

# Outside-in Software Development

A Practical Approach to Building  
Successful Stakeholder-based Products

Foreword by Tom Poppendieck

Carl Kessler and John Sweitzer



# Register Your Book

at [www.ibmpressbooks.com/ibmregister](http://www.ibmpressbooks.com/ibmregister)

**Upon registration, we will send you electronic sample chapters from two of our popular IBM Press books. In addition, you will be automatically entered into a monthly drawing for a free IBM Press book.**

Registration also entitles you to:

- Notices and reminders about author appearances, conferences, and online chats with special guests
- Access to supplemental material that may be available
- Advance notice of forthcoming editions
- Related book recommendations
- Information about special contests and promotions throughout the year
- Chapter excerpts and supplements of forthcoming books

## Contact us

If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

Editorial Director, IBM Press  
c/o Pearson Education  
One Lake Street  
Upper Saddle River, New Jersey 07458

e-mail: [IBMPress@pearsoned.com](mailto:IBMPress@pearsoned.com)

Visit us on the Web: [www.ibmpressbooks.com](http://www.ibmpressbooks.com)

*This page intentionally left blank*

# **Outside-in Software Development**

## RATIONAL AND SOFTWARE DEVELOPMENT

### IBM Rational® ClearCase®, Ant, and CruiseControl

Lee ■ ISBN 0321356993

### Implementing IBM® Rational® ClearQuest®

Buckley, Pulsipher, and Scott ■ ISBN 0321334868

### Implementing the IBM® Rational Unified Process® and Solutions: A Guide to Improving Your Software Development Capability and Maturity

Barnes ■ ISBN 0321369459

### Outside-in Software Development

Kessler and Sweitzer ■ 0131575511

### Project Management with the IBM® Rational Unified Process®

Gibbs ■ ISBN 0321336399

### Software Configuration Management Strategies and IBM® Rational® ClearCase®, Second Edition

Bellagio and Milligan ■ ISBN 0321200195

### Visual Modeling with IBM® Rational® Software Architect and UML™

Quatrani and Palistrant ■ ISBN 0321238087

## COMPUTING

### Autonomic Computing

Murch ■ ISBN 013144025X

### Business Intelligence for the Enterprise

Biere ■ ISBN 0131413031

### Grid Computing

Joseph and Fellenstein ■ ISBN 0131456601

### Inescapable Data: Harnessing the Power of Convergence

Stakutis and Webster ■ ISBN 0131852159

### On Demand Computing

Fellenstein ■ ISBN 0131440241

### RFID Sourcebook

Lahiri ■ ISBN 0131851373

### Service-Oriented Architecture (SOA) Compass

Bieberstein, Bose, Fiammante, Jones, and Shah ■ ISBN 0131870025

## INFORMATION MANAGEMENT

### An Introduction to IMS™

Meltz, Long, Harrington, Hain, and Nicholls ■ ISBN 0131856715

### DB2® Express

Yip, Cheung, Gartner, Liu, and O'Connell ■ ISBN 0131463977

### DB2® for z/OS® Version 8 DBA Certification Guide

Lawson ■ ISBN 0131491202

### DB2® SQL PL, Second Edition

Janmohamed, Liu, Bradstock, Chong, Gao, McArthur, and Yip  
ISBN 0131477005

### High Availability Guide for DB2®

Eaton and Cialini ■ 0131448307

### The Official Introduction to DB2® for z/OS®, Second Edition

Sloan ■ ISBN 0131477501

### Understanding DB2® 9 Security

Bond, See, Wong, and Chan ■ ISBN 0131345907

### Understanding DB2®

Chong, Liu, Qi, and Snow ■ ISBN 0131859161

## WEBSPHERE

### Enterprise Java™ Programming with IBM® WebSphere®, Second Edition

Brown, Craig, Hester, Pitt, Stinehour, Weitzel, Amsden, Jakab, and Berg  
ISBN 032118579X

### Enterprise Messaging Using JMS and IBM® WebSphere®

Yusuf ■ ISBN 0131468634

### IBM® WebSphere®

Barcia, Hines, Alcott, and Botzum ■ ISBN 0131468626

### IBM® WebSphere® Application Server for Distributed Platforms and z/OS®

Black, Everett, Draeger, Miller, Iyer, McGuinness, Patel, Herescu, Gissel, Betancourt, Casile, Tang, and Beaubien ■ ISBN 0131855875

### IBM® WebSphere® System Administration

Williamson, Chan, Cundiff, Lauzon, and Mitchell ■ ISBN 0131446045

## LOTUS

### IBM® WebSphere® and Lotus®

Lamb, Laskey, and Indurkha ■ ISBN 0131443305

### Lotus® Notes® Developer's Toolbox

Elliott ■ ISBN 0132214482

## OPEN SOURCE

### Apache Derby—Off to the Races

Zikopoulos, Baklarz, and Scott ■ ISBN 0131855255

### Building Applications with the Linux® Standard Base

Linux Standard Base Team ■ ISBN 0131456954

### Performance Tuning for Linux® Servers

Johnson, Huizenga, and Pulavarty ■ ISBN 013144753X

## BUSINESS STRATEGY & MANAGEMENT

### Can Two Rights Make a Wrong?

Moulton Reger ■ ISBN 0131732943

### Developing Quality Technical Information, Second Edition

Hargis, Carey, Hernandez, Hughes, Longo, Rouiller, and Wilde  
ISBN 0131477498

### Do It Wrong Quickly

Moran ■ 0132255960

### Irresistible! Markets, Models, and Meta-Value in Consumer Electronics

Bailey and Wenzek ■ ISBN 0131987585

### Mining the Talk: Unlocking the Business Value in Unstructured Information

Spangler and Kreulen ■ ISBN 0132339536

### Reaching the Goal

Ricketts ■ 0132333120

### Search Engine Marketing, Inc.

Moran and Hunt ■ ISBN 0131852922

### The New Language of Business: SOA and Web 2.0

Carter ■ ISBN 013195654X

# Outside-in Software Development

A Practical Approach to Building Successful  
Stakeholder-based Products

**Carl Kessler and John Sweitzer**

**IBM Press  
Pearson plc**

**Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Cape Town • Sydney • Tokyo • Singapore • Mexico City**

**[ibmpressbooks.com](http://ibmpressbooks.com)**

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2008 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Tara Woodman, Ellice Uffer

Cover design: IBM Corporation

Associate Publisher: Greg Wiegand

Marketing Manager: Kourtayne Sturgeon

Publicist: Heather Fox

Acquisitions Editor: Bernard Goodwin

Managing Editor: John Fuller

Project Editor: Lara Wysong

Copy Editor: Audrey Doyle

Indexer: Ted Laux

Compositor: Molly Sharp

Proofreader: Linda Begley

Manufacturing Buyer: Anna Popick

Published by Pearson plc

Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com

For sales outside the U.S., please contact:

International Sales

international@pearsoned.com

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: DB2, Lotus, Tivoli, WebSphere, Rational, IBM, the IBM logo, and IBM Press. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and the Windows logo are trademarks of the Microsoft Corporation in the United States, other countries, or both. Linux is a registered trademark of Linus Torvalds. Intel, Intel Inside (logo), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both. OSF/1 and UNIX are registered trademarks and The Open Group is a trademark of The Open Group in the United States and other countries. Other company, product, or service names mentioned herein may be trademarks or service marks of their respective owners.



This Book Is Safari Enabled

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days. Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- Go to <http://www.awprofessional.com/safarienabled>
- Complete the brief registration form
- Enter the coupon code 5JK2-99EJ-QNXV-TIJB-F4LT

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail [customer-service@safaribooksonline.com](mailto:customer-service@safaribooksonline.com).

---

*Library of Congress Cataloging-in-Publication Data*

Kessler, Carl.

Outside-in software development : a practical approach to building successful stakeholder-based products / Carl Kessler and John Sweitzer.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-13-157551-6 (pbk. : alk. paper)

1. Computer software—Development. I. Sweitzer, John. II. Title.

QA76.76.D47K485 2007

005.1—dc22

2007027587

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.  
Rights and Contracts Department  
501 Boylston St., Suite 900  
Boston, MA 02116  
Fax (617) 671-3447

ISBN-13: 978-0-13-157551-6

ISBN-10: 0-13-157551-1

Text printed in the United States on recycled paper at Courier in Stoughton, Massachusetts.  
First printing, September 2007

*As you'd expect from a book that advocates thinking from the outside-in, this book is dedicated to the clients and business partners who seek and deserve to get high-quality software products that help solve real business problems and enable their businesses to grow. Because one of the stakeholder groups we describe in this book is the set of insiders, we are similarly comfortable dedicating this work to the IBMers we've had the good fortune to work with over many years, who deliver top-notch software on a regular basis and who work every day to deliver innovation that matters using outside-in thinking.*

*The most important dedication, of course, is to our families, who have put up with our work habits and the added hours of writing, with tireless patience and good humor. We are fortunate beyond words. We now look forward to having more time to spend with them.*

# Contents

Foreword	xv
Preface	xvii
Acknowledgments	xxiii
About the Authors	xxv

<b>Chapter 1</b>	■	<b>Introducing Outside-in Development</b>	<b>1</b>
		There may be challenges to overcome	2
		Consider a different way of thinking about software product development	2
		Consider some proven techniques as well	3
		It takes a whole team to succeed	3
		<i>A whole-team perspective is essential</i>	3
		Understand your stakeholders	4
		Understand organizational context	5
		Make your products consumable	7
		Align with your stakeholders' goals	7
		Define success in your stakeholders' terms	8
		Become an outside-in developer	9
		The leader's role in outside-in development	9
		Essential point: You can get started now	10

---

<b>Chapter 2</b>	■	<b>Understanding Your Stakeholders</b>	<b>11</b>
		Introduction to stakeholder understanding	12
		Identify the stakeholders	16
		Understand the stakeholders' goals	18
		<i>Tips for describing stakeholders' goals</i>	21
		<i>Identify constraints as they come up in discussion</i>	23
		The four stakeholder groups	23
		<i>Principals</i>	24
		<i>End-users</i>	25
		<i>Partners</i>	28
		<i>Insiders</i>	29
		<i>Consider the relationships between stakeholders</i>	30
		Engage in dialog with stakeholders	31
		<i>Understand principal stakeholders' goals</i>	31
		<i>Understand end-user stakeholders' goals</i>	38
		<i>Understand partner stakeholders' goals</i>	39
		<i>Understand insider stakeholders' goals</i>	39
		Align client discussions with stakeholder goals	41
		The leader's role in understanding your stakeholders	43
		Essential points	44
		Key terms	45
<b>Chapter 3</b>	■	<b>Understanding Organizational Context</b>	<b>47</b>
		Introduction to organizational context	49
		<i>Conflicting feedback due to different organizational contexts</i>	51
		<i>Organizational context variances in practice</i>	52
		<i>Use organizational design awareness for software market selection</i>	56
		<i>Additional dimensions of organizational context variance</i>	57
		Handling diverse client requirements	59
		<i>You might not please everyone</i>	59
		<i>One customer does not make a market</i>	59

---

	<b>Use organizational context to speak your stakeholders' language</b>	<b>60</b>
	<i>Use organization variance patterns to get and use stakeholder feedback</i>	64
	<b>The leader's role in using organizational context</b>	<b>68</b>
	<b>Essential points</b>	<b>68</b>
	<b>Key terms</b>	<b>68</b>
<b>Chapter 4</b>	<b>■ Making Products Consumable</b>	<b>71</b>
	<b>Introduction to product consumability</b>	<b>73</b>
	<i>Why is consumability important?</i>	75
	<b>Identify consumability meta-tasks</b>	<b>78</b>
	<b>Use a consumability score card to guide investments</b>	<b>82</b>
	<i>Use the consumability meta-tasks in the score card</i>	83
	<i>Build the score card</i>	90
	<b>Choose which consumability meta-tasks to emphasize</b>	<b>93</b>
	<i>Validate your consumability data</i>	95
	<i>Include the real world in the score card</i>	96
	<i>Include your integrated product portfolio in the score card</i>	96
	<i>Go beyond the score card</i>	98
	<b>Allow more people to use your product</b>	<b>99</b>
	<i>Globalization</i>	99
	<i>Accessibility</i>	100
	<b>The leader's role in making products consumable</b>	<b>101</b>
	<b>Essential points</b>	<b>102</b>
	<b>Key terms</b>	<b>103</b>
<b>Chapter 5</b>	<b>■ Aligning with Stakeholder Goals</b>	<b>105</b>
	<b>Introduction to stakeholder alignment</b>	<b>107</b>
	<b>Focus dialog on what matters to stakeholders</b>	<b>108</b>
	<i>All of your stakeholders matter</i>	109
	<i>Feedback quality depends on what you show and to whom</i>	115
	<i>Getting effective feedback about your product's infrastructure is difficult</i>	118

**Plan to have the capacity to stay aligned with your stakeholders 120***Overcome the challenge of building the capacity to respond 121**Build a capacity model that maps to your development methodology 122**Remember to share what you did 123**Questions to ask when planning capacity 126**Be careful about trimming capability 126***Minimize noise in your deliverables to improve stakeholder feedback 127***Invest in your team 128**Why bring defects into a discussion about staying aligned with stakeholder goals? 129***Help ensure quality 129***Collaborate 130**Automate 131**Be careful about unanswered defects 132**Learn from your defects 134**Bring the stakeholders' world into your work 135**Bring the stakeholders themselves into your world 136**Let everyone on your team experience reality 137***Rehearse for the success of your product 137***Plan for a dress rehearsal of your product 139**Open the doors to your critics 139**Dress rehearsals apply in any development methodology 140**Invite partner stakeholders to the rehearsal 140**Include as many insiders as possible in the rehearsal 141***The leader's role in aligning with stakeholder goals 143****Essential points 144****Key terms 144****Chapter 6 ■ Defining Success in Your Stakeholders' Terms 147****Introduction to stakeholder success in production 148****Wave one: It is all about principal stakeholder success 150***Set a success goal for your product 151**Discover whether your goal is being met 152*

*Use reverse residencies with the first wave of clients* 153  
*Keep the right mindset* 154  
*Plan to have the capacity to support the first wave of success* 156

**Use what you learn from the first wave** 157

*Learn how your product performs in the wild* 157  
*Pay attention to the defects that affect first-wave clients* 158  
*Be impatient to get more capabilities out to your stakeholders* 160  
*Use the right ship vehicle to deliver follow-on capabilities* 161

**Wave two: It is all about the long-term commitment to your clients** 164

*Provide clarity about your support model* 165  
*Plan to adjust your product dynamically* 165  
*Get feedback from your second-wave stakeholders* 166  
*Use the feedback you get to affect your next release* 166

**Wave three: Help your stakeholders deal with the old and the new** 168

*Consider how existing clients will migrate to the new release* 168  
*Provide migration assistance* 168  
*Support your stakeholders* 169

**The leader's role in defining success** 169

**Essential points** 170

**Key terms** 170

**Chapter 7 ■ Becoming an Outside-in Developer** 173

**How to adopt outside-in development techniques** 174

*Use outside-in techniques to establish a stakeholder-based culture* 175  
*Culture change comes from both ends of the organization chart* 176  
*Deploy outside-in development in-place* 177  
*What should you do to get started?* 178  
*Multiple leadership opportunities exist* 179  
*Are there circumstances in which a more aggressive move makes sense?* 180

<b>Insights into effective use of outside-in development</b>	<b>181</b>
<i>Plan to incrementally adopt outside-in development techniques for your team</i>	181
<i>Target the right initial principal stakeholders</i>	182
<i>How to use outside-in development for multiple software projects</i>	184
<i>How outside-in development fits with waterfall projects</i>	189
<i>How outside-in development fits with Lean Six Sigma projects</i>	192
<i>How outside-in development fits with lean agile projects</i>	194
<b>The leader's role in adopting outside-in development</b>	<b>196</b>
<b>Essential points</b>	<b>196</b>
<b>Key terms</b>	<b>197</b>
■ <b>Index</b>	<b>199</b>

# Foreword

**S**ilver bullets, we are told, are very effective for managing werewolves. Decades of experience, however, have shown them to be dramatically less effective for managing software development. Magical creatures invented in previous centuries still haunt us today despite our best efforts to manage them away. Our bright hopes for prosperity and satisfying work that we can take pride in remain even more rare than werewolves in contemporary software development organizations.

Albert Einstein claimed that, “The world we have made as a result of the level of thinking we have done thus far creates problems we cannot solve at the same level of thinking at which we created them.” Thinking more carefully about software development as a magical process that turns requirements into valuable software may actually be the problem rather than the solution we seek. Even beautifully crafted, flawless software actually has very little practical value in and of itself. The design of the product or the business process the software is embedded in determines its potential value. The actual value comes when the product is used to satisfy some need. The job is not done until the need is met.

When you examine the value chain from initial concepts of unmet needs or goals to the achievement of those goals through ongoing use of a new product or business process, it immediately becomes clear that software is just one part. Thinking at the value chain level strives to avoid the sub-optimization that follows from separating design and implementation of the software from design of the product or process (requirements) and from integrating, deploying, supporting, and actually using the software-enabled product. Decisions made in each area significantly affect work in each of the other areas. Your teams will deliver the greatest value when tradeoffs are made to optimize the value chain rather than any one area at the expense of the others.

Organizations that depend on software maintain an entire menagerie of mythical creatures, governance processes, requirements processes, software development processes, sales and marketing processes, operations processes, and more. Each is strong and beautiful and constantly fighting with the others and delivering more waste than prosperity for the organization or value for the customer. Silver bullets aimed at optimizing any one usually make life miserable for the others.

Each part of the value chain, from concept to sustainable cash flow is of course carried out by people. People are required because design and development are fundamentally learning activities, not production. People skilled in each area must communicate richly, clearly, and frequently to make coordinated and consistent design, implementation, and delivery decisions that rapidly deliver the best results they are jointly capable of. Cross-functional teams take deep pride in delivering great products and effective business processes.

Solid software practices are necessary but alone are sadly insufficient. Instead, the value delivery problem must be thought of from the outside in. In this book, Carl Kessler and John Sweitzer offer a helpful approach to framing the problem, starting with identifying key stakeholders who determine the real value of the final result. They identify how decisions made by the product development team need to respect the contexts and typical concerns of each stakeholder category.

Outside-in thinking complements any approach your teams may be taking to the actual implementation of software, but it changes how you measure success. A successful outside-in team does a lot of learning and not much speculation. Agile software processes, which emphasize frequent feedback, are most likely to deliver the solutions all the stakeholders will truly value. But, without an outside-in context, even well-executed agile processes risk coming up with ‘technical success,’ which is a euphemism for business failure.

An outside-in approach is a systems thinking approach. Treating the end-to-end value chain as a value delivery system will not slay our menagerie of mythical creatures; it will enable them to function as a balanced ecosystem. This takes hard, disciplined work that integrates all levels of an organization. It is a higher level of thinking than the one that falls under the spell of the promise of yet another silver bullet.

—Tom Poppendieck

# Preface

**I**magine that your next development project successfully delivers precisely what your clients need, and was built by a productive, high-morale team. Your product achieves broad early adoption and positive references and reviews.

Sounds attractive, doesn't it?

This is the promise of outside-in software development. Using the concepts of outside-in development and practical guidance in this book, you can make this scenario your reality.

## Who this book is for

This book is for people who want to improve the products they work on.

The way we see it, this makes you a leader.

Maybe that's what your title says today, or maybe it is something you aspire to. Maybe your colleagues already think of you that way. Or maybe you don't really care about that stuff, and you just want to work on more successful products, on products that make a real difference to your clients.

If you see yourself in this description, you're the reader we want to reach.

Outside-in development applies across all development functions. We've written this book for an extremely wide range of readers, spanning executive, management, and nonmanagement roles. Outside-in development techniques can be used by coders, testers, technical writers, designers, support engineers, architects, user interface designers, performance stress testers, and folks in all sorts of additional specializations, including marketing, sales, business strategy, product management, business development, product pricing, finance, and services.

If we left out your role, it was unintentional.

Successful software product development is more than bug-free code. Winning products make the organizations that use them successful. These winning products are built by effective, cross-functional teams.

If you see yourself on such a team, this book is for you.

## How to read this book

Your authors use the words *you* and *we*, and occasionally our names, *John* and *Carl*, throughout this book. The approach is pretty simple: *You* means you the reader, no matter what your role.

*We* means we the authors. Thus, *John* or *Carl* refers to, well, one or the other of us.

We also use the word *client* quite often as a catchall term to refer to any possible customer of your product. Table 1 provides a handy decoder.

**Table 1** Who is being addressed, and by whom, in this book

	Who is being addressed, and by whom		
	You	We	“John” or “Carl”
Meanings	Anyone on the cross-functional development effort. This includes coders, testers, and technical writers, as well as market management, marketing, sales, services and support folks, and managers and executives.	The authors, presenting our particular point of view.	An author’s perspective, typically sharing a personal experience relating to the material being discussed.
	An explicit role within the cross-functional development team. Either you’ll know who you are from the context of the material (such as “test this way...”) or because we’ll be explicit about it (such as “for those of you who are testers...”) or in a sidebar (such as “the leader’s role in...”).		

## **Allow us to draw your attention to special topics**

We want you to be able to use the material in the book to fuel discussions with colleagues and allow you to quickly introduce outside-in development concepts to your project. To help you with that, we'll point out a few conventions.

Each chapter concludes with an *Essential points* wrap-up section.

Chapters 2–7 also include a review of *Key terms*.

You'll also find a section titled *The leader's role...* in each chapter, written for anyone who is interested in taking the opportunity to help her team.

## **Try out concepts and techniques as you read**

Here's a tip to make your experience with this book more productive. Build your own list of essential points as you finish each chapter, so you can start having conversations with your coworkers and using the techniques immediately.

Begin with the *Essential points* section and imagine how to apply these practices in your own environment. Decide which techniques you will try out next, either within your team or even for your own work items.

There's no need to wait until you've gotten to the last page of the book. Use these outside-in development concepts in conversations, meetings, client reviews, whatever: We're certain you'll be pleasantly surprised at how natural the approach feels in practice.

## **Why write about this topic?**

We wrote this book because our clients need software products that help their businesses succeed.

We know that they get software from many sources, not just from the firm we work for. We are all better off when our products meet our clients' real business needs, enabling their rapid and efficient business success.

Yet all too often, we see teams across the industry focus on the processes and methods of producing code without enough attention to whom might use it—or why they would want to.

*Outside-in Software Development* shares our experiences and practices, along with those of many of our colleagues, and presents a conceptual framework so that software development teams can readily understand, apply, and benefit. In the end, this means that businesses that use your and our software products will gain the most.

## An overview of the contents

*Outside-in Software Development* comprises seven chapters. The first chapter introduces the big-picture concepts of outside-in development, or *OID*.

The next four chapters provide the foundations of outside-in development. The remaining chapters lay out some practical, proven approaches to successfully implement *OID* in your development shop.

Here's what to expect from the rest of this book.

- Chapter 1, *Introducing Outside-in Development*

An overview of outside-in development (*OID*) thinking and what to expect from it.

- Chapter 2, *Understanding Your Stakeholders*

Outside-in development defines specific categories of stakeholders and practical approaches to clarify which of those stakeholders you will serve, what their goals are, and how you can satisfy those goals with your product.

- Chapter 3, *Understanding Organizational Context*

Techniques that reveal the ways your potential clients are currently organized and the ways they want to be organized in the future, providing insights that help in dialog about their goals, align design and development more effectively to meet those goals, and deliver winning software products.

- Chapter 4, *Making Products Consumable*

Introduces the term *consumability* along with practical techniques to assess and improve this notion of a product's capability to be rapidly and efficiently deployed, used, and supported.

- Chapter 5, *Aligning with Stakeholder Goals*

Continuously enhance your understanding of your stakeholders' goals and how your product will effectively satisfy those goals. Delivering a product that allows clients to achieve their business goals demands a variety of good practices and a way to continuously improve upon them; the practical approach to outside-in development includes many of these proven techniques.

- Chapter 6, *Defining Success in Your Stakeholders' Terms*

The outside-in development approach doesn't consider product development to end when the software ships to clients. Instead, the software life cycle extends into a *production* period for which the development team has accountability across three waves of activity.

- Chapter 7, *Becoming an Outside-in Developer*

You can apply outside-in development techniques at any point in the product life cycle; proven approaches to successful adoption will increase your odds of success.

## You can use outside-in development across the industry

The outside-in development approach is emphatically not limited to large organizations such as IBM.

Although we benefited by being able to distill and refine these techniques from hundreds of large- and small-scale projects, the lessons learned apply to small, localized teams as well as large, geographically distributed teams.

Probably the biggest difference between large and small teams is role overlap: A small team may find that individuals fill multiple roles. The product management lead on a small project, for example, may also be the lead programmer or lead marketer.

As mentioned earlier, when we write “you” without further specification, we intend for the reader to apply the material to his specific role or roles. The bottom line is that there’s no large-company magic here.

Everyone can use the outside-in development perspective.

### **Process- and methodology-independent**

When we say everyone can use outside-in techniques we refer as well to the variety of development processes and approaches that you might use. We’ve used outside-in development practices on large waterfall projects that took more than a year to ship as well as on leaner, more agile style projects that delivered iterations in four-week time-boxes. This book is not about yet another development process.

### **A business-to-business tone that applies broadly**

We wrote *Outside-in Software Development* from the perspective of a software vendor, building products for business customers. But wait, there’s more.

If you do in-house application development, work on open source projects, work for government agencies or nonprofit organizations, or just build software for fun, you’ll find the material applicable to you as well.

No matter *why* you build software, you should feel free to keep reading. Yes, the examples will be tuned to your software vendor colleagues.

But the content will be useful to everyone who wants to build the right software products, really well.

*This page intentionally left blank*

# Acknowledgments

**B**oth of the authors work for the IBM Corporation, where we've been using outside-in development approaches for many years across a wide range of products. We've been quite fortunate in that our seniormost leaders tend to be so heavily client-driven that outside-in development thinking comes naturally to them. As a result, most of our work has been to unify and extend practices already in use to make them more consumable by those developing software products.

This project started after a conversation Carl had with Alfred Spector, who was, at the time, the vice president of strategy and technology for IBM's Software Group. The topic of discussion was how to help our clients and partners build more successful software. Sending a small number of us around the world to share our practices in small groups is hardly scalable. In addition, as IBM steadily adds to the number of its developers, organically and through acquisition, we also needed a way to efficiently reach a large number of geographically dispersed people. Alfred pointed out that a book would be just the thing.

John started thinking about these issues when Al Zollar asked him to be the architecture lead for an "integration and usability project" back in the early 1990s. This assignment put in motion a set of collaborations which evolved John's design practices from inside-out to outside-in, based on the influence of colleagues such as Joe Pesot, Tony Temple, Carol Jones, Robert Uthe, Rod Smith, and Charles Hill.

We're building on a strong foundation of experience. Tony Temple, Dick Berry, Julian Jones, Carolyn Brown, and Karel Vredenburg introduced the term *outside-in design* as a next step in the engineering discipline for user interface development. Bernard Kreppel, Craig Zabell, and Bob Biamonte developed insights about the use of consumability score cards and workbooks, and Bill Woodworth, Scott Will,

Paul Gibson, and Ted Rivera identified and shared winning practices for quality and test improvement.

As you'd expect from any large organization, the senior executive leaders set the tone that drives the team's culture and behavior. At IBM, we start with a huge advantage for outside-in thinking: The first of our three corporate values is "Dedication to every client's success."<sup>1</sup> That is, of course, at the heart of outside-in development. But it is no surprise. Our bosses and colleagues have strongly encouraged stakeholder-based thinking for many years, notably Steve Mills, Danny Sabbah, Robert LeBlanc, Ambuj Goyal, Al Zollar, and Kristof Kloeckner. All of them deserve credit for building a successful global software development organization whose primary organizational mantra is to take a client-first view of the world.

We want to specifically recognize the leadership of Danny Sabbah and Kristof Kloeckner in introducing the notion of consumability for large-scale development projects and driving it into execution. We've also enjoyed the support of our colleagues and managers, especially Suzanne McKinney and Lee Roberts.

There is terrific camaraderie in our industry, and we've benefited from it greatly. We're greatly indebted to Tom Poppendieck, Per Kroll, Ted Rivera, and Christine Draper for reviewing our early manuscripts and for their many excellent suggestions. We are grateful beyond description, and also want to be clear that none of them is responsible for any of our errors.

---

1. The other two IBM values are "Innovation that matters, for our company and for the world" and "Trust and personal responsibility in all relationships." In July 2003, IBM's CEO, Sam Palmisano, invited all of IBM's 319,000 employees at the time to join a "values jam" on the corporate global intranet. Tens of thousands participated and it was the IBMers themselves who determined that our actions would be driven by these three values.