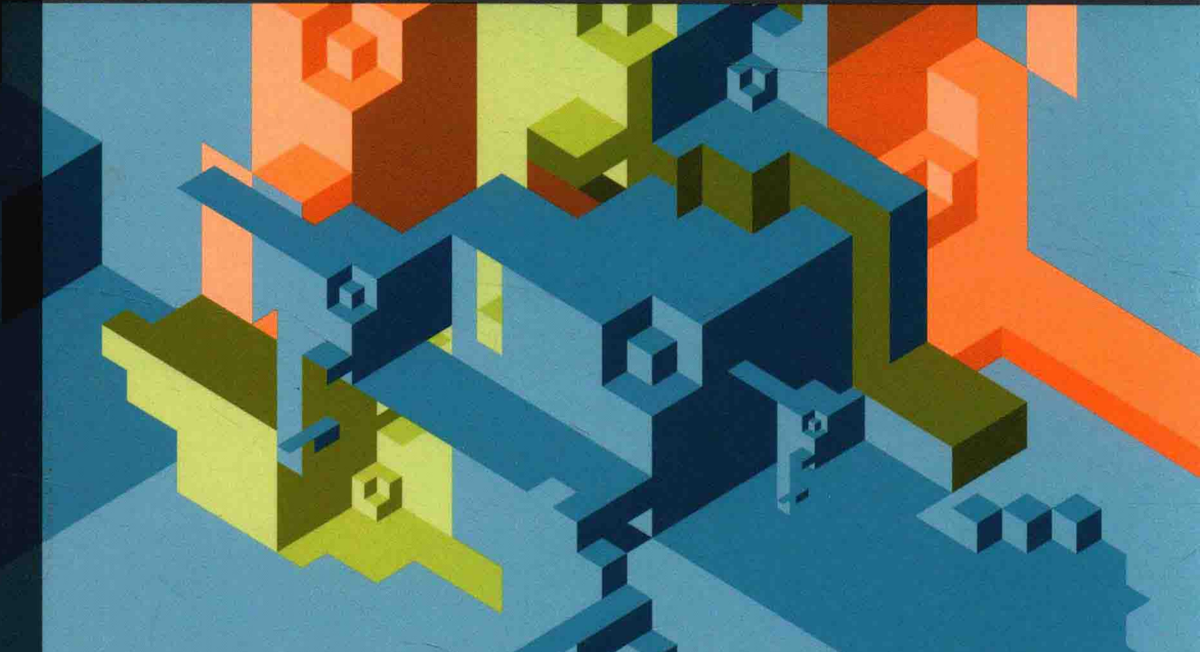
Microsoft

Agile Project Management with Kanban



Best Practices

Eric Brechner

Microsoft

Agile Project Management with Kanban

Eric Brechner

With a contribution from James Waletzky

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2014951864
ISBN: 978-0-7356-9895-6

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Devon Musgrave
Developmental Editor: Devon Musgrave
Project Editor: Devon Musgrave
Editorial Production: Rob Nance, John Pierce, and Carrie Wicks
Copyeditor: John Pierce
Indexer: Lucie Haskins
Cover: Twist Creative • Seattle

Praise for *Agile Project Management with Kanban*

"I have been fortunate to work closely with Eric for many years. In that time he has been one of the most productive, consistent, and efficient engineering leaders at Xbox. His philosophy and approach to software engineering are truly successful."

—Kareem Choudhry, Partner Director of Software Engineering for Xbox

"Eric easily explains why Kanban has proven itself as a useful method for managing and tracking complicated work. Don't expect this book to be an overview, however. Eric channels his deep understanding and experiences using Kanban at Microsoft to help you identify and avoid many of the common difficulties and risks when implementing Kanban."

—Richard Hundhausen, President, Accentient Inc.

"Learning how Xbox uses Kanban on large-scale development of their platform lends real credibility to the validity of the method. Eric Brechner is a hands-on software development management practitioner who tells it like it is—solid, practical, pragmatic advice from someone who does it for a living."

—David J. Anderson, Chairman, Lean Kanban Inc.

"As a software development coach, I continuously search for the perfect reference to pragmatically apply Kanban for continuous software delivery. Finally, my search is over."

—James Waletzky, Partner, Crosslake Technologies

"Kanban has been incredibly effective at helping our team in Xbox manage shifting priorities and requirements in a very demanding environment. The concepts covered in Agile Project Management with Kanban *give us the framework to process our work on a daily basis to give our customers the high-quality results they deserve."*

—Doug Thompson, Principal Program Manager, Xbox Engineering

"An exceptional book for those who want to deliver software with high quality, predictability, and flexibility. Eric's in-depth experience in the software industry has resulted in a realistic book that teaches Kanban in a simple and easy way. It is a must-read for every software professional!"

—Vijay Garg, Senior Program Manager, Xbox Engineering

Introduction

Dedicated to Corey Ladas, who invited me to see through new eyes.

I'm a professional software developer. I've been one for decades (currently with Xbox). I don't get paid to be a certified process geek. I don't get paid to be an evangelical process zealot. I get paid to deliver software that customers love, with high quality, on time, and at low cost.

I develop software in a highly volatile environment, where priorities, requirements, and expectations change daily. I develop software in a highly competitive environment—for market share among products and for compensation among peers. My software development peers inhabit this same world, regardless of where they work or what products they produce.

I'm always looking for advantages over my competition—ways that make my life easier and more productive while also resulting in better products for my customers. When I was at Bank Leumi, Jet Propulsion Laboratory, Graftek, and Silicon Graphics in the 1980s, I focused on what we now refer to as design patterns and unit testing. During the 1990s, while I was at Boeing and starting out at Microsoft, my teams and I tried Waterfall milestones and stabilization periods of different durations, T-shirt estimation, asserts, bug jail, continuous integration, and design and code reviews. In the 2000s, the Microsoft teams I managed experimented with Team Software Process, Scrum, code inspection, static analysis, planning poker, pair programming, and test-driven development. Now in the 2010s, I've found continuous deployment, Kanban, and a little nirvana.

Some of the methods I just listed may not be familiar to you. Most of the professional software developers I've known don't like experimenting with how they do their jobs. They find an approach that works for them, usually the one they learn from their first professional software team, and tend to stay with that approach.

Trying different methodologies is painful and has an initial drain on productivity, but it has enabled my teams to outperform those of my peers. My teams are significantly smaller than other teams doing similar work, yet they produce significantly more value at significantly higher quality in the same or less time. That's not because I've filled my teams with mythical developers who work twenty times faster than anyone else. (I've got good people, but they aren't mythical.) My teams simply get more value out of every hour.

You could experiment with all the methods I've tried, but that's time-consuming and tedious. Though my teams and I have learned from every experiment, not all have been of equal value. My current and former teams still use design patterns, unit testing, continuous integration, design and code reviews, static analysis, planning poker, pair programming, test-driven development, and continuous deployment to varying degrees. However, it was Scrum that had the biggest impact on team productivity and quality—that is, until we switched to Kanban four years ago. With Kanban, for the first time in my long career, I can honestly say that every minute of work my teams do adds value for customers to our products. No time or effort is wasted, and quality is assured.

This book is about how you can duplicate my success with your teams. I've done all the experimenting. I've taken all the missteps. I've culled what's important, and I've laid it out for you in plain language and straightforward steps so that you get just the benefits. Don't let your peers read this book, make use of Kanban, and start making you look antiquated. Take the easy steps I describe and start producing better software that customers love—with high quality, on time, and at low cost.

Who should read this book

This book is for practicing or aspiring software development professionals. You might have started creating software in the 1960s or are just now graduating from college. You could be part of an established software company or an IT organization within a larger company, or you could be a do-it-yourself app or web developer. You might be a software analyst, project manager, program manager, developer, tester, project lead, or development manager. So long as you are a serious practitioner of software development, you will find this book enlightening and invaluable.

This book provides pragmatic and prescriptive step-by-step instructions on how to produce the most value for your customers, with the highest quality at the lowest cost in the least amount of time. I've included diagrams, tables, charts, worksheets, rude Q & A sections, and troubleshooting sections to clarify concepts and guide you toward success. This book also has chapters especially for people who want to adapt from traditional Waterfall methods or evolve from Scrum.

This book might not be for you if . . .

Although the last chapter, "Further resources and beyond," covers the basic theory behind Kanban and other techniques, this book might not be for you if you're looking for a deep reference. Students, academics, and consultants might prefer a different text for in-depth analysis of theory and practice. I suggest several such texts in the last chapter.

Organization of this book

The book follows the progression that a feature team (3–10 people) might experience when learning Kanban:

- Chapter 1, "Getting management consent," covers approaches and steps for gaining consent from management to use Kanban (a necessary condition before you start). This chapter includes an open letter to your manager with a sample proposal.
- Chapter 2, "Kanban quick-start guide," can get you going with Kanban within a few days, provided you have an existing backlog of work. The chapter also includes a troubleshooting section.
- Chapter 3, "Hitting deadlines," helps team members fill and order their backlog as well as estimate how long their project will take and how many resources they'll need.
- Chapter 4, "Adapting from Waterfall," and Chapter 5, "Evolving from Scrum," are for teams that currently use traditional Waterfall or Scrum. These chapters summarize the argument for using Kanban, provide the steps to adapt or evolve to Kanban, and answer the questions a team might have. These chapters and their rude Q & A sections are based on my direct experience with introducing traditional Waterfall and Scrum teams to Kanban.
- Chapter 6, "Deploying components, apps, and services," focuses on delivering the value you produce with Kanban to customers—everything from continuous integration to continuous deployment.
- Chapter 7, "Using Kanban within large organizations," is for teams that use Kanban within large projects of hundreds or thousands of engineers, including how to fit in and report up.
- Chapter 8, "Sustained engineering," is a special contribution from James Waletzky about how to apply Kanban to perform postrelease software maintenance.
- Chapter 9, "Further resources and beyond," provides an overview of the theoretical underpinnings of Kanban and covers how you can improve beyond

the practices described in the previous chapters. This chapter provides resources for those who want to continue learning and evolving.

Acknowledgments

I'll start by congratulating Microsoft Press on its thirtieth anniversary. It was at the anniversary party in Redmond that Devon Musgrave approached me about writing this book. Many thanks to Devon for his suggestion and for championing the book's publication. I'm also deeply indebted to my editor, John Pierce, who did wonders to the readability and consistency of my words.

This book had six reviewers: David Anderson, Corey Ladas, Richard Hundhausen, James Waletzky, Doug Thompson, and Vijay Garg. Doug and Vijay currently work on two of my teams and use Kanban every day. Their feedback was essential to the clarity and accuracy of the text and its examples. James Waletzky is a passionate practitioner of Agile. We've worked together in the past, and his candor and critique have guided my writing for years. James, in his awesomeness, also supplied the chapter on sustained engineering (Chapter 8). Rich joined this project late but provided tremendous suggestions and a tight connection back to the Agile Project Management series. In all, I believe you can't produce worthwhile designs, code, or commentary without thoughtful expert review. To the extent that this book is worthwhile, it is due to my exceptional reviewers.

I want to especially recognize David Anderson and Corey Ladas. David has been an industry leader in project-management techniques throughout his career. He is the originator of the Kanban Method for evolutionary improvement. David speaks and trains professionals around the world. David has always been generous with his time and insights ever since we first collaborated at Microsoft years ago. David's contributions to this book's accuracy, framing, and language are essential and extensive. Even with all his traveling, David found substantial time to carefully review this work, for which I am immensely grateful.

Corey Ladas's influence on my thinking and career cannot be overstated. Corey introduced me to Scrum, Agile, axiomatic design, TRIZ, House of Quality, and numerous other techniques. In 2007, Corey invited me to Corbis to see the work that he and David Anderson were doing there with Kanban. I was instantly enthralled. Although it would take me a few years to try it myself, I immediately shared the work with as many peers as would listen. Corey is a deep thinker, who consistently challenges the status quo. He is fearless and unflinching. Corey can be tough and defiant, but he is always honest and insightful. I am delighted to call him my friend. Corey was the inspiration for this book.

Finally, I'd like to thank my past managers (Curt Steeb, Boyd Multerer, and Kareem Choudhry) for making Xbox such a great place to work, all my team members over the years who embraced experimentation and shared in my success, and, most of all, my wife, Karen, and my sons, Alex and Peter, for making me so very happy.

Downloads: Sample files

I've provided a couple of sample files, which you can download from the following page:

http://aka.ms/pmwithkanban/files

The first file is a sample of a letter and proposal you can provide to your management to gain consent to use Kanban. The second is an Excel workbook with every sample spreadsheet shown in the book, including those to calculate work-in-progress (WIP) limits, estimate completion dates, and even chart productivity and quality over time.

Follow the instructions on the page to download the files.

System requirements

The files provided online are in the Office Open XML format. The basic requirement for using the files is to have an Excel Viewer and Word Viewer installed on your computer.

You can download the Excel viewer from *http://www.microsoft.com/en-us/download/details.aspx?id=10.*

You can download the Word view from *http://www.microsoft.com/en-us/download/details.aspx?id=4.*

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

http://aka.ms/pmwithkanban/errata

If you discover an error that is not already listed, please submit it to us at the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com.*

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com.*

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

We know you're busy, so we've kept it short with just a few questions. Your answers go directly to the editors at Microsoft Press. (No personal information will be requested.) Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*

CHAPTER 1

Getting management consent

This book describes how you can manage software projects with great efficiency, predictability, and simplicity using Kanban. Kanban helps you deliver value to your customers faster than Waterfall, Scrum, or just about any other project-management method. It helps you deliver that value with high quality, on time, and on budget, yet it will still fail miserably without the consent of management.

Before you engage in any kind of change, even one as effective as Kanban, you must gain consent from your management. Why? Because employees tend to do what they are told to do—until, that is, those employees realize that they are actually rewarded for doing something else. The people who tell employees what to do are managers. The people who determine employee rewards are managers. Change only succeeds if managers say, "Yes, you should do this," and later reward employees who do it.

Thus, for you and your team to successfully adopt Kanban, you must first convince your management to support it, or at least to not obstruct it and not penalize team members for using it. Even if you are a manager yourself, you still must ensure that your management chain won't oppose Kanban.

Convincing management to make a change, even a good change, can be difficult. A proven approach is to present a proposal that briefly outlines the problem, the solution, the risks with mitigations, measures of success or failure, and a plan for going forward.

For your convenience, I provide a sample proposal in this chapter, in the form of an open letter to your manager. (You can download an editable electronic copy. See the book's introduction for details.) If your manager approves, you can move forward with confidence. If your manager does not approve, even after you address his or her concerns, you should either stop reading and shelve this book or find a new manager. Good luck!

The topics covered are:

An open letter to your manager
Moving forward
Checklist

An open letter to your manager

Dear Sir or Madam,

Our team would like to use Kanban to manage its project work. This proposal lays out the need for this change, why Kanban was selected as a solution, the risks involved and suggested mitigations, and a plan to roll out the change. We look forward to your feedback to this proposal and to enacting the plan once we've addressed any concerns you may have.

Problem

We currently spend significant time doing work that's unrelated to delivering value to our customers.

- We attend an excessive number of meetings about planning and process.
- Problems fester for weeks or months before they are noticed, analyzed, and corrected.
- Careless team members are rewarded for pure speed, encouraging them to create costly bugs and submit incomplete work.
- Quality goes unchecked for weeks or months, which builds up an extensive amount of rework.
- Schedules slip as requirements change and work is reprioritized, which forces more meetings about planning and process and wastes the effort spent on abandoned work.

In the end, our products are delivered late with less functionality and lower quality at higher cost. We used to just accept this outcome because we've always worked this way. However, we now feel we've found a simple and effective solution.

Solution

Kanban is a simple project-management technique that's based on Toyota's just-in-time scheduling mechanism. Using Kanban to manage our project work will allow us to focus all our time and energy on delivering value to our customers.

- Kanban has planning meetings only on demand and no special meetings about process.
- Kanban visualizes project workflow, spotlights bottlenecks the day they occur, and forces team members to immediately resolve the issue or swarm to fix it.
- Kanban prevents careless team members from prematurely designating work as complete.
- Kanban enforces clear quality bars at each step, driving quality upstream.
- Kanban minimizes work in progress, freeing teams to adjust daily to new priorities and requirements with little sunk cost and allowing a team to deliver on time.

Kanban isn't a magic bullet. It won't fix every problem. What it can do is simplify our project management; reduce time lost to meetings, bottlenecks, and rework; better govern our product quality; and make our throughput of customer value smoother, faster, and more predictable.

As with any work change, our team will need a few weeks to adjust to Kanban and a few months to master it. However, we'll all enjoy the benefits—delivering on time with greater functionality and higher quality at lower cost.

Risks

Any change has risks. We list the ones for Kanban along with our planned mitigations.

Risk	Mitigation
Current work could be disrupted as the team acclimates to the new approach.	Adopt Kanban at the start of a new project or milestone.
Some team members could object to the change.	Ensure that team leaders and influencers agree to try Kanban for two months.
A lack of experience with the new approach could cause it to fail.	Invest in a Kanban coach, training, or book for all team members.
The team's productivity could drop during the first few weeks of adjustment.	Reduce expectations of output for the first month.
Status and tracking tools may need to be updated.	Have the team's project manager enter status information daily into the existing tracking tools.
Issues with dependencies and requirements changes could disrupt adoption.	Kanban is inherently good at dealing with scheduling disruptions, so dependency issues and requirements changes will serve well as practice.

Although an initial drop in productivity is expected during the first few weeks (which occurs with any change), the subsequent increases in productivity should quickly recover the lost output.

Plan

Here is an outline of our plan of action, divided into four phases:

Phase	Activities	Duration
1	• Brief team leaders and influencers on Kanban (in progress) • Gain agreement on a two-month trial period • Communicate adoption plan to team one month in advance • Provide team with access to Kanban online resources and books • Utilize experienced coach to answer early questions • Create baseline of productivity and bug metrics	1 month
2	• Train team on Kanban with consultant or team expert • Collect and organize backlog of work • Document current steps used to produce value for customers • Construct progress signboard on wall near team's location • Determine work-in-progress (WIP) limits • Establish completion criteria for each step	2 weeks

Phase	Activities	Duration
3	• Run daily standup meetings • Enter status information daily into existing tracking tools • Track productivity and bug metrics • Utilize consultant or team expert to answer team questions • Adjust WIP limits and completion criteria as needed • Reduce expectations of output during this period	1 month
4	• Run daily standup meetings • Enter status information daily into the existing tracking tools • Track productivity and bug metrics • Celebrate performance improvements	Ongoing

Each phase is contingent upon a successful prior phase. Success is measured by completion of each phase's activities and the continued engagement of the team. Overall success is measured by improvement of productivity and bug metrics over the baseline set in the first phase.

We look forward to your feedback to this proposal and to enacting the plan after we've addressed any concerns you may have.

Sincerely,

A team passionate about delivering the greatest value to its customers

Moving forward

Some managers will support the initiative of your team and approve your proposal with few reservations. For them, tracking productivity and bug metrics as your team improves will provide all the positive feedback they need to stand by their decision.

Some managers will be more skeptical and ask for a deeper review. These managers often need to experience Kanban for themselves to remove the mystery, understand the source of productivity and quality gains, and gain confidence in the approach that only personal experience provides. Fortunately, Kanban simulations are available that deliver hands-on practice with the approach and demonstrate how and why it works. It's worth bringing in an experienced Kanban coach to run the simulation and answer your manager's questions. Coaches often have programs that specifically target management concerns.

After your manager's concerns are addressed, you can move forward with the plan described in the letter and detailed in the chapters that follow. For Waterfall and Scrum teams, I have included specific chapters that illustrate how to smooth the adoption of Kanban.

Inside Xbox

When my teams switched to Kanban, I didn't need management permission. As an Xbox development manager, I have the discretion to specify the methods my teams use. However, my teams did need to report progress on features and bugs using the Xbox-wide tracking system.

(We first used a Microsoft internal system, Product Studio, then switched to Visual Studio Team Foundation Server (TFS), and now use Visual Studio Online.) In Chapter 7, "Using Kanban within large organizations," I talk about how to fit Kanban into a big project seamlessly.

I first used Kanban with some of my Scrum teams that worked on web services. I had described Kanban to all my teams and asked whether any would be interested in trying it. Two teams were, and they quickly adapted their work cadence for continuous delivery. Both teams liked Kanban and said they wouldn't go back to Scrum. Both also quickly landed on many of the specific recommendations you'll read about in the coming chapters.

Eight months after I introduced my teams to Kanban, a reorg moved me to a new Xbox group. I had to build much of the new group from scratch, hiring engineers from around Microsoft who were mostly experienced with Waterfall. Since we had no established practices, I simply stated that my teams used Kanban. The Waterfall engineers took to it quickly and were free to adjust Kanban to their needs, so long as they followed the key principles I outline in this book. I capture many of their adjustments in the "Troubleshooting" section in Chapter 2, "Kanban quick-start guide."

Checklist

Here's a checklist of actions to take when gaining management consent for using Kanban:

- ❑ Ask management for consent to use Kanban.
- ❑ Update the proposal language, risks, and plan as needed.
- ❑ Send the proposal to your management.
- ❑ Provide your management with a review of Kanban and with hands-on experience with Kanban as needed.
- ❑ Address any concerns your management raises.
- ❑ Create a baseline of productivity and bug metrics.
- ❑ Execute your plan of action, tracking your improvements in productivity and bug metrics.

CHAPTER 2

Kanban quick-start guide

Kanban provides a simple approach to delivering high-quality value to your customers, on time and on budget. If you already have a set of work items to do and a team ready to do them (as most teams do), use this chapter to get started right away. For those who don't yet have a set of work items or who need to put a team in place, Chapter 3, "Hitting deadlines," describes how to plan your project and staff your team.

The quick-start steps for using Kanban to complete your work backlog are:

Step 1: Capture your team's high-level routine
Step 2: Redecorate your wall
Step 3: Set limits on chaos
Step 4: Define done
Step 5: Run your daily standup
Troubleshooting
Checklist

Step 1: Capture your team's high-level routine

Team members do many kinds of work:

- Discuss the product with partners, teammates, and customers.
- Write and answer email, and engage in relevant social media.
- Find, evaluate, and fix bugs and operational issues (tickets).
- Track feedback.
- Produce improvements to products and infrastructure.

While all this work is important, Kanban focuses primarily on the last and most intrinsic item: producing improvements to products and infrastructure.

There are exceptions that Kanban can also manage, such as the following:

- Preparing for a major presentation.
- Writing proposals or major design documents, or rolling out major changes.