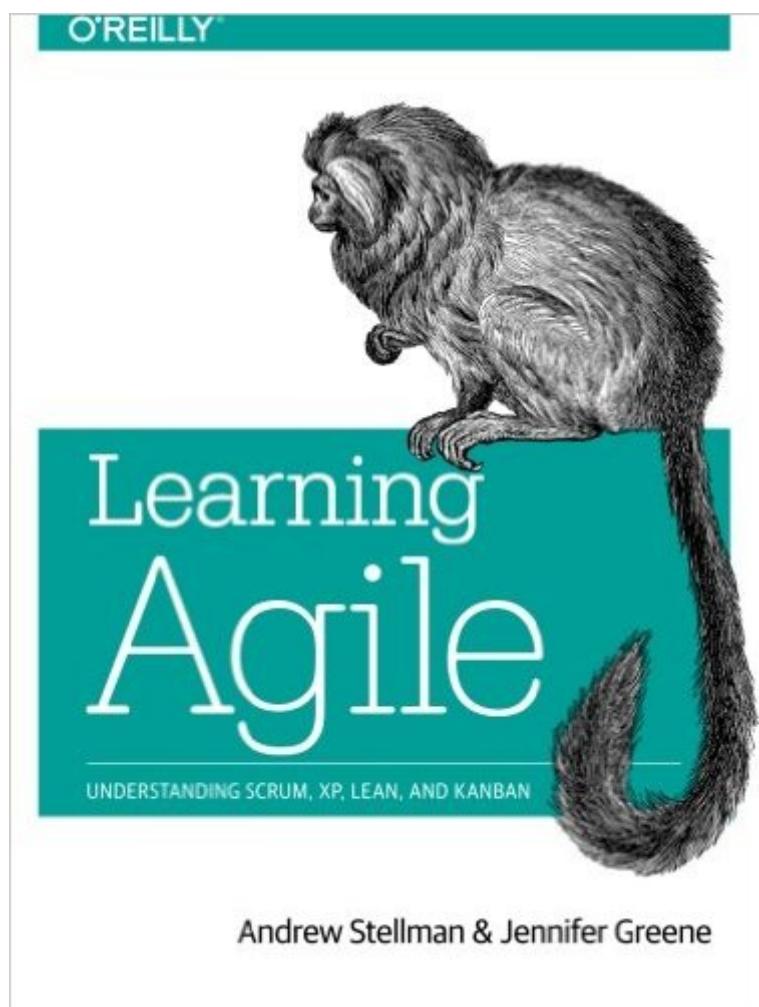


The book was found

Learning Agile: Understanding Scrum, XP, Lean, And Kanban



Synopsis

Learning Agile is a comprehensive guide to the most popular agile methods, written in a light and engaging style that makes it easy for you to learn. Agile has revolutionized the way teams approach software development, but with dozens of agile methodologies to choose from, the decision to "go agile" can be tricky. This practical book helps you sort it out, first by grounding you in agile's underlying principles, then by describing four specific--and well-used--agile methods: Scrum, extreme programming (XP), Lean, and Kanban. Each method focuses on a different area of development, but they all aim to change your team's mindset--from individuals who simply follow a plan to a cohesive group that makes decisions together. Whether you're considering agile for the first time, or trying it again, you'll learn how to choose a method that best fits your team and your company. Understand the purpose behind agile's core values and principles. Learn Scrum's emphasis on project management, self-organization, and collective commitment. Focus on software design and architecture with XP practices such as test-first and pair programming. Use Lean thinking to empower your team, eliminate waste, and deliver software fast. Learn how Kanban's practices help you deliver great software by managing flow. Adopt agile practices and principles with an agile coach.

Book Information

Paperback: 420 pages

Publisher: O'Reilly Media; 1 edition (December 10, 2013)

Language: English

ISBN-10: 1449331920

ISBN-13: 978-1449331924

Product Dimensions: 7 x 1 x 9.2 inches

Shipping Weight: 1.5 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars 37 customer reviews

Best Sellers Rank: #94,724 in Books (See Top 100 in Books) #102 in Books > Textbooks > Computer Science > Software Design & Engineering #121 in Books > Business & Money > Management & Leadership > Project Management > Business #135 in Books > Computers & Technology > Business Technology > Software > Enterprise Applications

Customer Reviews

Understanding Scrum, XP, Lean, and Kanban

This book is called Learning Agile because we really want you to learn agile. We've spent the last 20+ years working with real teams building real software for real users day in and day out. We've also spent the last 10+ years writing books about building software (including two very successful books in the O'Reilly Head First series about managing projects and learning to code). This experience has helped us find many different ways to get complex and technical ideas into your brain without boring you to death. We've done our best to take this material and make it as interesting and engaging as possible. We use narratives and illustrations, include key points and coaching tips, and answer many frequently asked questions that routinely come up when teams try to implement agile in the real world on their own teams— and all of these things can help you and your team learn agile quickly so that you can build and deliver better, more valuable software, and do it faster than before. Who we wrote this book for Do any of these scenarios describe you and your team? You tried an agile practice, but it didn't really work out. Maybe you implemented daily standup meetings, and now your team meets every day--but you still get blindsided by problems and miss deadlines. Or you started writing user stories and reviewing them with your team and stakeholders, but your developers still find themselves dealing with just as many last-minute changes to add extra features that continue to pop up. Or maybe your team tried to go agile wholesale by adopting a methodology like Scrum or XP, but it seems somehow "empty"--like everyone is going through the "required" motions, but your projects are only marginally improving. Or maybe you haven't tried agile yet, but you recognize that your team is facing serious challenges, and you don't know where to start. You're hoping that agile will help you with those demanding users who constantly change their minds. Each change your users make requires more work for your team, and leads to "duct tape and paperclips" spaghetti code solutions that make the software increasingly fragile and unmaintainable. It could be that your projects are simply controlled chaos; the primary way software is delivered is through long hours and personal heroics, and you think that agile may offer your team a way out. What if you're an executive who's worried that teams working on important projects will fail to deliver? Maybe you've heard about agile, but you don't really know what it means. Can you simply tell your team to adopt agile? Or will you need to change your own mindset along with the team? If any of those situations is familiar to you, and you want to improve how your team works, this book will help. We explain the agile methodologies: why they're designed the way they are, what problems they address, and the values, principles, and ideas that they embody. By giving you the "why" in addition to the "how," we'll help you to recognize the principles that apply to the particular development problems specific to your team, company, and projects. And we'll show you how to use that information to guide your choice of methodologies and practices.

What we want for you: We want you to understand the ideas that drive effective agile teams, and the values and principles that bring them together. We want you to understand the most popular agile schools of thought--Scrum, XP, Lean, and Kanban--and how they can all be agile, even though they're very different from each other. We want to teach you specific agile practices that you can apply to your projects today--but we also want to give you the framework of values and principles that you'll need to implement them effectively. We want to help you understand your own team and company better, so that you can choose an agile approach that matches your mindset (or comes as close as possible)--but also help you and your team start to learn a new way of thinking that will help you become a more effective agile team.

For anybody who works in an Agile team or wants to help guide their team to becoming more Agile, I would recommend reading this book. I think the biggest selling point for me was that it actually describes the differences between the various Agile methodologies. With all the different Agile methodologies that exists (and I'm sure there's many more beyond the 4 described in the book), it's easy to feel like you're swimming in an ocean of buzzwords. Luckily, I think the authors recognize the importance of actually "showing" you what Agile looks like and have vignettes sprinkled throughout the book to show you teams that are struggling and eventually teams that are succeeding with Agile. The first methodology they discuss is Scrum. They do this very intentionally because Scrum is probably the easiest Agile methodology to actually "adopt". It has clear practices and more closely resembles "traditional" software development process compared to the other methodologies. The second methodology they discuss is XP which I have heard is highly influential in the Agile field with luminaries such as Kent Beck and Martin Fowler as early leaders of the field, however to be honest I have never heard of anyone working in a software team they would describe as "XP". The most charitable way to interpret this is that many of the XP values, principles, and practices have trickled down into mainstream "Agile" consciousness. The easiest way to summarize XP is "embracing change" and the authors show you how they support that overarching goal through practices such as unit testing to facilitate refactoring and delaying decisions until the last moment. The last part describes the Lean and Kanban methodologies which are closely related. In short, they focus on continuous improvement. Before reading the book, I had heard about a Kanban board and the idea of moving tasks around different columns, but the real eye-opener was their emphasis on the importance of Work In Progress (WIP) limits. They show you how vital it is to have WIP limits and why you need to be careful of ignoring it (which oftentimes happen as they demonstrate). Probably my favorite part of the book is when they describe teams that partially apply

Agile methodologies while still retaining much of their legacy software development practice and end up achieving OK but "better than nothing" results. I think it's easy to think of Agile as the end all be all of good software development practice but it's a journey to actually achieving it and I like how they show you the realistic challenges of going from a traditional software development methodology to agile.

I've done a lot of Agile reading and training, and this is the best (along with Personal Kanban). Make this your go-to Agile reference text.

Everything about this book is about right. Clear, Lots of (fictional) scenarios where they follow a team story and demonstrate the pros/cons and value of the agile approach. The authors have gone out of their way to make this a fun read with applied theory and situations that help you lock in the theory much better. Nice job.

I picked up this book in advance of planned Agile training. Our company is moving from Waterfall to Agile and I had no background at all in Agile. I highly recommend this book. The high points: Very readable and thorough introduction to concepts and acronyms. Excellent examples and stories with characters who are working their way through using becoming Agile. AND Unlike most technical books this was something I read in two quick sittings (about 3 hours or so). When I was finished I felt like I understood the why and the what of Agile and was thoroughly prepared for my Agile training sessions. Here is the downside: there is not a lot work-flow process oriented information beyond key concepts and relationships. But, I wasn't ready for that to be honest so the presentation level was perfect in my case. If you are looking to understand how Agile handles very specific things you should probably look for a more advanced book. From my perspective, I couldn't have been more pleased. This was a great place to start.

I'm new to Agile and still reading this book but it makes the concepts so easy to understand and remember due to the illustrations and stories it uses. Would highly recommend to others.

Very digestible content for non programmers and PM.

Given that I've been familiar with Agile for at least 12-13 years and with Lean for nearly twice that long, I didn't expect that an introductory-level book would have a great deal to offer me except,

maybe, a refresher course. However, a number of the tools & techniques used in the book provide significant benefit even if you're already familiar with the material:^{*} Narratives, which do an excellent job of illustrating real-world software development problems and practical means of addressing them.^{*} The idea of "Better-Than-Not-Doing-It" results, which explains why many Agile implementations plateau after a short time. This is an elegant model that succinctly explains many sub-optimal process improvement efforts that I've observed over the years.^{*} Comparative methodology. Having a single resource that objectively compares and contrasts various Agile methodologies or "schools of thought" is very valuable. Reading about each approach back-to-back reinforces the common principles in a way that high-level overviews or single-methodology books rarely do. I also believe that the book will serve the beginner audience well because it's very well-organized and provides many references to additional learning resources. However, with my exposure to the material before reading, I can't state that opinion with 100% certainty. Similar to the authors' other books that I've read -- "Applied Software Project Management", "Head First C#", and "Beautiful Teams" -- "Learning Agile" is eclectic and has a practical, hands-on focus. Personally, I prefer that approach, especially when it integrates principles and lessons learned from such diverse subjects as the Toyota Production System, the Unix toolset, martial arts, and basketball (via the teachings of John Wooden). The authors even stress that a waterfall process CAN work, given the right conditions, which is rare for Agile-focused material. However, the eclectic/practical approach results in a tome that's probably not particularly well-suited for use as certification prep. The book takes roughly 8-10 hours to read. I recommend that you read topically-related chapters -- 2/3 (Agile values/principles), 4/5 (Scrum), 6/7 (XP), and 8/9 (Lean/Kanban) -- in the same sitting, or with as little time elapsed between readings as you can manage. The chapter pairs share Narratives, and the second chapter per pair builds upon ideas introduced in the previous chapter. All of the chapters reference and reinforce ideas that were introduced earlier in the book. This is partly because of the book's thematic cohesion and partly the result of a deliberate strategy for improving reading comprehension. Some suggestions that I think would add value to a future edition, if there is one:^{*} Further discussion of situations in which Agile isn't a great fit, preferably using Alistair Cockburn's Project Classification Scale [Defect Criticality x Team Size] for determining the process formality requirements of a project.^{*} More extensive coverage of the types of waste found in software development, bringing in wider-ranging ideas about wastefulness from Lean and other topics like the Unix philosophy, such as: unused employee creativity, over-design / over-specification, & solving the wrong problem the "Right" way. Disclosure: I was provided with a free review eBook in exchange for my honest feedback. Previously, I was a technical reviewer for "Applied Software

Project Management" and co-authored a chapter for "Beautiful Teams" (by the same authors). Also, more than a decade ago, I was a coworker of Andrew Stellman's for approximately a year.

[Download to continue reading...](#)

Agile Product Management: Product Owner: 27 Tips To Manage Your Product And Work With Scrum Teams (scrum, scrum master, agile development, agile software development) Agile Project Management: QuickStart Guide - The Simplified Beginners Guide To Agile Project Management (Agile Project Management, Agile Software Development, Agile Development, Scrum) Learning Agile: Understanding Scrum, XP, Lean, and Kanban Agile Project Management: Agile Revolution, Beyond Software Limits: A Practical Guide to Implementing Agile Outside Software Development (Agile Business Leadership, Book 4) Agile Software Development with Scrum (Series in Agile Software Development) Agile : Agile Project Management, A QuickStart Beginners 's Guide To Mastering Agile Project Management ! Agile Project Management with Kanban (Developer Best Practices) The Art of Scrummaging: A History, a Manual and a Law Dissertation on the Rugby Scrum - A Rugby Scrum Thesis Intermittent Fasting: Everything You Need to Know About Intermittent Fasting For Beginner to Expert – Build Lean Muscle and Change Your Life (Lean Lifestyle, Lean Muscle, Lose Fat) Scrum: a Breathtakingly Brief and Agile Introduction Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)) Agile Product Management with Scrum: Creating Products that Customers Love (Addison-Wesley Signature Series (Cohn)) Succeeding with Agile: Software Development Using Scrum Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition (Addison-Wesley Signature Series (Cohn)) Agile Testing: A Practical Guide for Testers and Agile Teams Head First Agile: A Brain-Friendly Guide to Agile and the PMI-ACP Certification Agile Project Management, A Complete Beginner's Guide To Agile Project Management! Agile Project Management QuickStart Guide: A Simplified Beginners Guide To Agile Project Management Lean Six Sigma: A Beginner's Guide to Understanding and Practicing Lean Six Sigma Agile and Lean Program Management: Scaling Collaboration Across the Organization

[Contact Us](#)

[DMCA](#)

[Privacy](#)

[FAQ & Help](#)