用户中心设计

—— 集成化方法

（影印版）

USER-CENTERED DESIGN

An Integrated Approach

■ Karel Vredenburg

Scott Isensee

Carol Righi

PRENTICE HALL PTR

高等教育出版社
Higher Education Press

用户中心设计
——集成化方法

（影印版）

USER-CENTERED DESIGN

An Integrated Approach

Karel Vredenburg
Scott Isensee
Carol Righi

高等教育出版社

图字： 01-2003-3470 号

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

出 版 说 明

20 世纪末，以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用，带动了世界范围信息产业的蓬勃发展，为许多国家带来了丰厚的回报。

进入 21 世纪，尤其随着我国加入 WTO，信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展，但与发达国家相比，甚至与印度、爱尔兰等国家相比，还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力，最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材，在有条件的学校推动开展英语授课或双语教学，是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此，教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求，一是要高水平，二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下，经过比较短的时间，第一批由教育部高等教育司推荐的 20 多种引进教材已经陆续出版。这套教材出版后受到了广泛的好评，其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品，代表了目前世界信息科学技术教育的一流水平，而且价格也是最优惠的，与国内同类自编教材相当。这套教材基本覆盖了计算机科学与技术专业的课程体系，体现了权威性、系统性、先进性和经济性等特点。

目前，教育部正在全国 35 所高校推动示范性软件学院的建设，这也是加快培养信息科学技术人才的重要举措之一。为配合软件学院的教学工作，结合各软件学院的教学计划和课程设置，高等教育出版社近期聘请有关专家和软件学院的教师遴选推荐了一批相应的原版教学用书，正陆续组织出版，以方便各软件学院开展双语教学。

我们希望这些教学用书的引进出版，对于提高我国高等学校信息科学技术的教学水平，缩小与国际先进水平的差距，加快培养一大批具有国际竞争力的高质量信息技术人才，起到积极的推动作用。同时我们也欢迎广大教师和专家们对我们的教材引进工作提出宝贵的意见和建议。联系方式: hep.cs@263.net。

高等教育出版社
二○○二年九月

Foreword

Picture the following scenario: A software development team is gathered around a table, in the early stages of planning for the design, development, and implementation of a software product. Let's say it is a piece of business software to run on a Windows platform. The discussion has been going on for a while, and there is much ambient excitement; it's a good idea and the team members know it.

> Marketing Director: "All of our market research says we're right on target with the functionality. Competitive intelligence tells us there's no competition on the immediate horizon. But the key to our success will be getting to the marketplace quickly."
>
> Product Director: "Great Bob. Mary, how long do you expect design and development will take?"
>
> Software Development Manager: "We've been talking about that, and we think we have a plan to hit the target window. By eliminating all software testing, we believe we can have this baby ready to launch in three months."

Now, what happens next? High-fives all around? Or does the director start going through her mental Rolodex for a replacement software development manager?

Likely it's the latter. For it would be folly to try to cut development corners by eliminating testing; way too many coding defects would probably cripple the effort and the product would be a disaster shortly after shipping.

For at least three decades now, User-Centered Design professionals have been admonishing product development teams to systematically gather and analyze user data as a routine part of development. "You budget time to debug the code and the hardware," goes the familiar plaint, "why not budget time to debug the design? Do you think designing is any easier, or less prone to error?" Sometimes the product development manager recognizes the importance of systematic, professional UCD, and sometimes it takes a

disaster. It takes just one such disaster–a product that has wondrous functionality and is defect-free, but is a failure in the marketplace simply because people can't figure out how to carry out their tasks–to drive a product development manager to the joys of User-Centered Design.

Now that imaginary but, I hope, illustrative example may hold true for traditional computer products. But how about for the Web? Surely we don't need to worry about all this high-falutin' "User-Centered Design" approach if we're just building a Web site. Right?

Wrong. Indeed, four factors conspire to make a user-centered design approach even *more* important for Web sites and Web-based applications.

1. **Easy-to-use development tools open the field of "design" to more people.** The fact that I know how to use a hammer does not qualify me, let me assure you, to build a house. Just because someone has taken a short-course in HTML does not mean that that person can do a good job of designing Web pages. Now, I don't mean to imply that those who struggled to learn C++ are likely to be any better at design. It's just that now the population of "developers" is so much bigger, even more discernment is needed to identify those who are also good at design. And there are simply more "designs" out there (Web pages *ad infinitum*) that diverge from standards, to reflect a "designer's" joy of individuality, to frustrate the user who "just has a dang task to perform," and precious little time to perform it. I do *not* wish to imply that all software developers are bad designers. But when one looks, for example, at the software development curricula in colleges or at the wide array of Web sites that run from simple to use to "ohmigawd," one realizes that assuming all developers are good designers would be a bad bet—for the product development team and its investors.

2. **The time-to-market expectations for Web sites and Web apps are already anorexic, and on a diet.** Thanks to the aforementioned development tools, a person can get *some* web site up in, say, 20 minutes. Will it be useful? Will it be usable? Will any stray visitor likely return? These are all good questions. But there is a tendency for these questions to get buried in the noise, when the siren call of "how soon can you get the site up?" echoes through the development halls. When you are in a hurry, it is even *more* important to follow a (perhaps constrained, but intelligently selected) course of User-Centered Design, because, as I'll argue in the next paragraph, you may have only one chance.

3. **The expectations for Web sites are different than the expectations for traditional software applications.** Nobody but the serious propeller-

heads buys "Release 1" of any software product. Maybe, if you desperately needed the functionality, and were feeling a little saucy, you plunked down your money for Release 2. By the time Release 3 came out, you thought you'd give it a try, reasonably expecting the product not to crash your system or visit other unexpected events upon your computing life. For Web sites, you have one chance! When the buyer/ information seeker can go next door with a single mouse click you had better make that first visit a positive experience. Web companies who expected to weed out the design bugs in the first release are, for the most part, ex-Web companies.

4. **Who the heck *are* our users out there?** When a company is selling system management software to big companies, it is a relatively easy matter (if it'll just take the time to do the research!) to find out who its users will be. They'll have a certain range of education, a certain range of experience at systems management tasks, perhaps a certain percentage will have had experience with a competitor's product or a predecessor product. When your Web site goes live, it can be accessed by, literally, *everyone*. (If, that is, you've made it accessible for folks with disabilities—see Chapter 2 for more.) Will your site be as usable by a 50-year-old female social worker in Austin as it will by a 14-year-old tennis player in Fiji? Should you care? For Web sites and Web-based applications it is imperative to understand who your target audience is and to test representative users to see if they find your site usable.

So far I have addressed two main themes:

- Designing is not easier than coding or building hardware, so you should budget time to debug your design, and
- This is even truer on the Web than it is with traditional hardware, software, and service products.

Let me add a third, before lauding Vredenburg, Isensee, and Righi for excellently addressing all three.

- **There is danger in "amateur User-Centered Design."**

If a person claims to be a programmer and is a poor one, this fact gets revealed fairly early in the engagement (certainly by the time of system test). If a person claims to be a UCD engineer and is a poor one, this fact may not be revealed until the customer support team is inundated with calls, or the Web site has had hundreds of thousands of visitors ("thanks to that excellent marketing campaign we spent so much on") only to have 95% of them leave before making their intended purchase because they couldn't figure out how to navigate the site.

UCD is not rocket science, but it's not common sense, either. There are universities that offer Master's and Ph.D. programs in human-computer interaction, usability engineering, and related fields, but because of the rapid evolution of these fields, there's no universally accepted curriculum. And while there is a Board that certifies Human Factors Professionals and Professional Ergonomists, an important goal for such a certifying body is still to get businesses to understand the importance and value of a solid base of professional expertise and to hire and assign based on that understanding. And so the development manager who is interested in hiring a User-Centered Design expert, or the technical writer who wishes to become a UCD practitioner, or the product development team that wishes to implement a full course of User-Centered Design, all have uncertain paths before them.

They all will gain clarity about their path if they read this book. Vredenburg, Isensee, and Righi are, individually and as a team, a rare mix. They have all spent years in the usability and UCD trenches, being absolute models of how to support software products empirically, creatively, and cost effectively. (I have had the pleasure of working directly with each of them and so have firsthand data on this matter.) Also, they have all trained others, in formal classes, in conference tutorials, and as informal mentors. And they have shared their wisdom with the field via conference presentations, books, and technical articles. Among them, they have a combined 50+ years carrying out, inventing, managing, proselytizing, training, and documenting UCD methods. This book represents a crisp, readable, actionable distillation of all their accumulated knowledge about the invaluable design approach that is User-Centered Design.

I've said that the authors are expert at conducting and teaching UCD methods, and that's true. But this book—as suggested by its subtitle, "An Integrated Approach"—is more than a series of how-to chapters. Their presentation of an "integrated approach" includes some historical perspective (as they help you anticipate the "Yeah, buts" you're likely to hear). It offers time-tested advice on driving user-centered design into your organization, on cost-justifying (quantifiably) your usability expenditures, on good communication in order to drive awareness throughout your organization. With pedagogic tools like case studies (many illustrative case studies!), FAQs, and an included CD with tools for the practitioner, they not only present the content, but demonstrate their sincerity about "tooling up" the emerging UCD practitioner or software development team.

I believe the key to successfully kicking off a User-Centered Design approach, or to becoming a successful UCD practitioner, is discernment. It is important to discern *what* sorts of user data are needed *when* in the develop-

ment cycle. It is important to discern *which* methods to employ to gather those data. It is important to discern *who* to pursue as test participants, *how* to carry out the evaluations, and *how* to apply these data to design. And it is important to discern which methods the emerging usability engineer, or the team just now pursuing a User-Centered Design approach, might likely carry out successfully, and which to save for the experienced professional. This book will clearly communicate to both the emerging and experienced UCD practitioner, manager, or other IT professional what he or she doesn't know, and what he or she needs to know, and will go a long way towards bridging that gap.

A few months ago I attended an open house for a software development company that makes and sells wireless computing applications. I sought out the director of user interface design, with the intent of selling her some systematic usability engineering services. She was a pleasant person and welcomed my informal sales pitch. However, she said they already "did" UCD. "Oh, yes. I show the interface around to everyone in the office before we ship." I tried a little harder to convince her of the importance of a systematic, professional UCD approach, and she listened politely. By this point in the evening we were both out of business cards. "Here," she said brightly, "I'll just email you on my phone, using our new product, and then we'll both have each other's contact information." I said that would be great, and spoke aloud my email address. Ten minutes and many grimaces later she *thought* she had successfully sent one email. If they're still in business, I will send her a copy of *User-Centered Design: An Integrated Approach.*

Randolph G. Bias, Ph.D.
Austin Usability
Austin, TX
August, 2001

Preface

Do you want to change what people say about your product or system, referring to it as "elegant, simple, and powerful" rather than "ham-fisted, ugly, and unusable"? Or perhaps they're not saying anything at all. Then this book is for you. It provides an integrated approach to User-Centered Design (UCD) with an emphasis on using UCD to make products easy to buy, learn, and use. It focuses on designing a compelling "total customer experience" — everything a customer sees, hears, and touches about a product or system.

The integrated version of User-Centered Design, described in this book, was initially developed at IBM in the early 1990s by Karel Vredenburg. A team of experts at IBM continuously worked with the first author to improve this version of UCD. It is currently in its third major version. Scott Isensee and Carol Righi were instrumental in the evolution of the approach not only while they were at IBM but also after they left the company. Since leaving IBM, Scott has applied UCD effectively to the design of the i-opener information appliance at NetPliance and currently implements UCD at BMC Software. Carol formed her own company, Righi Interface Engineering, Inc. She has led UCD projects both while at IBM and now as a consultant for clients such as Chrysler, MetLife, and the Usability Professionals' Association and has taught User-Centered Design classes to major corporations. We continue to run a highly popular workshop entitled, "How to Introduce, Deploy, and Optimize User-Centered Design in Your Organization."

This book is a distillation of our collective experience and that of our colleagues in introducing UCD to many hundreds of companies and deploying it on a few thousand projects over the past 10 years. We have used the approach to design products ranging from mainframe computers to integrated circuits, notebook computers to Web appliances, database software to speech recognition software, Web site portals to the Web site for the Olympics, and we have used it on consulting projects worldwide for many

industries, including healthcare, finance and banking, aerospace, insurance, automotive, and retail.

We sincerely hope that you enjoy reading the book and applying the information contained within it.

Acknowledgments

The process of defining and implementing integrated UCD has involved many people. As the process continues to become an ever more important part of the way products are developed, more people are becoming involved. The people who practice UCD every day are making it a success. Special thanks goes to John Schwarz for providing Karel Vredenburg 6 months and a team with whom to develop UCD at IBM, Hershel Harris for leading the first successful UCD project, Al Zollar for helping to drive the first companywide UCD communications campaign, Steve Mills for recognizing the importance of UCD to the IBM software development and his support ever since, Lou Gerstner for the insight to drive UCD across all of IBM, and Tony Temple and Susan Mills for their continued leadership and support of the Ease of Use program at IBM. The IBM UCD Advisory Council, a group of approximately 30 discipline leaders and divisional UCD leaders from across the company, has made significant technical contributions to UCD. We can't mention everyone, but we would like to acknowledge a few people who have had the most substantial technical influence: Paul W. Smith, Mike Fischer, Julian Jones, Bob Jones, Frank Eldredge, Carolyn Bjerke, Rick Herder, John Boscarino, Simon Hakiel, Ken Stern, John Karat, Jim Lewis, Mike Stokes, Linda Liebelt, Dirk Willuhn, Mike DiAngelo, Charlotte Schwendeman, Paul McInerney, Dick Berry, Alan Tannenbaum, and Colin Powell. Case study information was contributed by Michael Boshes, Robert Atlas, Tony Haverda, Karen Lefave, Lynn VanDyke, Marion Radin, and Diane Wilson.

In addition to our coworkers, our peers at other companies have generously contributed tips from their experience and reviewed this manuscript to assure us that the advice we give is broadly applicable and accurate.

We are grateful to our students and development team members who have challenged our assumptions, prompted us to explain more clearly, and helped us to practice UCD on the UCD process itself. UCD is more efficient and practical because of it.

Thanks to the reviewers who provided many helpful comments and suggestions on this book.

Thanks most of all to our families, Erin O'Brien; Elliot, Emma, Rowan, and Noah Vredenburg; Dawn Isensee; and Rob and Sarah Ripperdan for their love and support during the writing of this book.

Introduction

User-Centered Design (UCD) makes products easy to use. Unfortunately, making products easy can be hard work. The goal of this book is to reveal the techniques we have found to be successful in developing and implementing a UCD program.

Markets for most products are becoming more competitive. Products that have unacceptable usability typically do not survive. Customers are demanding usable products. At the same time that customer expectations are rising, developers are being challenged to produce products on shorter schedules at lower cost. In this environment, techniques that are very efficient and effective are needed. The techniques discussed in this book are meeting that challenge at IBM and many other companies.

This book is written for the broad range of UCD professionals. For those who are new to the field or are starting a UCD program in an organization, it outlines the best approaches for introducing UCD. For those who are experts in the field or for organizations with mature UCD programs, it provides ways to optimize your implementation of UCD.

The purpose of this book is not to provide a comprehensive usability engineering handbook; nor is it intended to teach you how to "do design." Rather, this book is intended to complement those skill-oriented volumes that offer those approaches by providing practical advice on how to prepare for, deploy, and optimize an overall UCD approach into an organization.

Chapter 1 provides information that you can use to take stock of your current organization's position regarding the core elements of UCD. Chapter 2 summarizes the various aspects of our integrated approach. The critically important step of introducing UCD to your organization is the focus of Chapter 3, and Chapter 4 provides a detailed account of how to deploy and carry out UCD. Chapter 5 examines various ways to optimize your deployment of UCD with a variety of tools and technologies and discusses future trends in methodology and technology integration. The book contains

numerous case studies that describe how UCD was used in real-life projects, from small applications intended for in-house use to large-scale consumer products intended for a worldwide market. The bibliography provides additional resources beyond the material in the chapters. In addition, a CD with useful information, Web site links, and tools is included at the back of the book.

We recommend that you read the book from Chapter 1 through to Chapter 5. However, to facilitate the use of the book as a reference, we designed each chapter to stand alone to a certain degree. If you have read the entire book and now want to start to introduce your organization to UCD, you should proceed to Chapter 2 directly. Also, if you are an experienced UCD practitioner and are interested in optimizing your deployment of UCD, you may want to proceed directly to Chapter 5, skimming the preceding chapters as necessary.

Given that we practice User-Centered Design in everything we do, we are very interested in getting feedback from you, the book's users. When you are finished with the book, we would appreciate it if you would visit the Web site address *www.righiinterface.com/ucd/survey* to complete a brief survey about your experience with the book.

This book provides best practice based on our experience developing and implementing a UCD program at IBM and numerous other companies at which we have worked and for whom we have consulted. We hope that you will find the information in this book to be useful in helping to transform and/or optimize your own organization.

Contents _____