

A radical approach to getting IT projects done **faster** & **cheaper** than anyone thinks possible

SOFT- WARE IN 30 DAYS



How Agile Managers
Beat the Odds, Delight
Their Customers,
and Leave
Competitors
in the Dust

KEN SCHWABER and **JEFF SUTHERLAND**
Creators of **SCRUM**

**SOFT-
WARE
IN
30
DAYS**

SOFT- WARE IN 30 DAYS

How Agile Managers
Beat the Odds, Delight
Their Customers,
and Leave
Competitors
In the Dust

KEN SCHWABER and JEFF SUTHERLAND



John Wiley & Sons, Inc.

Copyright © 2012 by Ken Schwaber and Jeff Sutherland. All rights reserved.

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

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

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 publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Schwaber, Ken.

Software in 30 days: how Agile managers beat the odds, delight their customers, and leave competitors in the dust/

Ken Schwaber, Jeff Sutherland.

p. cm.

Includes index.

ISBN 978-1-118-20666-9 (pbk.); ISBN 978-1-118-22854-8 (ebk); ISBN 978-1-118-24090-8 (ebk);

ISBN 978-1-118-26574-1 (ebk)

1. Agile software development. 2. Scrum (Computer software development) 3. Computer software—Development. I. Sutherland, Jeffrey Victor. II. Title. III. Title: Software in thirty days.

QA76.76.D47S3223 2012

005.1—dc23

2011050969

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

*To Ikujiro Nonaka, Babatunde A. Ogunnaike, and
Hirotaka Takeuchi for their inspiration and guidance.*

Contents

<i>About the Authors</i>	<i>xi</i>
<i>Acknowledgments</i>	<i>xiii</i>
<i>Introduction</i>	<i>xv</i>

Section I Why Every Business in the World Can Produce Software in 30 Days	1
--	----------

You are probably frustrated with your software organization. You would like it to be quicker, more flexible, understand your needs better, and help you become more profitable. We look at why you are frustrated and how to fix the problem.

1 The Crisis in Software: The Wrong Process Produces the Wrong Results	3
---	----------

Many software organizations follow a development process that guarantees waste, uncontrolled risk, unpredictability, surprises, and low value. We will investigate why this process was chosen, how it guarantees failure, and look at some organizations that have recovered from it.

2 Scrum: The Right Process Produces the Right Results	17
--	-----------

There is a process that is appropriate for software development. When you get your developers to use it, you will immediately gain productivity, quality, value, control, predictability, and satisfaction. We look at how this happens in this chapter.

3 Try It Yourself: The Pilot 33

You have read our assertion that there is a better way for you to get software developed for you. However, a lot of people have made assertions and taken a lot of your money in the past, with little or no improvement. In this chapter we show you how to prove that our approach works for no money.

4 What Can I Do? 49

You learned how to do better and you've tried it yourself. You like the results and you know what to tell the software organization to do. In this chapter, we look at what you can do to help what you experience in the pilot project succeed.

Section II How to Produce Software in 30 Days 55

Having better software developed for your needs is not so much hard as it is different from what you are used to. In this section, we look at a progressively beneficial set of approaches to get you from where you are now to organizational agility.

5 Getting Started with Scrum 57

Our secret sauce for improving your benefits from software is called "Scrum." Yes, this is the rugby event that keeps the ball moving down the field. We'll discuss Scrum, how it works, and why it works in this chapter.

6 Scrum at the Project Level 63

Most persistent improvement in software development starts at the project level. You can use Scrum to further prove its utility, or on critically important initiative that must succeed. We'll explore what you can tell your developers to do after reading this chapter.

7 Develop a Scrum Capability 75

Success often breeds success. As more software initiatives using Scrum succeed, more people will want to get on the wagon. Rather than changing the entire organization, let's look at how we can set up a software development universe separate from the disappointing, existing department. You can increasingly reap benefits here on an increasing number of projects and releases.

8	Scrum at the Enterprise Level	101
	<i>Scrum at a project or release level provides initiative level agility, the ability to rapidly respond to opportunities or rise to challenges. To gain the most significant benefits, Scrum's empirical approach to software development must be fit into the organization as a whole. We'll look at how to do this, and why some approaches are short-lived and others persist.</i>	
9	Enterprise Transformation: Profound and Persistent Change	107
	<i>You want to make your organization leaner, more efficient, and agile on your watch. Even more, you want these benefits and their underlying causes to persist and become the organizational culture. We'll look at an enterprise change approach for achieving this in this chapter.</i>	
10	Scrumming Scrum	119
	<i>We devised Scrum for complex problem solving, like software development. We found Scrum a useful technique for managing organizational change, also a complex problem. The same benefits of transparency, waste removal, risk control, and predictability occurred. We'll look at this use of Scrum in this chapter.</i>	
	Appendix 1: Terminology	127
	<i>We slowly and progressively introduced some new terminology. This appendix is your reference for those terms.</i>	
	Appendix 2: The Scrum Guide	133
	<i>Read the canonical guide to Scrum, its roles, artifacts, and events. This is the bible of Scrum.</i>	
	Appendix 3: A Playbook for Achieving Enterprise Agility	153
	<i>This appendix presents a more detailed plan for enterprise change, as discussed in Chapter 10.</i>	
	Index	185

About the Authors

Jeff Sutherland and Ken Schwaber are the creators of Scrum, a software development process that delivers software functionality in 30-day increments. Scrum was born when Jeff and Ken presented a paper at the OOPSLA conference in Austin, Texas, in August 1995. This paper, “Scrum Development Process,” was the result of their collaboration prior to that point. The works of H. Takeuchio and I. Nonaka in their seminal works on lean knowledge creation, bottom-up intelligence, and teamwork had profoundly influenced Jeff. Babatunde Ogunnnike had profoundly influenced Ken in his work on industrial process control and the applicability of complexity theory and empiricism to software development.

In addition to being Scrum’s creators, Jeff and Ken have also served as its wards. With their guidance, Scrum has evolved over time; more recently, they have developed ways to speed up Scrum’s systematic evolution based on community experience and input. In “The Scrum Guide,” found in Appendix 2 of this book, Jeff and Ken offer the complete definition of Scrum.

Dr. Jeff Sutherland is the chief executive officer of Scrum Inc., in Cambridge, Massachusetts, offering training, guidance, and coaching to companies across the globe. Jeff is a distinguished graduate of the United States Military Academy and a Top Gun of his USAF RF-4C Aircraft Commander class. Jeff has advanced degrees from Stanford University and a PhD from the University of

Colorado School of Medicine. He is also a senior advisor to OpenView Venture Partners, helping them implement Scrum and agile practices in all their portfolio companies. Jeff has extended and enhanced Scrum at many software companies and information technology (IT) organizations over the years.

Ken Schwaber is a software development professional, having spent the past 40 years of his life as a programmer, analyst, consultant, product manager, and business owner. Early in his career, Ken tried unsuccessfully to make waterfall software projects successful; he later developed an alternative to waterfall. Ken has spent the past 20 years developing Scrum and working with organizations around world to help them take advantage of it. Ken is one of the original signatories of the Agile Manifesto and the founder of the Agile Alliance and the Scrum Alliance. He is currently working to improve the software profession through [Scrum.org](https://www.scrum.org). Ken and his wife, Christina, live in the Boston area. He is a graduate of the United States Merchant Marine Academy and has completed additional study in computer science at the University of Chicago and in business at the University of California at Los Angeles Anderson School of Management.

Acknowledgments

THIS BOOK WOULD not be what it is without the excellent copyediting of Arlette Ballew, the overall direction of Richard Narramore, and the laser focus of Carey Armstrong.

Introduction

WE, JEFF AND Ken, have been in the software industry, collectively, for 70 years. We have been software developers, managers in IT organizations and software product companies, and owners of both product companies and service organizations. More than 20 years ago, we created a process that lets organizations deliver software better. Since then, we have helped hundreds of organizations do the same. Our work has spread further than we have ever imagined possible, being put to use by millions of people. We are humbled by the extent of its adoption, and we are awed by the feats people have accomplished using it.

This is not the first book we have written on the topic of building software. It is, however, the first book we have written for people who do not themselves build software. This book is instead for leaders within organizations that depend on software for their survival and competitiveness. It is for leaders within organizations that can benefit from developing software rapidly, incrementally, and with the best return on investment possible. It is for leaders who face business and technological complexity that has made the delivery of software difficult. We have written this book so that these leaders can help their organizations achieve these goals, enhance their internal capabilities, improve their product offerings, and more.

This book is for chief executive officers (CEOs), executives, and senior managers who need their organizations to deliver better software in less time,

with lower cost, greater predictability, and lower risk. For this audience, we have a message: You may have had negative experiences with software development in the past, but the industry has turned a corner. The software profession has radically improved its methods and its results. The uncertainty, risk, and waste to which you are accustomed are no longer par for the course. We have worked with many software organizations that have already turned the corner; we want to help you do so, too.

In this book, we show you how to create business value using a process that delivers complete pieces of software functionality at least every 30 days. This book will show you how you can prioritize the functionality you want and have it delivered *à la carte*. It will show you how to gain transparency not only into business value, by tracking functionality delivered against functionality desired, but also into the health of the software development process and your organization as a whole. The tools in this book will help you work with your software organization to get up to speed with modern practices and begin to deliver the results you've been expecting all along.

This is software in 30 days.

**SOFT-
WARE
IN
30
DAYS**

SECTION



Why Every Business in the World Can Produce Software in 30 Days

WE REACH OUT to every leader in an organization who wants to build better software products with better value and predictability. The software industry is turning around and radically improving. The uncertainty, risk, and waste you are used to are no longer necessary. We have 20 years of data under our belts from working with the many organizations that have already turned the corner. We want you to do so also. We want you to be able to build valuable, quality software predictably with manageable risk.

We reach out to you for two reasons. First, you have been ill served by the software industry for 40 years—not purposefully, but inextricably. We want to restore the partnership. Second, software is no longer only in the back room. Software is everywhere, in more and more critical operations of our society. We want you to be able to build software that we can all reliably depend on.

We hope we achieve our goals for this book. Regardless, do not give up. You no longer need to accept the terrible software results of the past. Move on.

In this part of the book, we investigate why software development has been so bad. We move on to show how software has improved and the two

underlying epiphanies that have facilitated this. We then show you how you can pilot our approach, and what you can do to help it succeed. Section II provides you with increasingly rigorous steps to take advantage of our new approach, should the pilot convince you to do so.