

Preface

Imagine 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.