chapter 2.

Agile, in product development terms, is a descriptor for approaches that focus on people, communications, the customer, and flexibility. If you're looking for *the* agile methodology, you won't find it. However, some methodologies (for example, Extreme Programming), frameworks (for example, Scrum), techniques (for example, user stories), and tools (for example, physical task boards) have one thing in common: adherence to the Agile Manifesto and the 12 Agile Principles.

TIP

When you think of agile, think healthy. Healthy organizations and teams are agile, resilient, nimble, adaptive and responsive.